Blockchain Aplicado a Sistema Eleitoral

Felipe Lira de Oliveira, Pedro Henrique Pires Flores

Orientador: Dr. Dionísio Machado Leite Filho

[1]Faculdade de Computação (Facom) – Universidade Federal de Mato Grosso do Sul (UFMS) 79.070-900– Campo Grande – MS – Brazil

Abstract. *The present work proposes the investigation and implementation of an innovative electoral system, using* blockchain *technology as a foundational mechanism to ensure the security, transparency, and integrity of the voting process. With the growing concern about the reliability of electoral systems and the increasing incidents that compromise the legitimacy of elections, it becomes imperative to seek technological alternatives that can offer substantial improvements over traditional methods. In this context,* blockchain *emerges as a promising solution capable of mitigating issues related to fraud and security failures. This study focuses on the development of a prototype using the Hyperledger framework and smart contracts to create an electoral infrastructure that can be auditable, irrefutable, and free from external manipulation. The introduction provides a historical overview of elections in Brazil, exploring the challenges faced by the current system and the necessary evolution to meet the demands of a fair electoral process. Subsequently, the discussion delves into the concepts and functionalities of* blockchain*, specifically Hyperledger and smart contracts, establishing the theoretical foundations for the development of the proposed system.*

Resumo. *O presente trabalho propõe a investigação e implementação de um sistema eleitoral inovador, utilizando a tecnologia de* blockchain *como mecanismo de base para garantir a segurança, transparência e integridade do processo de votação. Com a crescente preocupação acerca da confiabilidade dos sistemas eleitorais e o aumento de incidentes que comprometem a legitimidade das eleições, torna-se imperativo buscar alternativas tecnológicas que possam oferecer melhorias substanciais em relação aos métodos tradicionais. Nesse contexto, o* blockchain *emerge como uma solução promissora, capaz de mitigar problemas de fraudes e falhas de segurança. Este estudo concentra-se no desenvolvimento de um protótipo utilizando o* framework *Hyperledger e contratos inteligentes para criar uma infraestrutura eleitoral que possa ser auditável, irrefutável e livre de manipulações externas. A introdução traça um panorama histórico das eleições no Brasil, explorando os desafios enfrentados pelo sistema atual e a evolução necessária para atender às demandas de um processo eleitoral justo. Em seguida, adentra-se na discussão sobre os conceitos e funcionalidades do* blockchain*, especificamente o Hyperledger e os contratos inteligentes, estabelecendo as bases teóricas para o desenvolvimento do sistema proposto.*

1. Introdução

O voto é a manifestação de uma escolha de um indivíduo por uma opção. Em sistemas eleitorais, é sinônimo de sufrágio, o qual é um direito do cidadão de expressar sua vontade na política elegendo seus representantes, seja de forma direta ou indireta, secreta ou pública (SUFRÁGIO, 2023). As democracias são regimes que durante um determinado ano satisfazem quatro critérios simultaneamente, entre eles a eleição do chefe do poder executivo e a eleição do legislativo (CHEIHUB; PREZEWORSKI, 1997). Alguns países como o Butão, Índia, Paraguai e Emirados Árabes Unidos dispõem de urnas eletrônicas, mas possuem algum tipo de comprovante de votação verificado pelo eleitor, o VVPAT (Voter Verifiable Paper Audit Trail) (IDEA, 2023).

No Brasil, até 1996 predominava o sistema de cédulas. "Em 1881, o eleitor colocava num envelope os seus votos através de cédulas confeccionadas pelos candidatos e partidos políticos, contendo o nome do candidato e partido, identificado pelo nome e pela cor partidária. Estes envelopes deveriam ser lacrados com goma arábica, disponível nas cabines. Uma vez lacrado, o envelope era colocado na urna" (TRE-MT, 2023).

Atualmente, o voto continua sendo por meio da urna eletrônica, mas há diversos estudos que colocam em dúvida a integridade, segurança e, principalmente, a auditabilidade do processo eleitoral. Uma equipe de segurança da informação da UNICAMP realizou, em 2017, uma série de testes nas urnas utilizadas nas eleições brasileiras, e o resultado foi a descoberta de inúmeras falhas no sistema das urnas anteriores ao modelo de 2020. Dentre as falhas, houve a possibilidade de alterar a mensagem mostrada para o usuário após o registro de seu voto e também houve progresso nas tentativas de desvio de votos de um candidato para o outro. (RIBEIRO; MENDIZABAL, 2021).

Outros países têm explorado o potencial da tecnologia para aprimorar seus sistemas eleitorais. Um exemplo notável é a Estônia, que se destaca internacionalmente por seu sistema de votação eletrônica. Desde 2005, a Estônia tem permitido que seus cidadãos votem pela internet, utilizando uma abordagem pioneira que combina identificação digital segura e infraestrutura de governo digital. O sistema estoniano representa um marco no que se refere à conveniência e modernização dos processos eleitorais (TSE, 2022).

A experiência da Estônia é, particularmente, relevante para este estudo, uma vez que exemplifica a viabilidade de um sistema eleitoral totalmente digitalizado, *on-line*, e proporciona um modelo de como a tecnologia de *blockchain* poderia ser implementada para reforçar a integridade e auditabilidade do voto. Assim, ao considerarmos o desenvolvimento de um sistema de votação baseado em *blockchain*, é instrutivo observar lições aprendidas com o caso estoniano, que ilustra os benefícios e desafios de se digitalizar completamente o processo eleitoral.

Dado o sucesso do *blockchain* no setor financeiro (PUC-RS, 2023), proporcionando confiabilidade e segurança ao *bitcoin*, a presente pesquisa se dedica a explorar sua aplicabilidade no contexto eleitoral. A proposta deste artigo é desenvolver um sistema de votação baseado em *blockchain* que promova a auditabilidade e verificação dos votos, potencializando a transparência e confiança nos processos eleitorais.

Assim, o objetivo deste trabalho é projetar um sistema de votação eletrônica baseado no *blockchain*, utilizando o framework Hyperledger e contratos inteligentes *solidity*, da *blockchain Ethereum*, a fim de aprimorar a integridade, segurança e auditabilidade nos

processos eleitorais.

Destacam-se como objetivos específicos:

1. Investigar e aplicar os princípios de descentralização e transparências proporcionados pelo *blockchain* para resolver questões associadas aos sistemas eleitorais atuais
2. Demonstrar a viabilidade técnica e a eficácia de um sistema eleitoral inteiramente eletrônico que incorpore a imutabilidade e rastreabilidade dos dados presentes em estruturas de *blockchain*.
3. Contribuir para o corpo de conhecimento sobre aplicações de *blockchain* fora do domínio financeiro, especificamente no contexto da governança eleitoral, e fornecer uma análise comparativa da eficiência, confiabilidade e segurança entre os sistemas eleitorais tradicionais e o sistema proposto.

Para alcançar os objetivos elencados neste artigo, a metodologia empregada na criação do sistema de votação eletrônica proposto é fundamentada na utilização do Hyperledger, um framework de *blockchain* de código aberto e modular, ideal para desenvolver soluções de *blockchain* empresariais. Este framework é a espinha dorsal do sistema, fornecendo a infraestrutura necessária para criar um ambiente de votação seguro, privado e resistente a falhas.

2. Referencial Teórico

A seguir são abordados os conceitos necessários à compreensão e realização do trabalho. O conceito de *peer-to-peer* é a base da estrutura de dados *blockchain* que, por sua vez, utiliza a *hash* como algoritmo matemático para trazer segurança e valida as transações efetuados por meio de algoritmos de consenso.

2.1. *Peer-to-Peer*

Sistemas *Peer-to-peer* (P2P) são ”um modelo de rede em que os nós atuam tanto como cliente quanto como servidor para os outros nós, permitindo que os recursos sejam compartilhados sem a necessidade de um servidor central”(COULOURIS; DOLLIN; HARIDAS, 2011, p. 4).

Complementando a definição de P2P apresentada por Coulouris et al. (2011), Tanenbaum e Van Steen (2008) descrevem o modelo de maneira complementar, destacando que ”em sistemas P2P, cada participante pode atuar como um cliente ou servidor ou ambos. Essa estrutura dá origem a uma forma resiliente e escalável de compartilhar recursos, promovendo eficiência na distribuição de carga, e maior tolerância a falhas devido à distribuição descentralizada dos recursos”(TANENBAUM; STEEN, 2008). Os autores reforçam a ideia de que em redes P2P os nós agem simultaneamente como clientes e servidores, compartilhando recursos de forma distribuída, sem a necessidade de um servidor central.

A rede P2P é composta de vários computadores ao redor do mundo conectados por um *software* comum. Nessa conexão virtual, os computadores acessam benefícios do sistema e, em troca, fornecem suporte a ele. No torrent, por exemplo, os usuários, ao baixarem arquivos, também os compartilham com outros, transformando seus computadores em servidores. Quanto mais usuários utilizam a rede, maior é a capacidade do sistema, pois mais servidores estão disponíveis (ABROL, 2023).

As redes P2P podem enfrentar limitações, como a dificuldade de impedir a disseminação de conteúdo inválido por usuários mal-intencionados. No entanto, a *blockchain* superou essas limitações com inovações como o *proof-of-work* (prova de trabalho), um protocolo que exige poder computacional e proporciona segurança à rede (ABROL, 2023).

2.2. *Blockchain*

Blockchain, em português "cadeia de blocos", inicialmente utilizada pelos sistemas de criptomoedas, pode ser descrita como um arquivo digital, que se mantém atualizado e disponível a todos os usuários da rede. Esta cadeia pode ser vista como um livro razão, contendo todas as transações já realizadas e verificadas, em ordem cronológica. Livro-razão é um livro do processo contábil por meio do qual é possível controlar separadamente a movimentação de todas as contas de uma empresa (RIBEIRO; MENDIZABAL, 2021).

Uma transação inserida na *blockchain* ficará lá permanentemente, não sendo possível a alteração ou exclusão deste registro, servindo como prova de sua validação. As transações podem ser qualquer registro de dado, como certidão de nascimento, registro de imóveis, recibos de serviço e transações financeiras, por exemplo.

Quando um registro é adicionado a um bloco da rede, junto com outros registros, este bloco é disseminado pela rede e, através de um protocolo de consenso, os nós participantes da rede certificam e atestam que o bloco é válido e juntos decidem adicionar o bloco à *blockchain*. Uma vez que o bloco está inserido, passa a integrar o sistema e se torna imutável. Sendo assim, apenas após a validação, o registro ou transação têm efeito de fato (RIBEIRO; MENDIZABAL, 2021).

RIBEIRO e MENDIZABAL (2021) também apresentaram 3 formas de *blockchain*, são elas:

Blockchain privada, que apresenta um ambiente controlado no que diz respeito à identidade dos participantes e possui maior nível de centralização. Seu uso é vantajoso em redes empresariais ou locais em que o acesso dos participantes da rede deve ser restrito.

A *blockchain* pública por sua vez, está na rede aberta a qualquer usuário, sem ser necessária a identificação dos nós e assim, pode ser totalmente descentralizada, sem que exista alguma autoridade ou controle dentro da rede. Este ambiente se aplica à maioria das criptomoedas.

Já as *blockchains* de consórcio são uma mistura da pública e privada. Pode permitir acesso público, porém mantendo algum nível de centralização, onde alguns nós selecionados podem ter controle maior e privilégios na rede.

Foram listadas por REBELLO et al. (2019), as propriedades da *blockchain*:

1. Descentralização: As transações em uma *blockchain* são realizadas de forma distribuída, sem a necessidade de uma entidade central controladora. O consenso é alcançado entre todos os participantes da rede.
2. Desintermediação: A *blockchain* elimina a necessidade de intermediários confiáveis para a troca de ativos. As transações podem ocorrer diretamente entre os participantes da rede, com a confiança estabelecida por meio do consenso.

3. Imutabilidade: Os dados armazenados em uma *blockchain* são permanentes e não podem ser modificados retroativamente. Qualquer alteração na *blockchain* é feita incrementalmente, garantindo a integridade dos dados anteriores.
4. Irrefutabilidade: As transações na *blockchain* são registradas de forma que não podem ser negadas ou alteradas, graças à imutabilidade da *blockchain*. Isso impede que os participantes neguem o envolvimento em transações anteriores.
5. Transparência: Todos os dados armazenados na *blockchain* são acessíveis a todos os participantes da rede. Nas *blockchains* públicas como Bitcoin e Ethereum, as transações são abertas e acessíveis a qualquer usuário com conexão à internet.
6. Auditabilidade: Devido à transparência da *blockchain*, todos os participantes podem verificar e auditar as transações registradas na *blockchain* em busca de erros ou atividades maliciosas. Isso também pode ser usado para responsabilizar os participantes por atividades indesejadas em blockchains federadas.
7. Disponibilidade: A estrutura da *blockchain* é replicada em cada participante da rede, garantindo a disponibilidade do sistema mesmo em caso de falhas, devido à redundância das informações.
8. Anonimidade: Os usuários e mineradores de uma *blockchain* são identificados por chaves públicas ou identificadores únicos, preservando suas identidades. A utilização de chaves públicas em cada transação aumenta o nível de anonimidade e dificulta o rastreamento dos usuários.

2.3. *Hash*

Função *hash* é um algoritmo matemático que toma como entrada um dado de tamanho arbitrário e transforma-o em um texto cifrado de tamanho fixo. A saída é chamada de *hash*. Mesmo uma pequena alteração em uma série de dados, cria um *hash* completamente novo. Essas funções são usadas em armazenamento de senhas, autenticação de mensagens, assinaturas digitais, certificados e em *blockchains* (MACHARIA, 2021).

A natureza matemática do algoritmo *hash* torna praticamente impossível a reversão de um *hash* para obter a entrada original. Isso significa que, uma vez que um bloco é adicionado à blockchain com o seu *hash* calculado, é extremamente difícil para alguém alterar os dados dentro do bloco sem ser imediatamente detectado (BOMBIG, 2023).

Em *blockchains*, cada bloco contém uma *hash* do bloco anterior. Este encadeamento garante a integridade da informação, tornando extremamente difícil alterar blocos antigos sem alterar todos os blocos seguintes. Extremamente difícil, pois fazer tal alteração exigiria um poder computacional massivo (UFRJ, 2023).

Portanto, a função *hash* serve como um alicerce essencial para a confiança e a segurança dos sistemas de votação baseados em *blockchain*, garantindo a integridade das transações e a legitimidade dos resultados eleitorais, uma vez que não podem ser alterados por terceiros.

2.4. Algoritmos de Consenso

Para garantir a consistência da *blockchain*, afirmaram REBELLO et al. (2019), que é necessário que múltiplos nós da rede validem as informações antes de adicionar um novo bloco. Como diversos nós podem realizar essa validação de forma simultânea, são estabelecidos os protocolos de consenso. Estes protocolos permitem que os nós cheguem

a um acordo sobre as transações válidas, garantindo a integridade e a confiabilidade da *blockchain*.

Nas *blockchains* públicas, qualquer nó da rede tem a capacidade de atuar como validador. Este modelo de consenso opera por meio da prova de trabalho (*proof-of-work* em inglês), que envolve a resolução de um desafio criptográfico chamado mineração, resultando em um alto custo computacional. A prova de trabalho traz consigo a vantagem de aumentar o número de validadores à medida que mais participantes entram no sistema, porém, isso pode impactar a velocidade das transações, uma vez que estas devem ser validadas por todos os participantes, podendo gerar múltiplas bifurcações (*forks*) na *blockchain*. Para lidar com esse cenário, é necessário aplicar a regra do ramo mais longo como vencedor. Dessa forma, a prova de trabalho representa um mecanismo de consenso probabilístico.

No entanto, (BLASCO, 2023) relatou que as *blockchains* que empregam o mecanismo de prova de participação (*proof-of-stake* em inglês) apresentam a preocupação com a concentração de poder de validação, uma vez que um nó que detenha mais de 50% das moedas teria a capacidade de controlar e potencialmente interferir nas transações sem a necessidade de obter o consenso do restante da rede.

A prova de história (*proof-of-history* em inglês) utiliza a *hash* de um bloco minerado como entrada na *hash* do próximo bloco, estabelecendo um mecanismo de verificação criptográfica para comprovar a passagem do tempo entre dois eventos. Por meio de uma função *append- only* que gera uma saída impossível de prever a partir da entrada, a prova de história demanda execução completa para produzir o resultado.

Em suma, os diversos algoritmos de consenso em blockchain refletem a evolução e a busca por soluções que conciliem eficiência, segurança e descentralização nas transações. Cada mecanismo apresenta desafios e vantagens específicas, influenciando diretamente na arquitetura e no funcionamento das *blockchains*.

2.5. Hyperledger

Hyperledger Fabric, segundo (Hyperledger Fabric, 2023) é uma plataforma de *ledger* distribuído (*Distributed Ledger Techonology*, DLT) de código aberto, desenvolvida especificamente para aplicações empresariais. Diferencia-se por sua arquitetura modular e configurável, que permite adaptação a uma variedade de casos de uso empresarial.

É notável por permitir a escrita de contratos inteligentes em linguagens de programação convencionais, como Java, Go e Node.js, facilitando a integração com práticas de desenvolvimento existentes. A plataforma também é permissionada, onde a identidade dos participantes é conhecida, permitindo a criação de redes com modelos de governança baseados na confiança mútua e no cumprimento de regulamentações como KYC (*Know Your Customer*, "Conhecer Seu Cliente") e AML (*Anti-Money Laundering*, "Anti-Lavagem de Dinheiro").

O suporte para protocolos de consenso plugáveis permite que seja customizada para atender requisitos específicos de desempenho e confiança, tornando-a uma escolha robusta para organizações que buscam implementar soluções de blockchain em ambientes regulados e com demandas de alto desempenho.

2.6. Contratos Inteligentes

ROCHA e LIMA (2023) definiram que contratos inteligentes são como códigos implantados sobre uma *blockchain* que facilitam a execução e cumprimento de cláusulas de um acordo estabelecido entre as partes envolvidas. Estes contratos são executados de maneira totalmente transparente na *blockchain*, de tal forma que, quando as partes concordam com os termos acordados, a rede *blockchain* executa o processamento de forma autônoma, uma vez que os dados e regras do contrato estão armazenados distribuída e imutavelmente na própria *blockchain*.

O registro em uma *blockchain* ocorre em três etapas distintas: implantação, execução e conclusão. Na etapa de implantação realiza-se a programação do contrato inteligente utilizando linguagens específicas como *solidity*, ACT, PACT, entre outras. Nessa fase, define-se a lógica e a especificação do comportamento do contrato inteligente através da codificação. Após a implantação, inicia-se a etapa de execução, na qual o contrato inteligente é acionado conforme as regras previamente estabelecidas durante a programação. Por fim, a etapa de conclusão ocorre quando todas as condições programadas são satisfeitas, findando assim o ciclo de vida do registro na rede distribuída.

A *solidity* é uma das linguagens de programação mais populares para a criação de contratos inteligentes, por ser uma linguagem moderna, de alto nível, orientada a objetos e inspirada em JavaScript.

2.7. *Solidity*

Solidity é uma linguagem de programação orientada a contratos inteligentes, usada para escrever aplicações descentralizadas (DApps) no *blockchain* Ethereum. Criada por Gavin Wood, Christian Reitwiessner e outros colaboradores da Ethereum, é uma linguagem de alto nível influenciada por JavaScript, C++ e Python, e foi projetada para ser fácil de entender e utilizar por desenvolvedores que já têm experiência com uma ou mais linguagens de programação modernas (SOLIDITY, 2023).

Os contratos inteligentes escritos em *solidity* são programas que executam exatamente da forma como foram programados, sem possibilidade de tempo de inatividade, censura, fraude ou interferência de terceiros. Eles são compilados em bytecode, que pode ser lido e executado na Ethereum Virtual Machine (EVM). Através dos contratos inteligentes, os desenvolvedores podem criar uma variedade de aplicações descentralizadas e tokens digitais, realizar transações automáticas e implementar regras de negócio complexas em ambientes descentralizados.

2.8. *Chaincode*

Vale (2020) nos diz que *chaincode* "é a representação dos *smart contracts* dentro da "Hyperledger Fabric" e que "é utilizado para definir termos e regras de um processo e, diferente dos *smart contracts*, pode integrar diferentes projetos em um único bloco". O *chaincode* pode ser escrito em Go, node.js ou Java que implementa uma interface pré-definida. O *chaincode* gerencia o estado do ledger por meio de transações submetidas pelas aplicações (Hyperledger Fabric, 2023).

O *chaincode* gerencia a lógica de negócio de um acordo firmado entre membros da rede blockchain, similar a um contrato inteligente.

3. Trabalhos Relacionados

É notória a quantidade de pesquisas científicas produzidas envolvendo *blockchain* com finalidades eleitorais. Para este trabalho, foram selecionados alguns estudos relevantes que servem de base para essa implementação e podem servir para futuras pesquisas na área.

MARTINHO e JABOUR (2019), buscaram desenvolver uma solução que pudesse substituir o sistema de eleição/votação tradicional por meio de uma aplicação baseada na *blockchain Ethereum*, levando-se em consideração o custo, a segurança e remoção das influências externas e disponibilidade do resultado final da eleição. Foi feito um teste com 8 eleitores e a coleta de algumas métricas como custo e velocidade.

Por fim, não conseguiram criar um sistema de autenticação de pessoas on-line de forma segura e sugeriram que fosse feito um cadastro físico ou verificação de informações cadastradas dos candidatos e eleitores em um banco de dados nacional seguro. Também não foi criado um mecanismo que permitisse outros turnos da eleição como, por exemplo, o segundo turno.

Conforme descrito por EHIN et al., (2022), a Estônia é um dos países mais digitalizados do mundo com relação à eleição, com uma infraestrutura de e- governo bem estabelecida e uma população que está confortável com o uso de serviços digitais em seu cotidiano. Isso inclui uma identidade digital segura para todos os cidadãos, que permite o acesso a uma ampla gama de serviços públicos e privados online.

A adoção do voto pela internet na Estônia não foi um evento isolado, mas sim parte de uma estratégia mais ampla para digitalizar a sociedade e tornar os serviços mais acessíveis. A familiaridade e confiança dos cidadãos estonianos na tecnologia digital são fundamentais para a alta taxa de adoção do voto pela internet. O governo estoniano promoveu ativamente o uso de soluções digitais e investiu em segurança cibernética, o que ajudou a construir a confiança do público.

O sistema de votação pela internet foi projetado para ser seguro, transparente e fácil de usar, o que é crucial para manter a confiança dos eleitores. A capacidade de votar de qualquer computador conectado à internet em todo o mundo oferece conveniência e acessibilidade, especialmente para aqueles que podem estar viajando ou vivendo no exterior durante as eleições.

A confiança no sistema é sustentada por medidas rigorosas de segurança, como a autenticação forte do eleitor e a verificação do voto, que permitem aos eleitores verificar se seus votos foram registrados corretamente. Essas medidas ajudam a garantir que o sistema de votação pela internet não apenas atenda aos padrões normativos para eleições democráticas, mas também reforce a confiança dos eleitores na integridade do processo eleitoral

O sistema de votação pela internet na Estônia permite que todos os eleitores emitam seus votos de qualquer computador conectado à internet em qualquer lugar do mundo. Desde a sua introdução em 2005, o sistema foi utilizado em várias eleições nacionais e do Parlamento Europeu, tornando-se uma opção popular e amplamente utilizada pelos eleitores estonianos. O processo de votação pela internet na Estônia é organizado da seguinte forma:

1. Autenticação do Eleitor: Os eleitores se autenticam usando sua identidade digital segura, que é parte integrante da infraestrutura de e-governo da Estônia.
2. Emissão do Voto: Após a autenticação, os eleitores podem emitir seus votos eletronicamente. O sistema é projetado para ser fácil de usar e acessível.
3. Segurança e Verificação: O sistema possui medidas rigorosas de segurança para garantir a integridade do voto. Isso inclui a verificação do voto, que permite aos eleitores confirmar se seus votos foram registrados corretamente.
4. Contagem e Tabulação: Os votos são contados e tabulados eletronicamente, com procedimentos estabelecidos para garantir a transparência e a precisão dos resultados.
5. Privacidade e Anonimato: A privacidade e o anonimato dos eleitores são preservados durante todo o processo de votação.

O sistema de votação pela internet na Estônia é um exemplo de como a tecnologia pode ser utilizada para facilitar o processo democrático, oferecendo conveniência e acessibilidade aos eleitores, ao mesmo tempo, em que mantém os padrões normativos para eleições democráticas.

Em 2018, o Secretário de Estado de *West Virginia*, Mac Warner, nos EUA, introduziu um protótipo de votação online por meio de um aplicativo criado pela Voatz, direcionado aos cidadãos eleitores em serviço militar ou residentes no exterior (MOORE; SAWHNEY, 2019). A iniciativa permitia que os eleitores utilizassem dispositivos iPhone com iOS 10.0 ou superior, ou Android com versões superiores a 6.0, para votar de qualquer local do mundo.

Os votos foram registrados em uma urna criptografada e armazenados em um sistema distribuído de servidores *blockchain*, utilizando a plataforma de código aberto Hyperledger. Uma vez registrados na *blockchain*, os eleitores puderam revisar seus votos, anular seu voto anterior e solicitar uma nova tentativa de voto. No entanto, é importante ressaltar que devido à natureza imutável da *blockchain*, ambos os votos (anulado e o segundo) foram registrados, mas apenas o último voto foi considerado válido para contagem eleitoral.

Esse caso exemplar de *West Virginia* destaca como a utilização da *blockchain* em processos eleitorais pode proporcionar segurança e acessibilidade para os cidadãos, mesmo quando se encontram distantes de seu país. No entanto, permanecem desafios cruciais a serem endereçados para garantir a integridade e a confiabilidade do voto eletrônico por meio da *blockchain*.

Com base nos trabalhos revisados e nos trabalhos relacionados implementados com sucesso em países como Estônia e o estado norte-americano de West Virginia, nota-se que sistemas embasados em *blockchain* podem contribuir significativamente para o aprimoramento de processos eleitorais, uma vez que os avanços tecnológicos facilitam o exercício do voto em nações democráticas. O exemplo de West Virginia permitiu que os militares em missões no exterior pudessem exercer o direito de voto de forma remota, sem a necessidade de deslocamento físico. Portanto, para o desenvolvimento de um sistema de votação eletrônica baseado em *blockchain*, pode-se recorrer a ferramentas já mencionadas como o *Hyperledger Fabric* e os contratos inteligentes em *Solidity* e *Chaincode*, integrando-os a uma interface amigável construída com a biblioteca ReactJS.

4. Materiais e Métodos

Neste capítulo, exploraremos a fundo a arquitetura do sistema, englobando todos os seus componentes essenciais. Vamos detalhar a implementação da rede *blockchain* usando *Hyperledger Fabric* e descrever como a arquitetura está integrada ao sistema, incluindo tanto as camadas de *backend* quanto de *frontend*. Para fornecer uma visão clara e completa, a (Tabela 1) apresenta um inventário detalhado das tecnologias empregadas, juntamente com suas respectivas versões, que são fundamentais para a construção eficaz do sistema.

Tecnologia	**Versão**
Hyperledger Fabric	2.5
NodeJS	18
ReactJS	18.2.0

Tabela 1. Versões de Tecnologia

Para criar uma rede blockchain para votação usando o Hyperledger Fabric, seguiríamos um processo estruturado focado na segurança, transparência e escalabilidade.

Começamos definindo as organizações participantes: cada estado ou região pode ser representado como uma organização separada na rede, com sua própria Autoridade de Certificação (CA) para gerenciar as identidades digitais dos eleitores e funcionários eleitorais. Cada organização terá vários *peers*, que são os nós da rede. Haverá dois tipos de *peers*: *endorsing peers*, que executam e endossam as transações (neste caso, votos), e *committing peers*, que validam essas transações e as cometem no *ledger* distribuído.

Os nós de *orderer*, fundamentais para a criação de blocos e manutenção da ordem das transações, utilizariam o algoritmo de consenso Raft. Este algoritmo garante que os blocos de transações sejam consistentes em toda a rede e são criados de maneira eficiente e segura.

A rede será segmentada em canais privados, um para cada estado ou região, para garantir a privacidade dos votos. Esses canais permitem a comunicação segura entre os membros específicos (ou seja, os nós de uma determinada região) e isolam os dados de votação de cada região.

O contrato inteligente (*chaincode*) será implementado em todos os *endorsing peers*. Este *chaincode* é responsável por registrar os votos e garantir que eles atendam aos critérios definidos, como unicidade e validade do eleitor. O *chaincode* também pode incluir lógica para a contagem de votos e a verificação de resultados.

O *ledger* será configurado para armazenar de forma imutável todos os votos emitidos, juntamente com um banco de dados, como o CouchDB[1], para gerenciar o estado atual da votação em cada região.

Para exemplo, em uma região ou estado vamos utilizar duas organizações, X e Y, onde X possui um componente do tipo *orderer* e dois componentes do tipo *peer*, e Y tendo apenas dois componentes do tipo *peer* (Figura 1).

[1]Banco de dados da Apache voltado para *web* que armazena os dados como objetos JSON

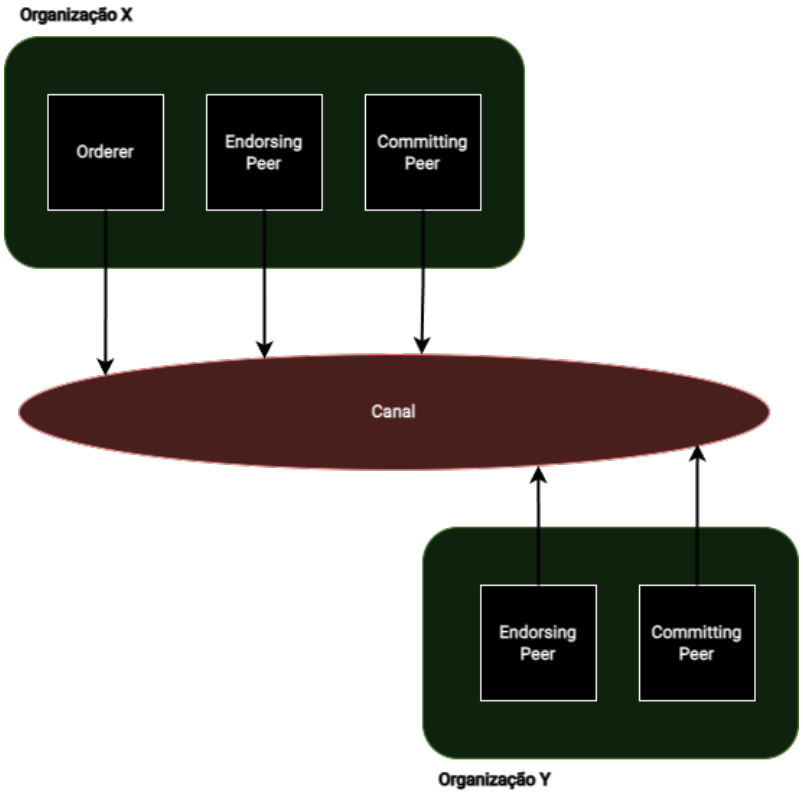

Figura 1. Arquitetura da Rede

Finalmente, a rede será integrada a aplicações de *frontend* e *backend*, permitindo que os eleitores registrem seus votos de maneira segura e conveniente. Estas aplicações se comunicarão com a rede blockchain através de SDKs fornecidos pelo Hyperledger Fabric.

O *backend* do sistema de votação desempenha um papel crucial na gestão e no processamento dos dados. Este *backend* interage com a *blockchain* através de APIs fornecidas pelo Hyperledger Fabric. Ele é responsável por funções como autenticação e autorização de eleitores, gerenciamento de sessões de votação e a agregação e processamento de dados de votação. O *backend* também atua como uma ponte entre a interface do usuário (*frontend*) e a *blockchain*.

O *frontend* do sistema é a interface através da qual os eleitores interagem com o sistema. Desenvolvido como uma aplicação web ou móvel, o *frontend* oferece uma interface de usuário amigável para autenticação (possivelmente utilizando biometria ou credenciais seguras), registro de votos e visualização de informações relacionadas às eleições. Este *frontend* comunica-se com o *backend* para enviar e receber informações, que são então processadas e armazenadas na *blockchain* (Figura 2).

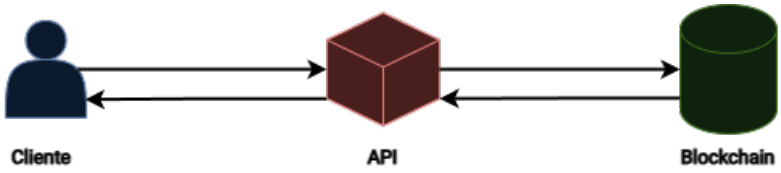

Figura 2. Estrutura da Aplicação

A integração entre *frontend*, *backend* e *blockchain* é crucial para a eficácia do sistema. Quando um eleitor emite seu voto através do *frontend*, o *backend* autentica o eleitor, assegurando que cada eleitor possa votar apenas uma vez. Uma vez autenticado e autorizado, o voto é enviado para a *blockchain*. Aqui, o contrato inteligente (*chaincode*) no Hyperledger Fabric processa o voto, realizando verificações de conformidade antes de gravá-lo no *ledger*.

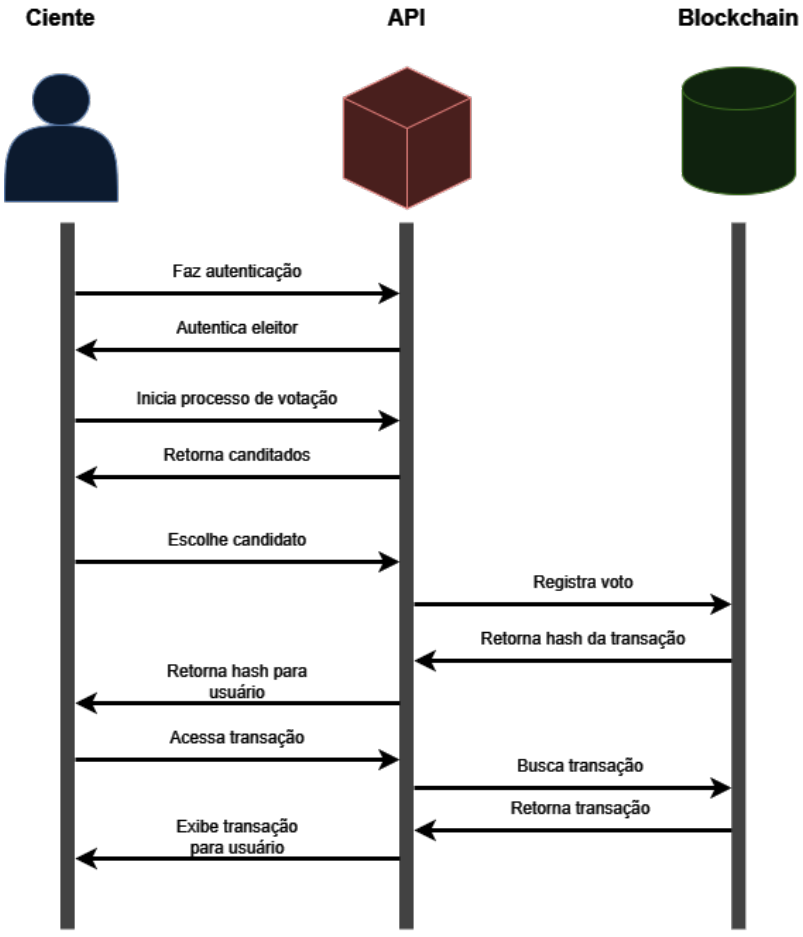

Figura 3. Diagrama Sequencial

Esta arquitetura garante que o processo de votação seja transparente, seguro e auditável, mantendo a confiança e a integridade do processo eleitoral (Figura 3).

5. Resultados

Como resultado, foi desenvolvido um contrato modelo, porém contendo a funcionalidade necessária para a execução de uma eleição. Está disponível no apêndice 1, o código do contrato na íntegra.

O código possui uma estrutura de dados denominada `Candidato` que armazena as informações de um candidato da eleição, com seu identificador (que pode ser o número de campanha do candidato, por exemplo), o nome e a quantidade de votos recebidos.

É essencial que o contrato armazene o proprietário para que somente a entidade responsável pela implementação do contrato possa realizar algumas tarefas como adicionar, editar e remover candidatos.

O código contém dois mapeamentos (ou *mappings*): o `chaveCandidato` que associa cada identificador ao respectivo candidato armazenado na blockchain; e o `votos` que vincula cada endereço do eleitor ao identificador do candidato escolhido, enquanto uma variável define a validade da votação, podendo ser alterada posteriormente.

No construtor do contrato, executado durante a implementação, define-se o proprietário e inicia-se uma eleição com o número de candidatos zerado.

Existem modificadores que executam funções mediantes determinadas condições. O `somenteDentroPrazo` executa a função apenas quando a votação não tiver expirado. O `somenteDono` permite que apenas o proprietário execute funções marcadas com este modificador. O `somenteNaoVotou` verifica se o eleitor já votou na eleição atual do contrato, executando as funções apenas se não o fez.

Foram desenvolvidas funções básicas para realizar uma eleição simples, permitindo apuração e visualização do voto pelo eleitor. Há funções de administração para adicionar/remover candidatos e, para o eleitor, votar e verificar o próprio voto. A apuração pode ser feita a qualquer momento pela função `apurar()`, retornando a lista de candidatos com seus respectivos votos.

O teste foi realizado na plataforma Remix[1]. Para implementar, seleciona-se uma conta e clica-se em *deploy* (Figura 4). Em seguida, é possível invocar funções e ler variáveis por meio da interface do próprio Remix (Figura 5). O proprietário adiciona os candidatos (Figura 6), nesse caso foram adicionados os candidatos "João Fulano" com o número 12 e "José Beltrano" com o número 34. Ao trocar de usuário, o eleitor registra o voto, inserindo o número do candidato, como na urna eletrônica (Figura 7).

[1]Remix. Disponível em: https://remix.ethereum.org/. Acesso em: 03 dez. 2023.

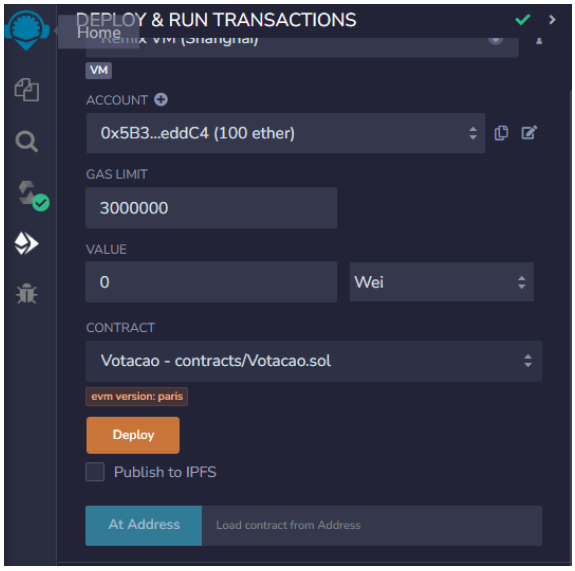

Figura 4. Fazendo o *deploy* do contrato no Remix

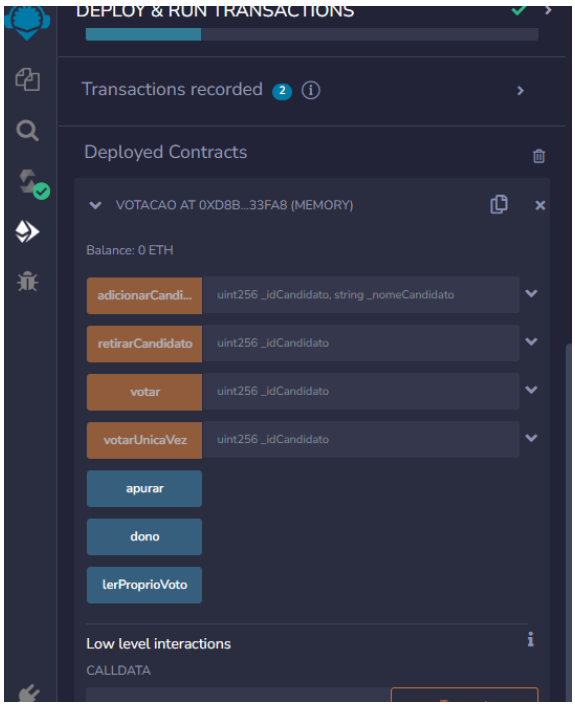

Figura 5. Interface do contrato no Remix

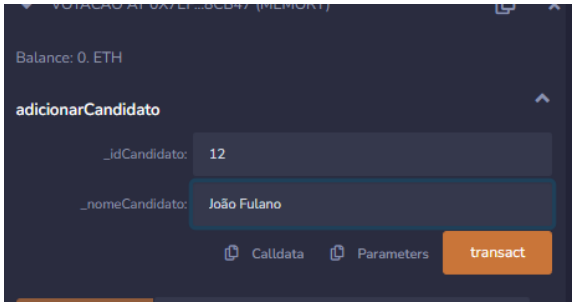

Figura 6. Adição de candidatos no contrato

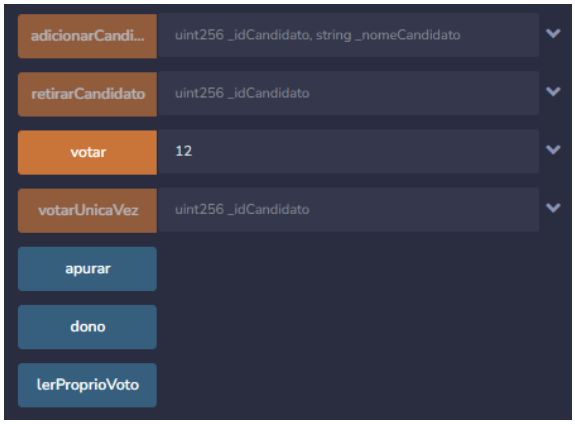

Figura 7. Votação no candidato de número 12

Depois, o eleitor pode conferir o candidato que votou escolhendo a função lerProprioVoto() e pode mudar de voto quantas vezes for necessário durante o pleito (Figuras 8 e 9). Esta ação anula o voto anterior, mas não o exclui da blockchain, e decrementa o voto do candidato escolhido anteriormente para incrementar a contagem do que escolheu atualmente, isso é verificado, por exemplo, fazendo a apuração (Figura 10)

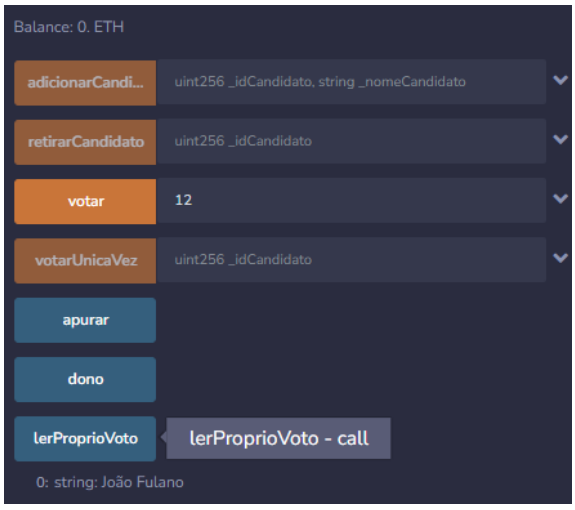

Figura 8. Leitura para conferência do voto

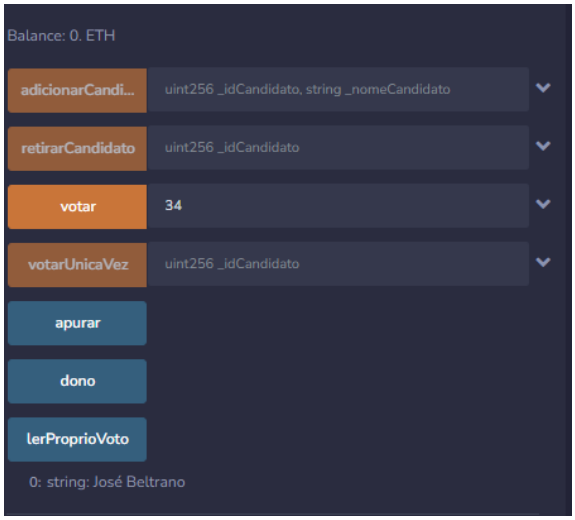

Figura 9. Mudança de voto para o candidato 34

Na apuração do voto, verifica-se na Array que o 1-José Beltrano, possui 1 voto que foi o voto transferido de João Fulano quando o eleitor mudou seu voto

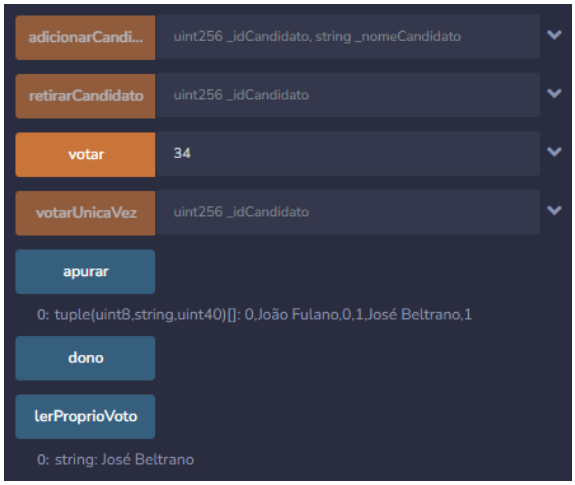

Figura 10. Apuração do voto logo no início

Por fim, feita uma votação mais ampla com 14 usuários, a apuração pode ser feita, ainda que o pleito não tenha terminado (Figura 11)

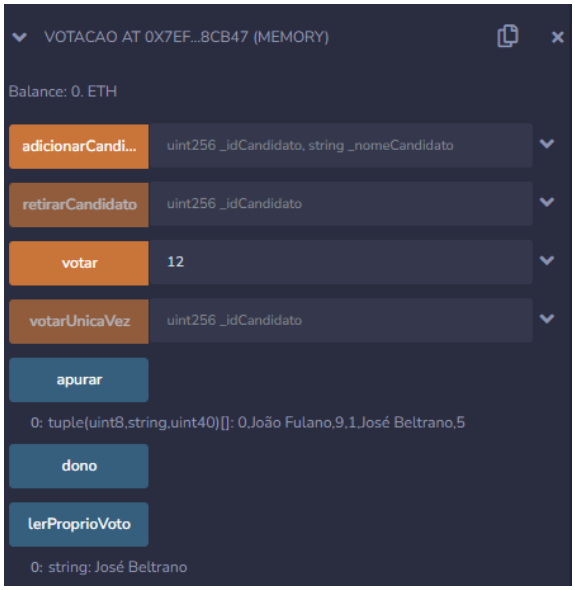

Figura 11. Apuração completa ao final

Quando a votação encerra, não é mais possível registrar voto, aparecendo a seguinte mensagem no backend que a transação não pode ser feita pois “O pleito já foi encerrado” (Figura 12)

Figura 12. Eleitor não pode votar após o fim do pleito

6. Conclusão

Verificou-se que é possível realizar uma eleição de forma totalmente digital através da implementação de contratos inteligentes em blockchains de forma descentralizada, possibilitando auditoria em tempo real. O contrato inteligente desenvolvido se comportou conforme o esperado dentro do planejado, porém, como não foi possível acessar uma rede com grande número de nós votando simultaneamente, não puderam ser observadas limitações de performance e escalabilidade.

O contrato não permite um segundo turno nem distinção regional de candidatos, como ocorre nas eleições brasileiras. Também não foi implementada identificação do eleitor, o que poderia ser feito posteriormente com a integração da autenticação governamental já utilizados em diversos sistemas do governo digital no Gov BR, ou de outra forma que seja mais conveniente para quem for implementar futuramente.

Os objetivos foram alcançados, com exceção da análise comparativa entre os sistemas eleitorais atuais e o proposto. Foi demonstrada a viabilidade técnica por meio de trabalhos relacionados e casos como o da Estônia e de West Virginia, os quais são digitais e o último usou blockchain no processo. Espera-se que este trabalho contribua com o

corpo de conhecimento sobre as aplicações de blockchain para o uso em eleições servindo de base para futuros estudos e pesquisas na área.

Trabalhos futuros poderão aprimorar o presente estudo, avaliando aspectos de escalabilidade e realizando testes abrangentes para verificar a confirmação simultânea de votos de grande quantidade de eleitores na *blockchain*. Adicionalmente, recomenda-se o desenvolvimento de contratos inteligentes mais robustos, capazes de modelar diferentes tipos de pleitos eleitorais. Para estimular o interesse do poder público no sistema de votação baseado em *blockchain*, sugere-se a realização de debates que evidenciem as qualidades inerentes à tecnologia blockchain, comparando-a à forma atual de votação por urna eletrônica. Pretende-se, dessa forma, contribuir para a difusão de soluções criptográficas seguras e transparentes aplicadas ao processo eleitoral.

7. Apêndice 1 - Código do Contrato

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.7;


contract Votacao {
    struct Candidato {
        uint8 id;
        string nome;
        uint40 qtdVotos;
    }


    address public dono;
    uint8 qtdCandidatos;
    Candidato[] candidatos;
    mapping(uint => Candidato) public chaveCandidato;
    mapping(address => uint) public votos;
    uint validade = block.timestamp + 1 minutes;


    constructor() {
        dono = msg.sender;
        qtdCandidatos = 0;
    }


    modifier  somenteDentroPrazo() {
        require(
            block.timestamp < validade,
            unicode'O pleito já foi encerrado'
        );
        _;
    }
```

```
modifier somenteDono() {
    require(
        msg.sender == dono,
        unicode"Função RESTRITA -
        Você não é o dono do contrato!"
    );
    _;
}


modifier somenteNaoVotou() {
    require(
        chaveCandidato[votos[msg.sender]].
        qtdVotos == 0,
        unicode"O eleitor já votou"
    );
    _;
}


function adicionarCandidato(
    uint _idCandidato,
    string memory _nomeCandidato
    ) public somenteDono{

    Candidato memory candidato =
        Candidato(qtdCandidatos, _nomeCandidato, 0);
    chaveCandidato[_idCandidato] = candidato;
    candidatos.push(candidato);
    qtdCandidatos++;
}


function retirarCandidato(uint _idCandidato)
public somenteDono {
    delete chaveCandidato[_idCandidato];
}


function votar(uint _idCandidato) public
somenteDentroPrazo{
    if (votos[msg.sender] != 0) {
        Candidato storage candidatoAnterior =
        chaveCandidato[votos[msg.sender]];
```

```
            candidatoAnterior.qtdVotos--;
            candidatos[candidatoAnterior.id].qtdVotos--;
        }
        Candidato storage novoCandidato =
        chaveCandidato[_idCandidato];
        novoCandidato.qtdVotos++;
        candidatos[novoCandidato.id].qtdVotos++;
        votos[msg.sender] = _idCandidato;
    }


    function votarUnicaVez(uint _idCandidato)
    public somenteNaoVotou somenteDentroPrazo {
        votar(_idCandidato);
    }


    function lerProprioVoto() public view returns
    (string memory) {
        return chaveCandidato[votos[msg.sender]].nome;
    }


    function apurar() public view returns
    (Candidato[] memory) {
        return candidatos;
    }
}
```

8. Referências

ABROL, A. *What Is Peer To Peer Network, And How Does It Work? [UPDATED]*. 2023. Disponível em: ⟨https://www.blockchain-council.org/blockchain/peer-to-peer-network/⟩. Acesso em: 17 nov. 2023.

BLASCO, A. *Tipos de consenso em blockchain: PoW, PoS e PoH. Lemon Wiki*. 2023. Disponível em: ⟨https://wiki.lemon.me/pt-br/crypto-101/tipos-de-consenso-blockchain-pow-pos-poh/\#Protocolo_de_consenso_Proof_of_Stake_PoS⟩. Acesso em: 17 nov. 2023.

BOMBIG, M. *Hash: a impressão digital dos dados. Exame*. 2023. Disponível em: ⟨https://exame.com/future-of-money/hash-a-impressao-digital-dos-dados/⟩. Acesso em: 17 nov. 2023.

COULOURIS, G.; DOLLIN, J.; HARIDAS, J. *Sistemas distribuídos: conceitos e projetos*. [S.l.]: Pearson Education do Brasil, 2011.

EHIN, P. e. a. Internet voting in estonia 2005–2019: Evidence from eleven elections. *Government Information Quarterly*, v. 39, n. 4, oct 2022.

Hyperledger Fabric. *Hyperledger Fabric Documentation*. 2023. Disponível em: ⟨https://hyperledger-fabric.readthedocs.io/en/latest/blockchain.html⟩. Acesso em: 17 nov. 2023.

IDEA, I. *ICTs in Elections Database — International IDEA*. 2023. Disponível em: ⟨https://www.idea.int/data-tools/data/icts-elections-database⟩. Acesso em: 17 nov. 2023.

MACHARIA, W. *Cryptographic Hash Functions*. 2021. Disponível em: ⟨https://www.researchgate.net/publication/351837904_Cryptographic_Hash_Functions⟩. Acesso em: 17 nov. 2023.

MARTINHO, L. L.; JABOUR, F. C. Sistema de eleições desenvolvido com a tecnologia de contratos inteligentes baseado em blockchain. *Seminérios de Trabalho de Conclusão de Curso do Bacharelado em Sistemas de Informação*, v. 5, n. 1, 2019.

MOORE, L.; SAWHNEY, N. *The West Virginia Mobile Voting Pilot: Under the Hood*. [S.l.], 2019. Disponível em: ⟨https://online.pucrs.br/blog/public/criptomoedas-bitcoin-blockchain-todo-mundo-ouve-falar-mas-poucos-entendem⟩.

PUC-RS. *Criptomoedas, Bitcoin e Blockchain: todo mundo ouve falar, mas poucos entendem*. 2023. Disponível em: ⟨https://online.pucrs.br/blog/public/criptomoedas-bitcoin-blockchain-todo-mundo-ouve-falar-mas-poucos-entendem⟩. Acesso em: 17 nov. 2023.

REBELLO, G. A. F. e. a. *Correntes de Blocos: Algoritmos de Consenso e Implementação na Plataforma Hyperledger Fabric*. [S.l.], 2019. Disponível em: ⟨https://www.gta.ufrj.br/ftp/gta/TechReports/RCS19c.pdf⟩. Acesso em: 17 nov. 2023.

RIBEIRO, L.; MENDIZABAL, O. *Introdução à Blockchain e Contratos Inteligentes: Apostila para Iniciante Relatório Técnico do INE*. [S.l.], 2021. Disponível em: ⟨https://repositorio.ufsc.br/bitstream/handle/123456789/221495/RT-INE2021-1.pdf⟩. Acesso em: 17 nov. 2023.

ROCHA, R.; LIMA, G. Proposta pragmática de contratos inteligentes no desenvolvimento de coleções: uma abordagem orientada à blockchain. *REVISTA IBERO-AMERICANA DE CIÊNCIA DA INFORMAÇÃO*, v. 16, n. 1, 27 mar. 2023.

SOLIDITY. *Solidity — Solidity 0.8.23 documentation*. 2023. Disponível em: ⟨https://docs.soliditylang.org/en/v0.8.23/⟩. Acesso em: 17 nov. 2023.

SUFRÁGIO. 2023. Disponível em: ⟨https://www.infopedia.pt/dicionarios/lingua-portuguesa/sufr\%C3\%A1gio/⟩. Acesso em: 13 nov. 2023.

TANENBAUM, A. S.; STEEN, M. V. *Sistemas distribuídos: princípios e paradigmas*. 2. ed. Porto Alegre: Bookman, 2008.

TRE-MT. *Evolução do Voto*. 2023. Disponível em: ⟨https://www.tre-mt.jus.br/institucional/memoria-eleitoral/evolucao-do-voto⟩. Acesso em: 14 nov. 2023.

TSE. *Eleições pelo Mundo: sistema de votação digital é realidade na Estônia*. 2022. Disponível em: ⟨https://www.tse.jus.br/comunicacao/noticias/2021/Novembro/eleicoes-pelo-mundo-sistema-de-votacao-digital-e-realidade-na-estonia⟩. Acesso em: 13 nov. 2023.

UFRJ. *O que é Blockchain*. 2023. Disponível em: ⟨https://www.gta.ufrj.br/ensino/eel878/redes1-2018-1/trabalhos-vf/blockchain/whatis.html⟩. Acesso em: 17 nov. 2023.