MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21c/2019/10.31.16.54-TDI

UM SISTEMA DE VOTO ELETRÔNICO UTILIZANDO A BLOCKCHAIN

Henrique Niwa

Dissertação de Mestrado do Curso de Pós-Graduação em Computação Aplicada, orientada pelo Dr. Celso Luiz Mendes, aprovada em 20 de setembro de 2019.

URL do documento original:
<http://urlib.net/8JMKD3MGP3W34R/3UB3GUB>

INPE
São José dos Campos
2019

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE
Gabinete do Diretor (GBDIR)
Serviço de Informação e Documentação (SESID)
CEP 12.227-010
São José dos Campos - SP - Brasil
Tel.:(012) 3208-6923/7348
E-mail: pubtc@inpe.br

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21c/2019/10.31.16.54-TDI

UM SISTEMA DE VOTO ELETRÔNICO UTILIZANDO A BLOCKCHAIN

Henrique Niwa

Dissertação de Mestrado do Curso de Pós-Graduação em Computação Aplicada, orientada pelo Dr. Celso Luiz Mendes, aprovada em 20 de setembro de 2019.

URL do documento original:
<http://urlib.net/8JMKD3MGP3W34R/3UB3GUB>

INPE
São José dos Campos
2019

Dados Internacionais de Catalogação na Publicação (CIP)

Niwa, Henrique.

n649s Um sistema de voto eletrônico utilizando a blockchain / Henrique Niwa. – São José dos Campos : INPE, 2019.

xxii + 84 p. ; (sid.inpe.br/mtc-m21c/2019/10.31.16.54-TDI)

Dissertação (Mestrado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2019.

Orientador : Dr. Celso Luiz Mendes.

1. Voto eletrônico. 2. Blockchain. 3. Bitcoin. 4. Ethereum. I.Título.

CDU 342.843.5:004.422.63

Aluno (a): *Henrique Niwa*

Título: **"UM SISTEMA DE VOTO ELETRÔNICO UTILIZANDO A BLOCKCHAIN"**

Aprovado (a) pela Banca Examinadora em cumprimento ao requisito exigido para obtenção do Título de *Mestre* em *Computação Aplicada*

Dr. Stephan Stephany

Presidente / INPE / SJCampos - SP

( ) Participação por Video - Conferência

(X) Aprovado *( ) Reprovado*

Dr. Celso Luiz Mendes

Orientador(a) / INPE / São José dos Campos - SP

( ) Participação por Video - Conferência

(X) Aprovado *( ) Reprovado*

Dr. Nandamudi Lankalapalli Vijaykumar

Membro da Banca / INPE / SJCampos - SP

( ) Participação por Video - Conferência

(X) Aprovado *( ) Reprovado*

Dr. Clóvis Torres Fernandes

Convidado(a) / ITA / SJCampos - SP

( ) Participação por Video - Conferência

(X) Aprovado *( ) Reprovado*

Este trabalho foi aprovado por:

( ) maioria simples

(X) unanimidade

São José dos Campos, 20 de setembro de 2019

"O rio corta a rocha, não por causa de sua força, mas por causa da sua persistência.".

JAMES N. WATKINS

A todos que me apoiaram e acreditaram, em especial minha ***família e amigos****.*

AGRADECIMENTOS

Agradeço pelo financiamento do CNPq, infraestrutura da CAP/INPE e paciência dos professores e coordenação.

RESUMO

O sistema de voto com urnas eletrônicas utilizado no Brasil é o maior utilizado no mundo, nenhum outro país com número de habitantes semelhante ou maior possui uma eleição cuja execução e apuração se realizem no mesmo dia. O problema é a falta de transparência e auditabilidade das urnas utilizadas, a verificação depende de técnicos do TSE e dos consultores convidados. Não há a possibilidade de uma auditoria completa no sistema. Este trabalho implementa uma arquitetura baseada em *blockchain* com protocolos de segurança e criptografia, distribuída geograficamente e com tempo de execução e apuração dos resultados semelhante ao atual. Isto permitiu que o sistema seja altamente verificável e auditável, além de permitir ao eleitor conferir seu voto e verificar que faz parte do total.

Palavras-chave: Voto eletrônico. Blockchain. Bitcoin. Ethereum.

AN ELECTRONIC VOTING SYSTEM BASED ON BLOCKCHAIN

ABSTRACT

The voting system with electronic ballot box used in Brazil is the largest in the world, no other country with a similar or larger number of inhabitants has an election whose execution and counting take place on the same day. The problem is the lack of transparency and auditability of the ballot boxes used, the verification depends on TSE technicians and guest consultants. There is no possibility of a full system audit. This work implements a blockchain based architecture with security and encryption protocols, geographically distributed and with similar execution time and results. This allowed the system to be highly verifiable and auditable, and allowed the voter to check their vote and verify that it is part of the total.

LISTA DE FIGURAS

Pág.

2.1 Arquitetura do algoritmo Civitas. 10
2.2 Estrutura de blocos em corrente. 14
2.3 Conceitos do blockchain. 15
2.4 Verificação de participação de hash em bloco. 17
2.5 Exemplo de transação. 18
2.6 Exemplo de transação com múltiplas saídas. 18
2.7 Exemplo de transação com múltiplas entradas. 19

3.1 Quantidade de transações diárias do *bitcoin* e *Ethereum*.(16/05/2019) . . 24
3.2 Mediana da taxa das transações diárias do *bitcoin* e *Ethereum*.(16/05/2019) 25

4.1 Modelagem de um voto como transação. 28
4.2 Nodos Leves/Clientes. 29
4.3 Autenticação, distribuição dos candidatos e direcionamento. 29
4.4 Troca de blocos e transações. 30
4.5 Arquitetura do sistema. 30
4.6 Valor de mercado do *Ethereum* do *bitcoin*. 31
4.7 Valor individual do *Ethereum* e do *bitcoin*. 32
4.8 Mediana do valor das transações diárias do *bitcoin* e *Ethereum*. 33
4.9 Endereços ativos diários do *bitcoin* e *Ethereum*. 34
4.10 Tempo de geração dos blocos no *blockchain* do *bitcoin*. 35
4.11 Cliente - Login. 45
4.12 Cliente - Presidentes. 46
4.13 Processo de auditoria. 47
4.14 QR Code verificador. 48
4.15 Visualizador *btc*. 48
4.16 Visualizador *blockchair*. 49
4.17 Visualizador *blockchain*. 50

5.1 Votos válidos de 2018. 58
5.2 Parcela de eleitores que poderiam votar no sistema proposto. 59
5.3 Formato das transações para distribuição das cédulas. 60
5.4 Quantidade de transações ao longo do tempo. 61
5.5 Consumo de memória RAM ao longo do tempo. 62
5.6 Histograma do número de transações por bloco.(50 milhões de votos) . . 65

6.1 Áreas com aplicações para o blockchain. 70

6.2 Modelagem de um sistema de confiança utilizando a *blockchain*. 72

LISTA DE TABELAS

Pág.

2.1 Um livro razão exemplo 13
2.2 *Blockchains* públicas, privadas e federativas 22

5.1 Tempo de criação de transferência para dois destinatários 57
5.2 Tempo de assinatura de transferência para dois destinatários 57
5.3 Tempo de envio de uma transferência para dois destinatários 57
5.4 Tempo para simulação da distribuição dos votos 60
5.5 Tempo para simulação da execução dos votos 61
5.6 Tempo de execução de cada voto 62
5.7 Tempo para apuração de 50 milhões de votos 63
5.8 Tempo para apuração com diferentes populações 64
5.9 Tempo para apuração de 100 milhões de votos 64
5.10 Tamanho das bases de dados 66
5.11 Comparativo entre o sistema atual no Brasil e utilizando *blockchain* . . . 68

LISTA DE CÓDIGOS

Pág.

2.1 Exemplo de bloco 15
2.2 Exemplo de *scriptPubKey* 19
2.3 Exemplo de *scriptSig* 19

4.1 Exemplo de mensagem assinada digitalmente 41
4.2 Pseudocódigo da apuração dos votos 44
4.3 Identificador da transação validada 47
4.4 Dados de uma transação pelo seu identificador 52
4.5 Dados de um bloco 53

SUMÁRIO

Pág.

1 INTRODUÇÃO 1
1.1 Motivação 1
1.2 Estrutura do documento 6

2 FUNDAMENTAÇÃO TEÓRICA 9
2.1 Protocolos de votação 9
2.2 Estrutura do Bitcoin 12
2.2.1 Bitcoin: Onde tudo começou 12
2.2.2 Sobre o blockchain 13
2.2.3 Encadeamento de transações 16
2.2.4 Tabela de transações utilizáveis 16
2.2.5 Transações 17
2.2.6 Prova de trabalho 20
2.2.7 Consenso 21
2.2.8 Consumo energético do bitcoin 21
2.2.9 Tipos de blockchain 22

3 REVISÃO BIBLIOGRÁFICA 23
3.1 Avaliando propostas de voto com *blockchain* já existentes 23
3.2 Avaliando projetos de *blockchain* existentes 25

4 ARQUITETURA PROPOSTA 27
4.1 Justificativa da escolha do *blockchain* para este trabalho 31
4.2 Utilizando permissões de acesso 34
4.3 Criando moedas não nativas dentro do blockchain 36
4.4 Possíveis problemas 37
4.4.1 Propaganda externa 37
4.4.2 Distribuição dos clientes 37
4.4.3 Fraude da eleição pelo governo 37
4.4.4 Sybil attack 38
4.4.5 Consenso bizantino 38
4.4.6 Transação duplicada 38
4.4.7 Criação de votos aleatórios (interferência externa) 38

4.4.8 Inclusão de blocos aleatórios . . . 39
4.4.9 Brute-force de votos . . . 39
4.4.10 Criptografia . . . 39
4.4.11 Tamanho do blockchain . . . 39
4.4.12 Segurança . . . 40
4.4.13 Verificação das transações . . . 41
4.5 Tecnologia . . . 42
4.6 Objetivos . . . 48

5 RESULTADOS EXPERIMENTAIS . . . 55
5.1 Ambiente e testes realizados . . . 55
5.2 Testes com computação centralizada . . . 56
5.3 Testes com computação descentralizada . . . 56
5.4 Testes com compilação estática e com bibliotecas . . . 57
5.5 Simulando processo eleitoral . . . 58
5.6 Distribuição . . . 59
5.7 Simulando os votos . . . 61
5.8 Apuração . . . 62
5.9 Auditoria . . . 65
5.10 Análise dos resultados . . . 66

6 OUTRAS APLICAÇÕES POSSÍVEIS . . . 69
6.1 Banco de dados Amazônia . . . 69
6.2 Blockchain CubeSat . . . 69
6.3 Blockchain Vant Swarm . . . 71
6.4 Blockchain IoT Sensors . . . 71
6.5 Blockchain Web-Of-Trust . . . 71

7 CONCLUSÃO . . . 73
7.1 Trabalhos futuros . . . 74

REFERÊNCIAS BIBLIOGRÁFICAS . . . 77

1 INTRODUÇÃO

Existem diferentes meios de votação pelo mundo, o mais comum e simples é o que se utiliza de cédulas de votação, um processo que consiste no eleitor ir a um centro designado, criar a marcação de sua preferência sendo observado por um grupo de auditores, mas com o voto ainda secreto e depositar sua ficha de papel em uma urna. Esses votos então são agregados de todos os diferentes centros de votação e posteriormente validados e contabilizados se utilizando de métodos manuais e automáticos. Para uma pequena quantidade de votantes é um meio simples e razoavelmente rápido de votar e contar, porém para grandes populações há um grande trabalho a ser feito e portanto levam-se dias para finalizar o evento. Também existe o risco de que contagens erradas, fraude nas cédulas e urnas de votação e ausência de eleitores atrapalhe e/ou mude o resultado.

No Brasil temos um sistema de voto onde se utilizam urnas eletrônicas, que fazem a contagem e contabilidade local dos votos. Para se ter uma apuração total, é necessário que as unidades de memória de cada máquina sejam enviadas para um local central, onde é realizada a leitura e agregação de votos. Essas unidades de memória e as máquinas em si necessitam de segurança para que os dados não sejam modificados por partes maliciosas interessadas.

O presente trabalho tem por objetivo criar um voto eletrônico seguro e eficiente, descentralizado e totalmente auditável, que possa ser usado em diferentes dispositivos e situações. Um sistema de voto precisa ser auditável e conferível pela população. Os requisitos foram de que equiparasse ao sistema utilizado no Brasil, utilizando no mínimo os mesmos números de eleitores, tempo de apuração e votação. A segurança das cédulas e conferência são requisitos superiores ao atual.

1.1 Motivação

Segundo Brunazo Filho (2014) há três gerações de urnas eletrônicas. O sistema atual de voto eletrônico do Brasil consiste em urnas eletrônicas que realizam a gravação dos dados de forma digital, também chamadas de primeira geração ou DRE (*Direct Recording Electronic voting machine* - máquina de gravação eletrônica direta do voto) , podendo ser conferidas apenas com a participação do administrador do sistema e do desenvolvedor do software. Existe uma segunda geração, proposta por Mercuri (2001) sobre a impressão de um comprovante do voto, possibilitando a auditoria contábil da votação, chamado IVVR (*Independent Voter Verifiable Record* - Registro Independente Conferível pelo Eleitor). No Brasil é comum ser chamado

de "Voto Impresso Conferível pelo Eleitor", ou VICE(BRUNAZO FILHO, 2014). Existe ainda uma terceira geração de sistemas eleitorais, os quais contam com RFID ou chips de identificação por rádio-frequência. Estes possibilitam a conferência do voto pelo eleitor independente de software e facilitam uma auditoria independente, sendo chamados "*End-to-End verifiability*"ou, E2E.

O sistema de urna eletrônica no Brasil tem as seguintes características[1]:

a) Local de armazenamento: Votos são gravados eletronicamente localmente dentro da urna.

b) Corrupção: Votos podem ser corrompidos em hardware (FERNANDES, 2006).

c) Localidade do voto: Necessidade de comparecimento a uma zona eleitoral.

d) Auditoria: É realizada por grupos convidados pelos órgãos competentes, sendo feita exclusivamente no software e apenas no processo de voto, não na contabilização; uma auditoria em tempo real em época de eleição não é possível.

e) Verificação: Um eleitor não pode verificar se seu voto foi computado corretamente. Mesmo que haja a impressão de um comprovante da escolha feita, o eleitor não tem como verificar que seu voto foi incluso no total. A contagem teria de ser feita utilizando todos os comprovantes. Os custos são estimados em 1,8 bilhão de reais(MARTINS, 2018). A proposta de emenda a constituição 135/2019 que propõe o voto impresso, diz que cédulas físicas devem ser expedidas para conferência do eleitor e que sejam depositadas em urnas para conferência.

f) Privacidade: No processo de voto atual, a privacidade do eleitor é garantida apenas pela câmara de votação.

Segundo Brunazo Filho (2001) encontramos a seguinte avaliação da urna eletrônica brasileira:

> "Integridade do Sistema - REPROVADO: [...] os programas das unidades eleitorais foram modificados depois de terem sidos apresentados aos auditores externos e não foram reapresentados para análise depois disso."(BRUNAZO FILHO, 2001)

[1] http://www.tse.jus.br/eleicoes/urna-eletronica/seguranca-da-urna

“Integridade e Confiabilidade dos dados - INCONCLUSIVO são permitidos testes apenas em máquinas previamente preparadas para o teste e não em máquinas prontas para votar.”(BRUNAZO FILHO, 2001)

“Anonimato do Eleitor - SEM GARANTIAS: A associação entre o voto e a identidade do eleitor é uma hipótese não afastada, visto que o número do eleitor é fornecido à urna eletrônica no mesmo momento em que ele digita seu voto e não se permite análise completa dos programas.”(BRUNAZO FILHO, 2001)

“Autenticação do Operador – REPROVADO: Em todos os terminais da rede do TSE, inclusive os que contem os códigos-fonte dos aplicativos e das bibliotecas padrão dos compiladores, está instalado um programa de manutenção remota, o PCAnyWhere, que possui "porta-dos-fundos"para acesso remoto. O Sistema SiS, de controle de acesso, também possui portas para manutenção de emergência.”(BRUNAZO FILHO, 2001)

“Auditabilidade do Sistema – INCONCLUSIVO: Fiscais de partidos de Boa Vista, RO, relataram testes de dupla carga dos programas nas urnas eletrônicas da 1ª Zona Eleitoral. Os arquivos de log destas urnas não revelam o fato, levando a crer que a carga dos programas na urna eletrônica apaga o log existente anteriormente, perdendo registros úteis para auditoria que poderiam revelar uma fraude. Considera-se este tópico inconclusivo, pois não se obteve uma confirmação formal do teste feito na presença do juiz e do promotor público.”(BRUNAZO FILHO, 2001)

“Transparência do Sistema – REPROVADO: como demonstrado no item "Autenticação do Operador".”(BRUNAZO FILHO, 2001)

“Disponibilidade do Sistema – NÃO ANALISADO: o TSE, no entanto, afirma ter criado defesas contra ataques DoS na sua rede durante o período da totalização dos resultados.”(BRUNAZO FILHO, 2001)

> “Confiabilidade do Sistema – APRESENTA FRAGILIDADES: nas unidades eleitorais os votos de cada candidato são constantemente mantidos em memória temporária, RAM, sem dígito de verificação. A cada voto novo, o acumulado é regravado em memória permanente, Flash Card, sem verificação de integridade (novo valor = valor anterior + 1). Assim, uma eventual troca indevida em um bit na RAM se propaga.”(BRUNAZO FILHO, 2001)

> “Facilidade de Uso – APROVADO COM RESTRIÇÕES: para um eleitor acostumado a teclados e computadores a urna eletrônica é fácil de usar, para outros nem tanto. Muitas reclamações de "foto de candidato errado", que surgiram em quase todos os municípios, foram devidas a dificuldades de uso da UE. O TSE não divulgou nenhuma estatística sobre a frequência deste problema, que não foi pequena.”(BRUNAZO FILHO, 2001)

> “Documentação e segurança – INCONCLUSIVO: os fiscais dos partidos não tiveram acesso a toda a documentação, inclusive não foram apresentadas as alterações no código-fonte nem as justificativas destas alterações intempestivas feitas depois destes códigos terem sido apresentados aos fiscais.”(BRUNAZO FILHO, 2001)

> "Integridade do Pessoal – NÃO AVALIADO: é difícil avaliar a "incorruptibilidade"de pessoas. Relata-se que houveram muitas denúncias de falhas da segurança na guarda física das urnas eletrônicas, com desvios e roubos.”(BRUNAZO FILHO, 2001)

Apesar do relatório ter sido realizado em 2001, houveram incidentes recentes como em 2017(BATISTA, 2018) e abaixo:

> “O Caso Diadema, SP - 2000: A análise desses arquivos revelou que todas as urnas eletrônicas tinham sido carregadas fora da cerimônia oficial de carga e lacração, dias antes da convocação por edital público, tendo todas ficado sem lacres durante dias. A grande maioria das urnas eletrônicas utilizadas - 431 de 451 - foram inseminadas com o software de votação nos dias 22 e 23 de setembro, 2 em 24/9, 7 em 25/9, 2 em 26/9, sendo que todas

> elas só foram lacradas no dia 28/9. Esses dados mostravam que a totalidade das urnas eletrônicas de Diadema em 2000, estiveram carregadas com os programas mas sem lacre e sem a presença de fiscais dos partidos políticos por vários dias." (CUNHA et al., 2014, 24)
>
> "O Caso Marília, SP - 2004: Em auditoria, os Arquivos de Espelhos de Boletins de Urna da 400º Zona Eleitoral indicavam que muitas seções eleitorais tiveram seus resultados recebidos para apuração antes do início da votação." (CUNHA et al., 2014, 26)
>
> "O Caso Alagoas - 2006: Diversas irregularidades nos arquivos gerados pelas urnas foram detectadas por auditores externos. Frente às evidências, o administrador negou acesso aos arquivos solicitados pelos auditores e transferiu ao requerente uma cobrança antecipada no valor de R$ 2 milhões para que fosse desenvolvida uma perícia das urnas. Diante do não pagamento do valor proibitivo, o requerente foi multado e condenado por litigância de má-fé." (CUNHA et al., 2014, 30)
>
> "O Caso Itajaí, SC - 2008: Nenhuma urna preparada para a votação passou pelo teste obrigatório prescrito pelo Art. 32 da Res. TSE 22.712/08. Um caso foi o da 97ª Zona Eleitoral onde a urna da seção 236 que foi sorteada para o teste obrigatório foi substituída por outra na hora do teste, preparada exclusivamente para este fim. A urna que foi utilizada para o teste foi posteriormente colocada à parte e recarregada, procedimento que destruiu eventuais provas nela gravadas." (CUNHA et al., 2014, 34)

A declaração universal dos direitos humanos[2], no artigo 21.3 define que todos tem o direito de participar de seu governo, direta ou indiretamente; o voto é universal e deve ser mantido em segredo.

Diante disto e dos problemas anteriores, definimos voto seguro como um procedimento que proteja a privacidade do eleitor e a integridade da eleição. O voto deste sistema é criptografado. As informações de origem, destino e quantidade são calculadas para o valor de sua *hash*. A eleição é feita de modo que os votos se concatenem

[2] https://www.un.org/en/universal-declaration-human-rights/

em forma de árvores e blocos lógicos, a mudança de uma informação inviabiliza todas as posteriores. A integridade da eleição é garantida pelo fato de que todos os softwares são abertos e apuração de milhões de votos pode ser auditada posteriormente por todos, para garantir que o resultado divulgado é o das urnas. No sistema brasileiro atual, não há como garantir isto, pois a apuração não é aberta e não há possibilidade de auditoria posterior, um sistema de voto impresso também poderia ter sua apuração violada e manipulada.

A eficiência deste trabalho está na velocidade de execução e apuração dos votos. A auditoria também é eficaz pelo fato de que pode ser feita em tempo real, posteriormente o banco de dados também pode ser distribuído sem a necessidade de recursos especiais.

O sistema atual é centralizado no sentido que todos os votos são contados e agrupados centralmente, onde o governo divulga o resultado. Neste sistema há a possibilidade de que um ou mais órgãos auditores acompanhem em tempo real o resultado, estes independem de localização.

Periodicamente o Tribunal Superior Eleitoral abre edital para convocar especialistas para o Teste Público de Segurança[3], estes resultados não são divulgados em sua íntegra. Os especialistas convocados realizam a auditoria em uma sala especial dentro do TSE, o código-fonte e hardware não deixam as dependências e o tempo de análise é limitado. Neste sistema todo o código é aberto e ao fim das eleições, o banco de dados pode ser auditado e contabilizado, de modo que a apuração oficial seja conferida por todos.

A acessibilidade deste sistema poderia afetar 20% da população, vide Seção5.5. São eleitores que deixam de votar e acabam tendo de justificar.

1.2 Estrutura do documento

O texto desta dissertação esta organizado da seguinte forma; no capítulo 2 são abordados protocolos de votação eletrônicos, uma descrição do *blockchain* e como foi feita a modelagem de uma cédula eleitoral no formato de uma transação. O capítulo 3 aborda a metodologia e filosofia utilizada no desenvolvimento deste trabalho. O capítulo 4 trata a arquitetura proposta e tecnologias utilizadas. O capítulo 5 apresenta os testes feitos e resultados obtidos, além de descrever todo o ferramental necessário. O capítulo 6 trata de usos variados do *blockchain* na indústria e em áreas que

[3] http://www.justicaeleitoral.jus.br/tps/

poderiam ser úteis dentro do INPE. Por fim, no capítulo 7 temos a conclusão final e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Protocolos de votação

Existem alguns estudos propondo protocolos de votação e consenso, como o Helios(ADIDA, 2008), um protocolo baseado na web criado de forma a ser um framework para votações, onde qualquer parte pode iniciar uma nova eleição e todos podem auditar. Este algoritmo já foi testado e estudado dentro da UFMG com certo sucesso[1](GRAAF, 2017, 86), porém a maior votação já realizada com ele utilizou de 8000 cédulas(ALONSO et al., 2018).

O Helios(ADIDA, 2008) consiste das seguintes etapas:

a) Um eleitor qualquer prepara seus votos, podendo mudar suas escolhas até a hora da submissão.

b) Submete o voto para um quadro de avisos online, onde todos podem ver seu voto criptografado e sua identificação.

c) Ao fim do período de votação, o sistema descriptografa os votos, mistura a identificação e faz a contagem, reembaralhando ao fim. Isso é feito para manter a privacidade dos votos.

d) Um auditor pode conferir se o embaralhamento dos votos foi feito de maneira eficiente por um certo período antes da divulgação dos resultados.

e) Resultados são descriptografados e postados.

Outro algoritmo de votação é o Civitas(CLARKSON et al., 2008) cuja arquitetura temos na Figura 2.1

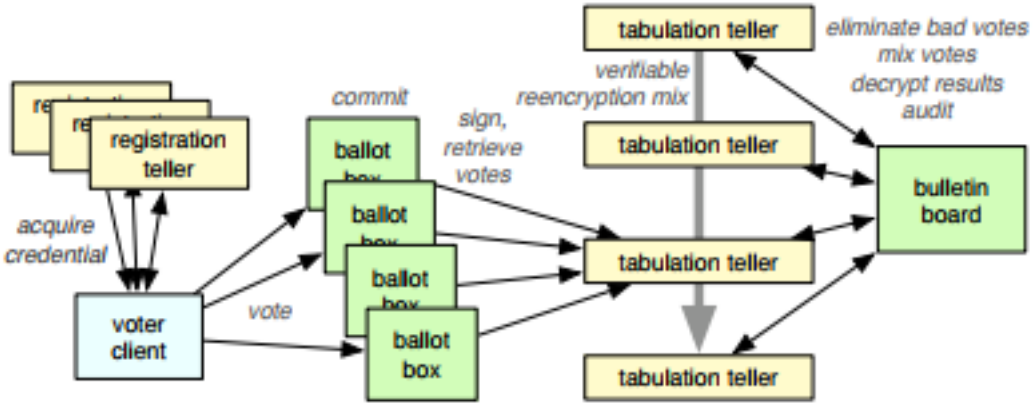
O algoritmo do Civitas(CLARKSON et al., 2008) precisa de uma preparação inicial que é:

a) Supervisor cria uma nova votação de um quadro de avisos virtual. Também identifica todos os agentes com suas chaves públicas de comunicação.

b) Em seguida, o supervisor posta todos os eleitores participantes e suas chaves públicas, que são duas, uma para registro e outra para designação do voto, sua participação na eleição.

[1] `https://votacoes.inscrypt.dcc.ufmg.br`

Figura 2.1 - Arquitetura do algoritmo Civitas.



Fonte: Clarkson et al. (2008).

c) Os agentes de tabulação criam uma chave pública para um esquema de encriptação distribuída e postam no quadro virtual. Essa chave requer a participação de todos os agentes para descriptografar mensagens.

d) Finalmente os agentes de registro criam credenciais para todos os eleitores, as quais são pares de chaves público-privadas em uma criptografia assimétrica. As chaves públicas são postadas no quadro e as privadas distribuídas aos respectivos eleitores.

A fase de voto consiste dos seguintes passos:

a) O eleitor se registra para receber sua credencial de voto com um agente de registro. Com a chave de designação de voto, o eleitor recebe o voto do agente de registro, como se recebesse uma cédula.

b) Eleitor pode votar quantas vezes quiser, mas no fim deve indicar qual voto é o final.

Fase de tabulação:

a) Agentes de tabulação recuperam os votos e as chaves públicas do quadro virtual.

b) Essas chaves são usadas para verificar os votos. É como se o voto fosse uma mensagem assinada pelo eleitor utilizando sua chave privada, esse voto pode ser verificado por qualquer um com acesso à chave pública da pessoa.

c) Os votos são embaralhados, para manter a natureza anônima do eleitor.

d) Os resultados são descriptografados e postados no quadro virtual, preservando a identificação dos eleitores.

Em testes de desempenho, segundo Clarkson et al. (2008), foi verificado que uma máquina com custo equivalente na época a US$ 1500 poderia realizar o cálculo e tabulação de voto para 500 eleitores em 5 horas, com um custo estimado em US$ 12 por eleitor, mas cujo custo seria amortizado em sucessivas eleições.

Em outro artigo, Kulyk et al. (2014) propõe um outro protocolo de voto, levando em conta os dois algoritmos anteriores (Helios (ADIDA, 2008) e Civita (CLARKSON et al., 2008)) e introduzindo algumas características, tais como:

a) Flexibilidade no tipo de votação (sim/não, múltiplos candidatos, prioridade);

b) Flexibilidade no eleitorado (mudança na lista de eleitores);

c) Rapidez na preparação da eleição;

d) Acesso remoto;

e) Usabilidade;

f) Eficiência na aferição dos votos.

A segurança é garantida por requisitos como:

a) Elegibilidade (apenas eleitores autorizados podem fazer parte);

b) Cada voto é computado apenas uma vez;

c) Voto secreto;

d) Mudança de voto é permitida até o voto final;

e) Resultados gerais não podem ser vistos até o fim da eleição;

f) Verificação;

g) Robustez para pequenas falhas.

Desde 2005 a população da Estônia utiliza um sistema de voto eletrônico auxiliar ao tradicional, onde na última eleição 30% da população utilizou deste meio. Foi organizado de forma que o eleitor possa fazer sua escolha através de um computador pessoal e posteriormente checando seu voto através do *smartphone*, essa separação de dispositivos traz uma segurança na verificação individual dos votos pela população.(LEETARU, 2017)(SOLVAK, 2016)

Segundo Anandaraj et al. (2015) e Grewal et al. (2015) a biometria deve ser utilizada na identificação e aumento na segurança dos eleitores. Algoritmos para embaralhamento da identidade dos eleitores (BENALOH, 2006) existem, porém precisariam ser adaptados ao *blockchain*. A geração das chaves privadas precisa ser segura, caso contrário pode-se descobrir através da chave pública, como demonstrado por Pierrick Gaudry nas eleições russas de 2019 (WRAY, 2019). Por causa de erros ao se tentar implementar sua própria criptografia é importante utilizar protocolos e implementações testadas pela indústria, como o código do *bitcoin*.

2.2 Estrutura do Bitcoin

2.2.1 Bitcoin: Onde tudo começou

A moeda eletrônica *Bitcoin*(NAKAMOTO, 2008) popularizou o *blockchain*, utilizando o conceito de consenso entre os participantes, ou seja, a transação precisa ser homologada, ou aceita por um número mínimo de participantes antes de ser inclusa. Além disso tanto as transações como os blocos são organizados de forma sequencial, criando uma corrente de blocos em que as transações estão organizadas na forma de árvore. As transações utilizam criptografia de chaves público/privada. A chave pública é utilizada nas comunicações do mundo para o indivíduo, usada como endereço do destinatário, apenas o usuário que possua a chave pública usada como destino poderá usar esta transação, para tanto usando sua chave privada para assinar transações.

As transações são acumuladas em filas de prioridade, que levam em conta o tamanho das transações, as taxas pagas e a ordem de recepção. É calculada uma *hash* em cima das transações do bloco e do cabeçalho do bloco anterior, em conjunto com um valor que pode ser modificado, para produzir uma *hash*. Esta *hash* final precisa atender determinado nível de dificuldade, esse valor que pode ser alterado é chamado de *nonce* e o objetivo, no bitcoin, é encontrar uma *hash* final com uma determinada

dificuldade, que aumenta ou diminui de acordo com o poder de computação da rede, quanto mais computadores estiverem participando, maior será a dificuldade, de forma que a geração de novos blocos seja em um intervalo de 10 minutos. Esse é o principal trunfo do *bitcoin*, baseado no algoritmo Hashcalc(BACK, 2002), ele define que um bloco só deve ser incluso caso sua tenha o número mínimo de 0's à esquerda em sua *hash*, pela natureza não determinística do algoritmo, é necessário um trabalho de força-bruta, em que todos os dados são modificados até que se atinja o resultado. Há um campo no cabeçalho específico para isso, o *Nonce*, um número de 32 bits que permite 4294967296 possibilidades. Muitas vezes mesmo percorrendo todas as variações do *Nonce* a *hash* ainda assim não é atingida, nesse caso mais campos do bloco podem modificados, por exemplo o campo que indica a data de criação do bloco, porém de modo que a hora de criação no bloco não indique futuro maior do que algumas horas e também que tenha sido criado antes do bloco anterior. Por último pode-se mudar a ordem das transações, excluindo ou trocando com a fila de espera.

Vários computadores ao redor do mundo estão permanentemente conectados à internet recebendo as transações e gerando novos blocos, eles competem entre si para a maior geração de blocos possíveis, a maior cada cadeia é escolhida como a principal e replicada, todos que estavam trabalhando em uma cadeia alternativa precisam conferir quais transações foram armazenadas e iniciar uma nova busca com o restante. Essa rede distribuída garante que não haja uma autoridade principal e a verificação de que cada bloco atende a dificuldade mínima é muito mais rápida do que a criação, desse modo todos podem verificar a validade de um bloco.

2.2.2 Sobre o blockchain

O *blockchain* é um conceito análogo a um livro-razão em que todas as transações de entrada e saída devem ser escritas em ordem histórica, com o saldo dessas transações anotado. A última informação sempre será a mais atual e levará em conta o histórico.

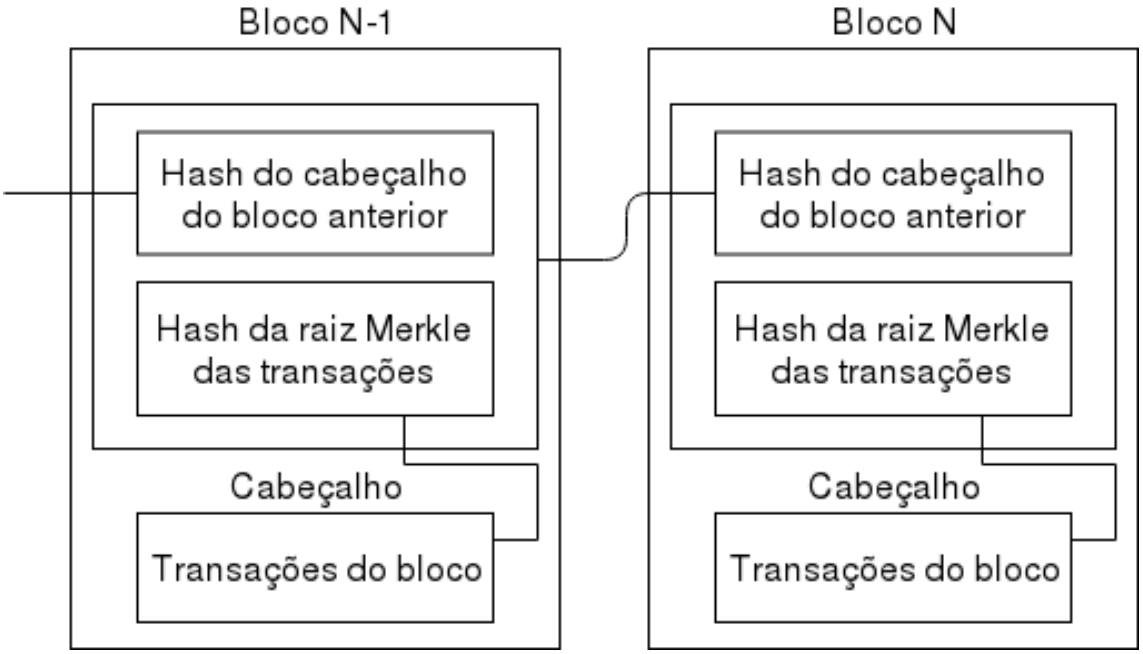
Tabela 2.1 - Um livro razão exemplo

Índice	Remetente	Valor	Destinatário
1	Renan	10	Matheus
2	Ana	50	Matheus
3	Matheus	60	Joana

Fonte: Autor.

A última entrada no livro razão (Tabela 2.1) só foi possível pelo fato de Matheus ter tido duas entradas anteriores como destinatário. Com o uso do *blockchain* se impede que as informações gravadas possam ser revertidas ou adulteradas, pois são assinadas digitalmente por uma *hash* e há um encadeamento dessas informações, criando uma corrente de blocos (*blockchain*), como ilustrado na Figura 2.2, um cabeçalho de um bloco pode ser visto na Listagem 2.1.

Figura 2.2 - Estrutura de blocos em corrente.



Fonte: Autor.

O *blockchain* se apoia em alguns conceitos como visto na Figura 2.3.

Alguns pontos são comuns a todas as blockchains, segundo Shelkovnikov (2016):

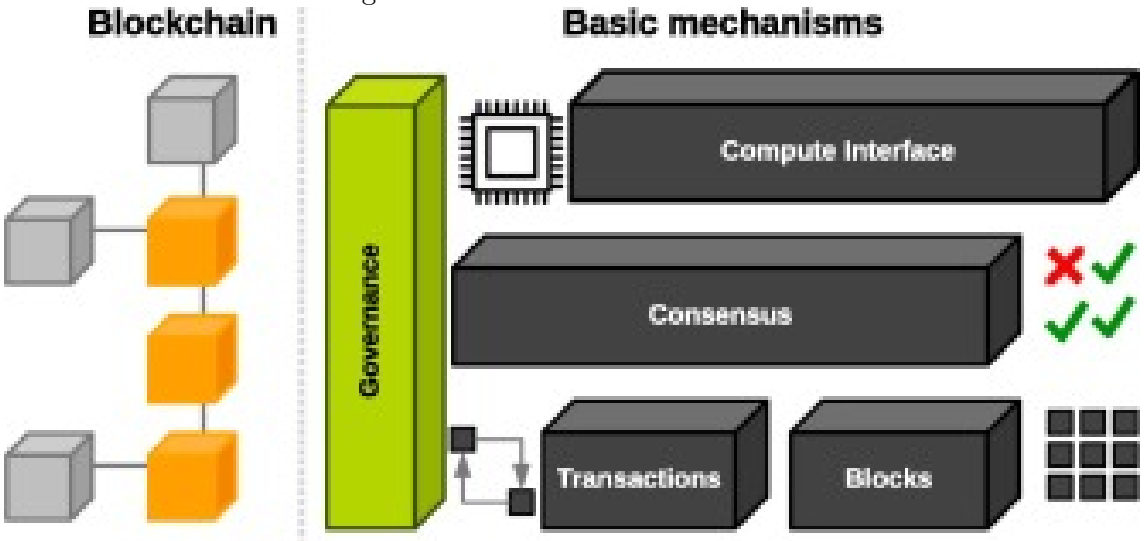
- Digitalmente distribuída entre um número de computadores em quase tempo real, ela é descentralizada e uma cópia de todos os registros esta disponível a todos os participantes da rede ponto-a-ponto.
- O consenso é atingido com a participação de vários usuários da rede
- A identidade é provada através de criptografia e assinaturas digitais
- Existem mecanismos que tornam difícil, mas não impossível a alteração de

Listagem 2.1 - Exemplo de bloco

```
{
    "hash" : "12c6514d14970a71...",
    "miner" : "1DWxicms4s6R3o6...",
    "confirmations" : 1,
    "size" : 482490421,
    "height" : 13,
    "version" : 3,
    "merkleroot" : "4ea2a3c2df...",
    "tx number" : 1777167,
    "time" : 1557339970,
    "Date" : "05/08/2019 15:26:10",
    "nonce" : 3,
    "bits" : "207fffff",
    "difficulty" : 4.656542374e-10,
    "chainwork" : "00000000000...",
    "previousblockhash" : "6a32..."
}
```

Fonte: Autor.

Figura 2.3 - Conceitos do blockchain.



Fonte: Sultan et al. (2018).

dados históricos

- Os registros possuem dados sobre data e horário

- Programável, podem conter instruções que executam somente em certas condições e acompanhados de dados adicionais. (*Compute blocks* da Figura 2.3)

2.2.3 Encadeamento de transações

O encadeamento de *hash's* das transações, é caracterizado por uma árvore binária *Merkle tree*, dessa forma temos uma sequência de transações.

Um exemplo seria um bloco com função *hash* h, cabeçalho do bloco como b e transação t, Nb é o bloco N, Nt é a *hash Merkle* raiz, para um bloco com 4 transações a composição dessa árvore binária seria:

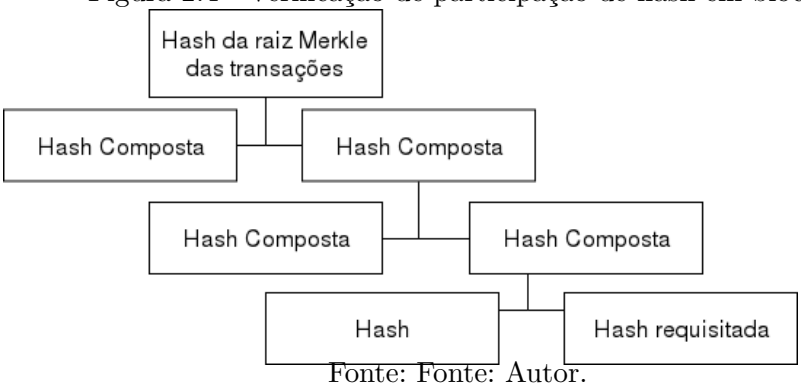
$$\begin{aligned} h(b(N)) &= h(b(N-1)) + h(t(Nt)) \\ h(t(Nt)) &= h(t(ab)) + h(t(cd)) \\ h(t(ab)) &= h(t(a)) + h(t(b)) \\ h(t(cd)) &= h(t(c)) + h(t(d)) \end{aligned}$$

Para provar que uma transação faz parte de um bloco basta requisitar a *hash* desta transação e as *hash's* que fazem parte da composição da *hash Merkle*, como visto na Figura 2.4. A *hash* composta é a *hash* das *hash's* abaixo. Isto faz com que a prova de que uma *hash* faz parte de um bloco seja proporcional a altura desta árvore, tendo como complexidade log n, com n sendo o número de *hash's*.

2.2.4 Tabela de transações utilizáveis

Verificar que uma *hash* faz parte de um bloco significa que ela faz parte da *blockchain*, caso ainda não tenha utilizada ela pode ser a entrada de outra transação. Uma tabela de transações que não foram usadas é chamada de UTXO (*Unspent transaction output* - Saída não gasta de transação), ao incluir uma transação na blockchain ela é retirada desta tabela. As saídas de novas transações são inclusas para poderem ser usadas posteriormente.

Figura 2.4 - Verificação de participação de hash em bloco.



Fonte: Fonte: Autor.

2.2.5 Transações

Cada transação consiste de uma ou múltiplas entradas e uma ou múltiplas saídas. Cada entrada foi uma saída anterior, como podem haver várias saídas é necessário indicar qual a numeração em que ela ocorreu. Uma transação tem o seguinte formato:

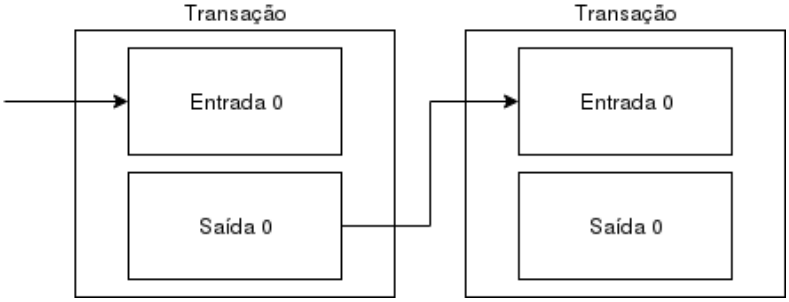
- Entradas: *hash* da transação de origem, índice de saída desta transação
- Saídas: endereço para saída, valor

Cada usuário, tanto quem recebe quanto envia, possui 3 dados importantes, seu endereço que é derivado a partir da chave pública, chave pública e chave privada. A criptografia utilizada é a *Elliptic Curve Digital Signature* (ECDSA), esse sistema deriva do algoritmo *Digital Signature Algorithm* (DSA). Ao enviar os dados utiliza-se em conjunto a chave privada do remetente e a chave pública do destinatário, o qual pode verificar que os dados foram assinados pelo remetente. As chaves utilizam uma função dupla de *hash* SHA256(Secure Hash Algorithm - Algoritmo de *hash* seguro), o endereço é a chave original com uma *hash SHA256* e depois uma RIPEMD160 (RIPE Message Digest - Resumo de mensagem RIPE).

No *blockchain* uma transação é criada e verificada atráves de uma linguagem de *script* semelhante a *Forth* e de funcionamento análogo a uma pilha de dados. Isto fica evidente no código fonte do bitcoin[2]. Para se enviar, cria-se um *script* utilizando uma transação de entrada que o autor seja o destinatário, uma quantidade da moeda nativa (igual ou menor da transação original) e o endereço do novo destinatário. Para

[2] https://github.com/bitcoin/bitcoin/blob/master/src/script/script.cpp

Figura 2.5 - Exemplo de transação.

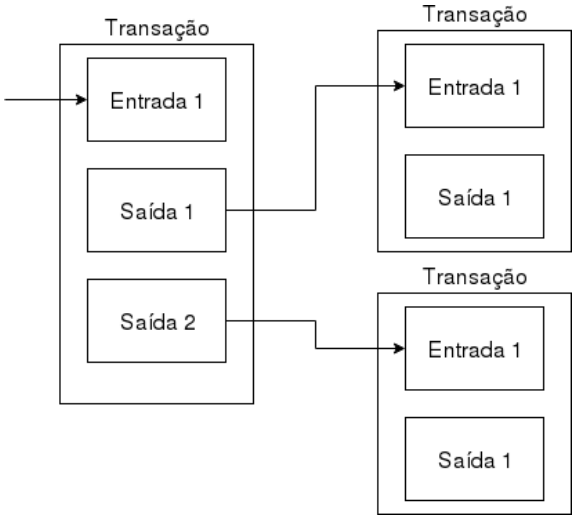


Fonte: Autor

utilizar esta transação como entrada o destinatário precisará validar sua própria chave, demonstrando que sua chave se traduz como endereço da transação.

A chave pública é usada para demonstrar que você é o dono do endereço para o qual uma transação foi enviada, a chave privada para assinar transações. A chave pública é derivada da chave privada, após isso, ambas as chaves passam por um processo duplo de *hash SHA256*, o endereço é a chave pública que ao invés de receber uma segunda *hash SHA256*, recebe uma *hash RIPEMD160*. Caso se perca a chave privada, se perde o acesso a todas as transações enviadas para o usuário. O duplo uso de *hash* dificulta a busca pela chave privada do eleitor, que poderia ser usada para forjar seu voto.

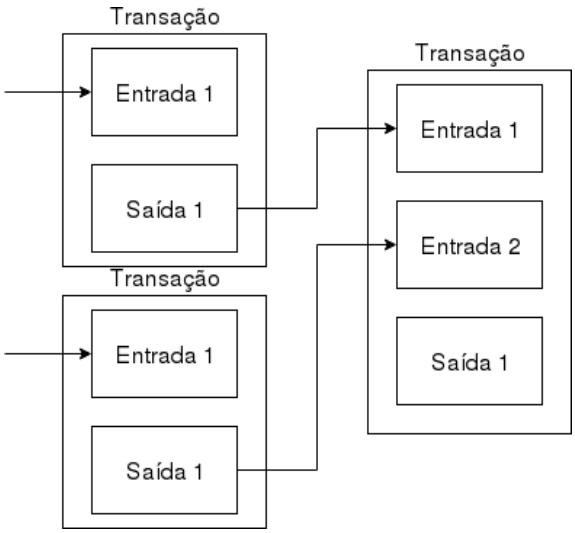
Figura 2.6 - Exemplo de transação com múltiplas saídas.



Fonte: Autor.

.

Figura 2.7 - Exemplo de transação com múltiplas entradas.



Fonte: Autor.

Como exemplo de uma transação, temos a *scriptPubKey* (Listagem 2.2), que apresenta o *script* que deve ser interpretado para utilização da transação.

Listagem 2.2 - Exemplo de *scriptPubKey*

```
1 OP_DUP OP_HASH160 dda93b6dd63e93bfa5d380136d4d73fa23e1e1cc
  ↪ OP_EQUALVERIFY OP_CHECKSIG
  ↪ 73706b716329fb4e9717caa77e92eab42ef48ab70400000000000000 OP_DROP
```

Fonte: Autor.

Como scriptSig pelos dados da mesma transação da Listagem 2.2 temos o exemplo na Listagem 2.3, que é a assinatura do bloco pelo remetente.

Listagem 2.3 - Exemplo de *scriptSig*

```
1 <sig> <pubKey> 304402204a9541c892ddd807af66a52d2512272372206e4244458
  ↪ e90731d6d6f58caee80022025e59d5448bcca6e708c6d38310a2f52df0e51e00
  ↪ 4c1e85b957abfe5375e343101
  ↪ 0237
2 e1c47aa687ee1e074f01c5da126e12fb3814c60581b5afd9f369d3faaf7d2f
```

Fonte: Autor.

Uma verificação dessa transação se processa da seguinte forma:

a) Concatena-se *scriptSig* e *scriptPubkey*.

b) Assinatura e chave pública (*scriptSig*) colocadas na pilha.

c) OP_DUP duplica último valor da pilha.

d) OP_HASH160 transforma a chave pública numa hash RIPEMD160

e) A constante <pubKeyHash> entra na pilha

f) OP_EQUALVERIFY compara <pubHashA> e <pubKeyHash>

g) OP_CHECKSIG compara se a assinatura <sig> foi feita usando a chave pública <pubKey>

h) <asset_data> entra na pilha

i) Os dados do asset são conferidos e retirados da pilha por OP_DROP

2.2.6 Prova de trabalho

O *bitcoin* possui um conceito fundamental que é a prova de trabalho(*proof-of-work*). Cada bloco para ser incluso precisa que sua *hash* tenha um determinado número de 0's à esquerda, esse número dita a dificuldade de geração de *bitcoins*.

O espaço amostral se modifica a cada novo bloco. Essa dificuldade é alterada pela algoritmo do *bitcoin* para se adequar à capacidade de processamento do sistema. Quanto maior o poder de processamento, maior será a dificuldade para encontrar novos blocos, desta forma o tempo de geração de blocos se mantém constante.

Existem outras criptomoedas com conceitos análogos para prevenir a criação de blocos aleatórios na cadeia de blocos, como o *Ethereum*(BUTERIN, 2015) que atualmente utiliza de prova de trabalho, mas que planeja migrar para *proof-of-stake*, prova de participação, onde o criador do próximo bloco não é aquele que possuir a cadeia mais longa de blocos, com a dificuldade correta de cada bloco, mas será escolhido através da combinação de tempo de vida na rede e recursos computacionais, ou seja o tamanho de sua participação na rede.

2.2.7 Consenso

Cada nodo da rede *bitcoin* possui uma versão completa da *blockchain* e ativamente calcula novos blocos para inclusão seguindo o conceito de prova de trabalho. O consenso entre os nodos de qual *blockchain* adotar é feito escolhendo-se a cadeia mais longa. Todos os nodos que possuíam blocos calculados e com transações fora dessa cadeia mais longa, precisam recomeçar tendo a nova cadeia como base. Assim as filas de transações precisam ser refeitas e verificadas quais transações precisam ser inseridas em novos blocos.

O consenso do sistema é utilizar a cadeia de blocos mais longa, em que os blocos sigam a progressão de terem suas *hash's* com zeros à esquerda em ordem crescente. Este mecanismo é facilmente verificável, basta computar a *hash* do cabeçalho do bloco, a árvore *Merkle* de transações, não sendo necessário modificar a árvore de transações ou o cabeçalho a fim de encontrar a *hash* de bloco correta.

2.2.8 Consumo energético do bitcoin

Devido ao consenso através da prova de trabalho, muito processamento é desperdiçado no *bitcoin*, prova disso são as estatísticas sobre o uso de energia segundo Vries (2019).

- Consumo anual estimado de energia (TWh): 73.12
- Receitas globais com a mineração de blocos: US$7,780,813,140
- Custos globais com a mineração de blocos: US$3,656,073,069
- País com consumo mais próximo do estimado da rede bitcoin: Austria
- Potência em Watts por GH/s: 0.103
- Potência da rede em PH/s (1,000,000 GH/s): 81,354
- Número de domicílios americanos que poderiam ser abastecidas com o consumo de energia da rede *Bitcoin*: 6,770,506
- Custo energético por transação (KWh): 610
- Número de dias que a energia de uma única transação sustentaria um domicílio americano: 20.63
- Porcentagem do consumo da rede bitcoin comparado ao global: 0.33%

Devido a este elevado consumo, na seção 4.2 explica-se qual mecanismo adotado ao invés da prova de trabalho.

2.2.9 Tipos de blockchain

Segundo Casino et al. (2019), a primeira geração de blockchains utilizava moedas nativas (primeira versão do *bitcoin* (NAKAMOTO, 2008)), a segunda geração introduziu o uso de *tokens*, *assets* ou moedas não-nativas e contratos inteligentes(*multichain, OpenAssets e Ethereum*(GREENSPAN, 2015)(BUTERIN, 2015)), a terceira geração fez o uso de permissões de acesso (*Hyperledger Fabric*).

Há ainda a distinção entre *blockchains* públicas, onde qualquer um pode participar e privadas, onde o acesso é controlado, ou seja existe uma unidade central de verificação e requer permissão para entrada de novos nodos. As redes federativas mantém o modelo de permissão mas a diferença é que um conjunto de nodos líderes é escolhido para verificar e delegar permissões dentro da rede, descentralizando a rede. Na Tabela 2.2 temos um comparativo das diferentes características. A propriedade de consenso significa a dificuldade necessária para criação dos blocos, o mecanismo quer dizer mediante a aprovação de qual entidade tem-se a aprovação da criação dos blocos. Identidade diz respeito à associação entre usuários e endereços na rede e quem pode fazer tal associação. Eficiência é sobre o desperdício de recursos computacionais. Imutabilidade refere-se à possibilidade de mudanças do histórico, ataques de colusão envolvem uma associação entre a maioria dos nodos (50% +1) para definir qual a cadeia de blocos deve ser utilizada. Propriedade diz sobre quem controla o sistema. Gerenciamento é a capacidade de criar permissões ou delegar. A aprovação é o tempo de espera para uma transação ser considerada parte do histórico, incluída na *blockchain*.

Tabela 2.2 - *Blockchains* públicas, privadas e federativas

Propriedade	Pública	Privada	Federativa
Consenso	Prova de trabalho pesado	Prova de trabalho leve	Prova de trabalho leve
Mecanismo(consenso)	Todos os nodos	Entidade central	Nodos líderes
Identidade	Pseudo anonimato	Usuários identificados	Usuários identificados
Eficiência do protocolo	Baixa	Alta	Alta
Consumo de energia	Alta	Baixa	Baixa
Imutabilidade	Impraticável	Ataque de colusão	Ataque de colusão
Propriedade	Pública	Centralizada	Semi-centralizada
Gerenciamento	Sem permissões	Com permissões	Nodos líderes permissionados
Aprovação	Em minutos	Em milisegundos	Em milisegundos

Fonte: Casino et al. (2019, 58)

3 REVISÃO BIBLIOGRÁFICA

A metodologia para desenvolver este trabalho foi:

- Criar alguns objetivos que deveriam ser atendidos (detalhados em 4.6).
- Procurar e comparar dentro da literatura trabalhos com sistemas de voto propostos e então procurar projetos *blockchain* disponíveis.
- Buscar por atualizações no conceito de *blockchain* como o uso de permissões e criação de moedas não-nativas.
- Por fim analisar os problemas encontrados e adotar soluções pertinentes para eles.

3.1 Avaliando propostas de voto com *blockchain* já existentes

Na literatura existem diversos trabalhos relacionados a voto e blockchain, a seguir apresentamos uma análise sobre cada um.

Em Bistarelli et al. (2017) foi criado um sistema a partir da *blockchain* pública do *bitcoin*. O problema de se utilizar a blockchain pública do bitcoin é o número de transações possíveis, o algoritmo do *bitcoin* foi criado de modo que para não haver inflação da moeda seja criado um novo bloco a cada 10 minutos (NAKAMOTO, 2008), cada bloco contém em média 2000 transações[1]. O número de votos possíveis por dia seria:

$$24 * 6 * 2000 = 288.000$$

Levando em conta que as transações que não pagam taxas podem nunca ser inclusas e a taxa média [2] estava em 1,7 US$ este sistema, utilizando o *Ethereum*, não seria viável para grandes quantidades de votos. Dessa forma os *blockchains* públicos do *bitcoin* e do *Ethereum*, não são dimensionáveis para votações de uma grande população. Seja pelo número de votos possíveis diários (Figura 3.1) ou pelas taxas de cada transação (Figura 3.2).

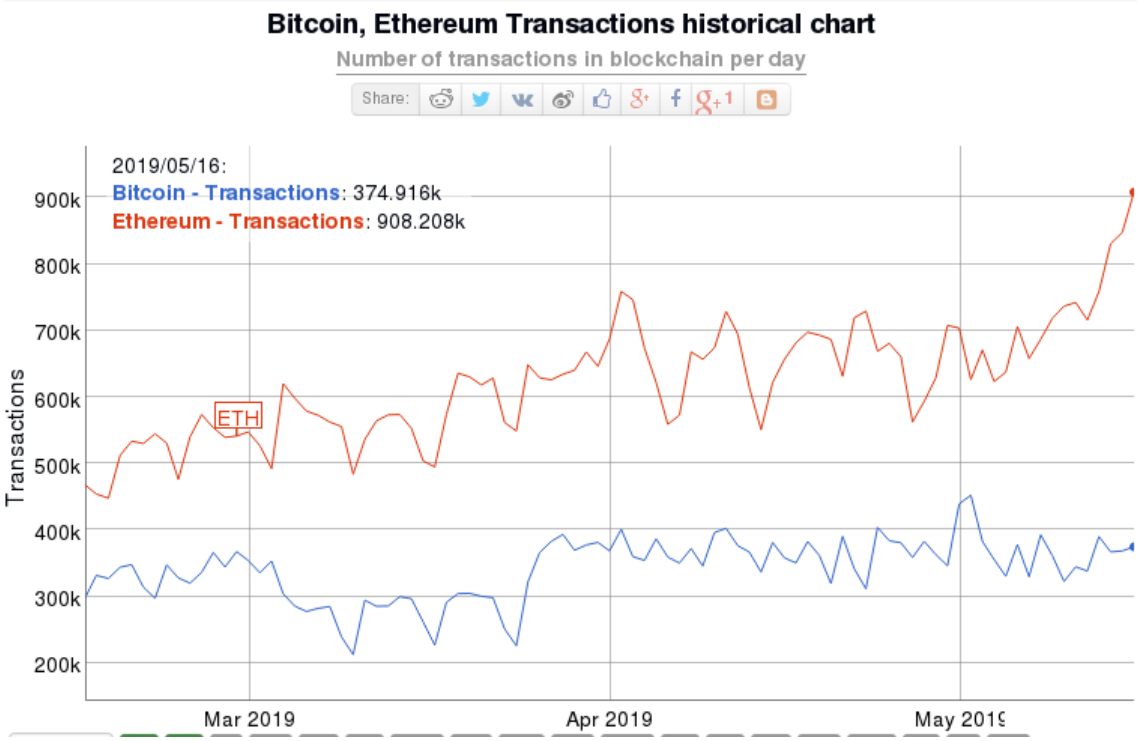
Segundo Hjalmarsson et al. (2018), Cooley et al. (2018), Shahzad e Crowcroft (2019), são tratados os diversos aspectos de infraestrutura. Em Wu e Yang (2018) temos uma proposta de topologia de sistema de votação. Em (ADIPUTRA et al., 2018) e (SINGH; CHATTERJEE, 2018) mencionam implementação porém não há resultados

[1] `https://outputs.today/` em 08/05/2019

[2] `https://bitinfocharts.com/comparison/bitcoin-transactionfees.html#3m` em 08/05/2019

quantitativos. Em (MUDLIAR et al., 2018) é discutido o uso de *blockchain* como registro nacional de identidade, este poderia ser utilizado em conjunto com um sistema de voto. Em (LUO et al., 2018) e (LEE; PARK, 2019) temos uma proposta de algoritmo de consenso para eleições. Alsunaidi e Alhaidari (2019) faz uma revisão de algoritmos de consenso. O assunto sobre voto sob coação é tratado em (ARAUJO et al., 2008), a privacidade do voto em (ZHANG et al., 2018). Os trabalhos (KHOURY et al., 2018), (YAVUZ et al., 2018), (ALENCAR; CORREA, 2017), (LAI et al., 2018) foram feitos com base no *blockchain Ethereum*, utilizando da plataforma pública e dos contratos inteligentes, porém não há resultados quantitativos. Sobre a viabilidade do *Ethereum*, temos que a escalabilidade da plataforma nas últimas semanas não ultrapassou 900 mil transações diárias[3] como pode ser visto na Figura 3.1. Apesar dos contratos inteligentes, cada transação não pode indicar o voto de mais do que um eleitor. Isto torna inviável o uso para grandes populações.

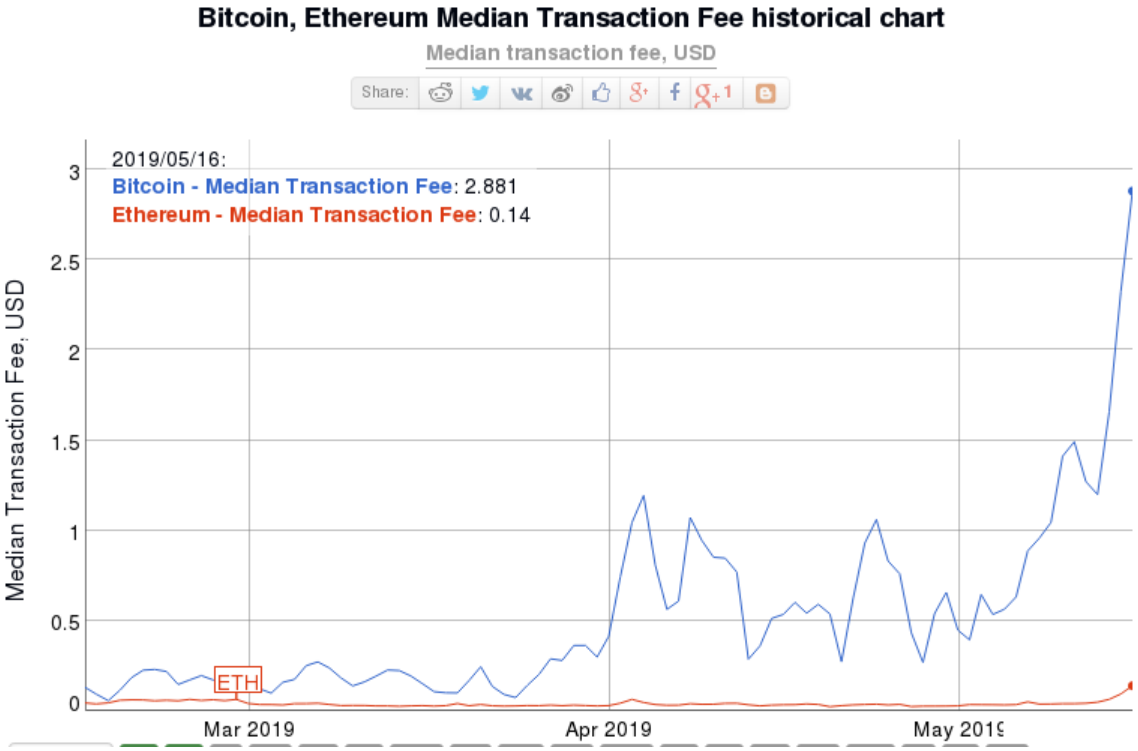
Figura 3.1 - Quantidade de transações diárias do *bitcoin* e *Ethereum*.(16/05/2019)



Fonte: BitInfoCharts (2019).

[3]https://etherscan.io/chart/tx em 05/2019

Figura 3.2 - Mediana da taxa das transações diárias do *bitcoin* e *Ethereum*.(16/05/2019)



Fonte: BitInfoCharts (2019).

3.2 Avaliando projetos de *blockchain* existentes

Dias (2016) publicou uma revisão sistemática de plataformas *blockchain* e em Dinh et al. (2017) comparativo de plataformas *Ethereum*.

Esta arquitetura poderia ter sido baseado em outro projeto *blockchain*, a seguir temos uma lista dos projetos atuais e na Seção 4.1 a justificativa da escolha.

- BigChainDB[4]: um sistema que partiu de bancos de dados distribuídos e depois adquiriu características do blockchain.
- Chain Core[5] + Stellar: Uma plataforma de blockchain como serviço, o uso de sua infraestrutura é paga.
- Corda[6]: uma *blockchain* desenvolvida para uso corporativo, com uma versão aberta.

[4]https://www.bigchaindb.com
[5]https://chain.com
[6]https://www.r3.com/corda-enterprise

- Credits[7]: uma plataforma de desenvolvimento corporativa.
- Elements Blockchain Platform[8]: extensões para o protocolo do *bitcoin*.
- Eris:db[9]: extensões para o protocolo do *bitcoin*.
- Ethereum[10]: uma plataforma descentralizada que permite o uso de contratos dentro de sua própria *blockchain*.
- Quorum[11]: uma plataforma corporativa baseada no *Ethereum*.
- Multichain[12]: uma extensão de código aberto criada a partir do código do *bitcoin*, projetada para transações financeiras de múltiplos ativos.
- Bitcoin[13]: Foi a criptomoeda inicial, criada em 2008, possui código livre, dezenas de desenvolvedores e milhões de usuários.
- Openchain[14]: uma plataforma corporativa para gerenciar ativos.
- HydraChain[15]: uma extensão do Ethereum para criação de blockchains com permissões.
- Hyperledger Fabric[16]: uma *blockchain* criada pela IBM, ela possui permissões e permite a execução de contratos inteligentes análogos ao *Ethereum*.

Saraf e Sabadra (2018) comparou *Hyperledger, Ethereum e Corda*, o primeiro é mais apropriado para desenvolvimento de algoritmos de consenso, o segundo onde o uso dos recursos dos mineradores é necessário e o terceiro em aplicações financeiras. Em um trabalho futuro poderia se utilizar do *Hyperledger* e com novas formas de consenso entre os nodos.

Ainda existem plataformas de desenvolvimento criadas por grandes empresas que visam facilitar o desenvolvimento, entretanto as soluções ficam atreladas àquela empresa. IBM[17], Oracle[18], Microsoft[19], Amazon[20] possuem serviços semelhantes, cujo uso requer infraestrutura paga.

[7] https://credits.com
[8] https://elementsproject.org
[9] https://erisindustries.com
[10] https://www.ethereum.org
[11] https://www.goquorum.com
[12] https://www.multichain.com
[13] https://bitcoin.org
[14] https://www.openchain.org/
[15] https://github.com/HydraChain/hydrachain
[16] https://www.hyperledger.org/projects/fabric
[17] https://www.ibm.com/blockchain
[18] https://www.oracle.com/br/cloud/blockchain/
[19] https://azure.microsoft.com/en-us/services/blockchain-service/
[20] https://aws.amazon.com/pt/blockchain/

4 ARQUITETURA PROPOSTA

Para implementar o sistema de voto proposto neste estudo, foi utilizada uma transação com somente uma entrada, que é a distribuição inicial. Votos extras poderiam ser expedidos, de forma que caso haja uma inclusão de novas categorias estas sejam possíveis com a arquitetura proposta. Deste modo haveria duas entradas, o eleitor teria 2 cédulas para usar.

As transações são usadas como cédulas eleitorais, cada uma representa um voto que pode ser fracionado. Na distribuição cada eleitor recebe 10 unidades, podendo distribuí-las entre os candidatos, exemplo:
Candidato A recebe 90%
Candidato B recebe 10%

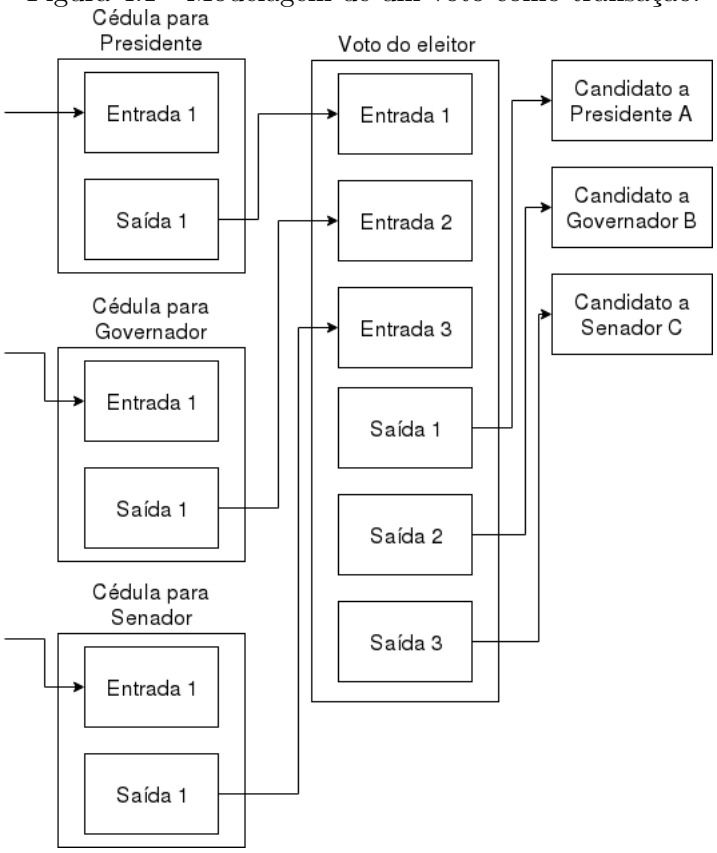
Estas 10 unidades servem para que a eleição tenha granularidade de 0.1 votos, poderia se modificar para 100 unidades e assim uma porcentagem de 0 a 100 para cada candidato.

Extrapolando para várias categorias de candidatos, podemos distribuir 100 unidades e na mesma transação efetuar a distribuição e checar na validação os diferentes pesos. Numa eleição de presidente, governador, senador, deputados federais e estaduais cada categoria teria 20 unidades e os votos de cada categoria seriam uma fração do todo.

Na Figura 4.1 temos a visualização desta modelagem. Esta modelagem possui três tipos de cédulas, que são as categorias. Assim como há três entradas, o número de saídas deve ser no mínimo três, mas podendo ser mais, caso o eleitor escolha mais de um candidato por categoria. A Figura 2.6 ilustra o voto realizado nos testes.

A arquitetura do sistema de votação eletrônica proposta nesta dissertação consiste em servidores, chamados de nodos, que fazem a criação de blocos e inclusão das transações. Estes servidores trocam blocos e transações entre si, além de receberem dos clientes as transações. Cada cliente cria e assina a transação antes de enviar. Além disso servidores de auditoria podem ser criados e incorporados à rede(SHAHEEN et al., 2017). Esta rede deve ser protegida por *firewall*. Embora não haja formas de acesso ilegais no sistema atual, novas descobertas no protocolo podem ocorrer. Os servidores devem ser máquinas com grande capacidade de processamento e acesso a disco de alto desempenho. Os aplicativos clientes precisam apenas implementar uma forma de requisição *http* com os devidos parâmetros. Estes clientes são os "nodos

Figura 4.1 - Modelagem de um voto como transação.



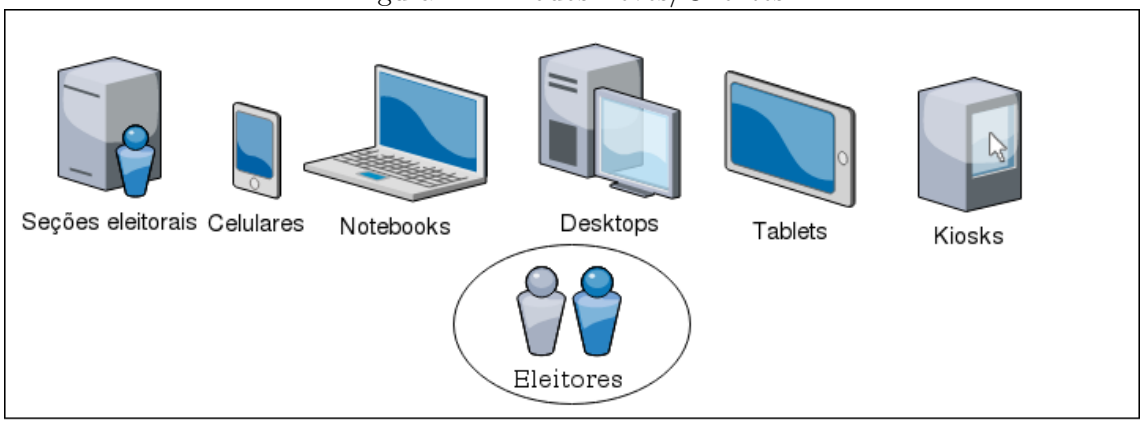
Fonte: Autor.

leves", não guardam todo o registro da *blockchain*, mas apenas a sua transação, o seu voto (Figura 4.2). A comunicação entre os nodos leves e completos passa por uma autenticação, o nodo leve também recebe uma lista de nodos completos que pode se conectar. Mesmo que o nodo leve se conecte a um nodo impostor, ele não envia sua chave privada, apenas seu voto assinado.

Cada cliente precisa inicialmente se conectar a um servidor de autenticação e direcionamento, o qual irá autenticar o usuário e indicar qual servidor *blockchain* deverá se conectar. Também envia os endereços dos candidatos possíveis para aquele eleitor, de acordo com o local de registro da pessoa ela terá candidatos regionais (Figura 4.3).

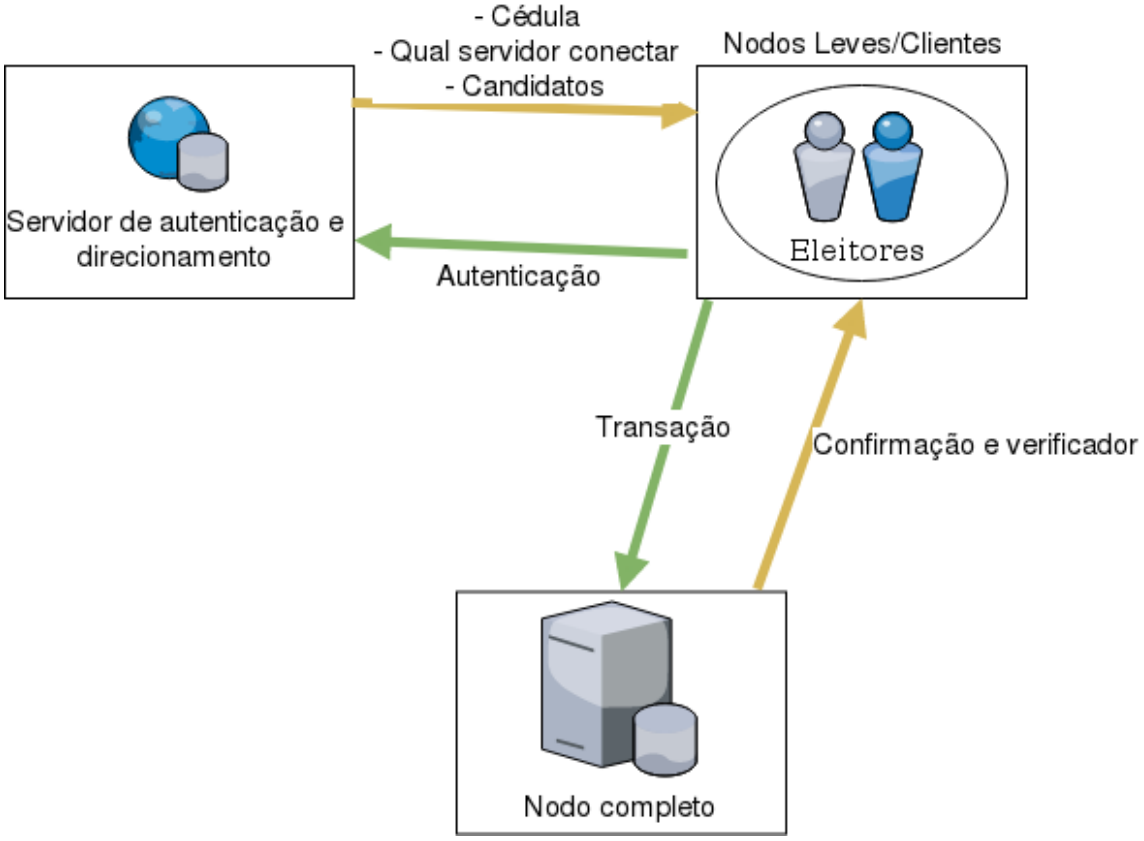
O relacionamento entre os clientes e nodos completos e entre nodos completos pode ser visto na Figura 4.4. Entre os nodos completos são trocadas mensagens contendo transações e blocos inteiros. Os clientes e nodos completos trocam apenas a transação e sua confirmação/verificador. Os nodos completos de auditoria são utilizados para verificar em tempo real o processo de voto e apuração.

Figura 4.2 - Nodos Leves/Clientes.



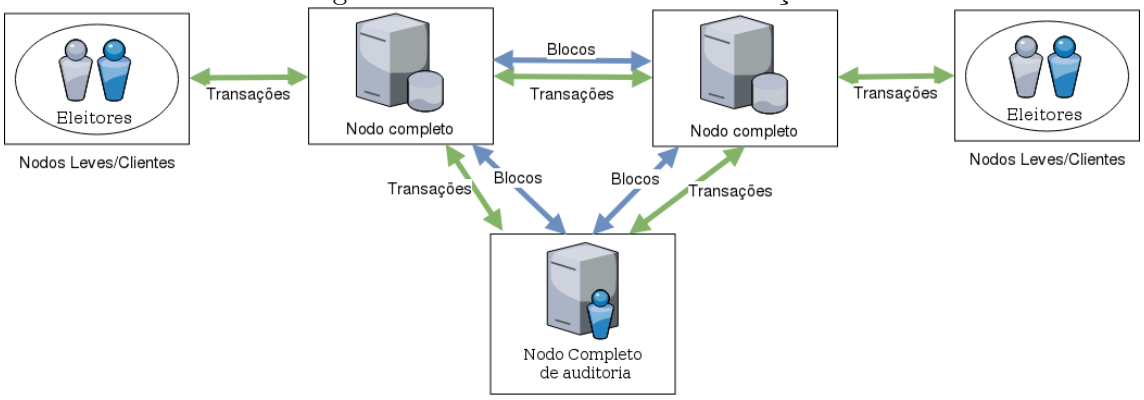
Fonte: Autor.

Figura 4.3 - Autenticação, distribuição dos candidatos e direcionamento.



Fonte: Autor.

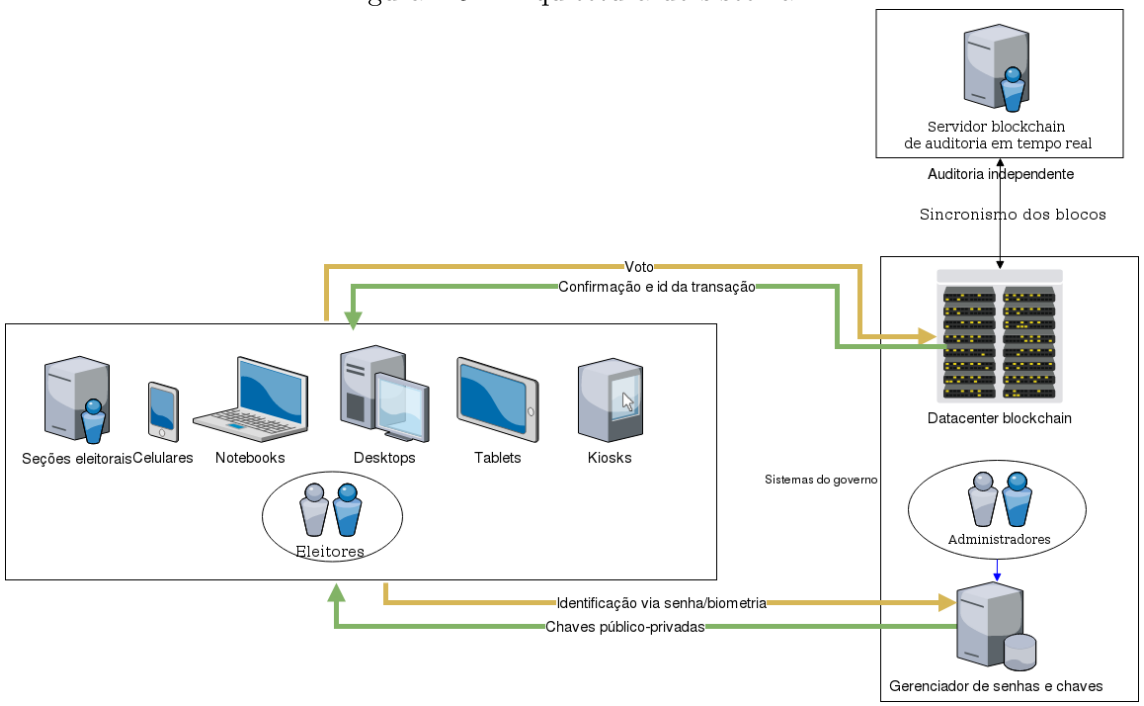
Figura 4.4 - Troca de blocos e transações.



Fonte: Autor.

Na Figura 4.5 temos a visualização de nodos completos (na forma de *datacenters*), servidores de autenticação e direcionamento (gerenciam chaves, senhas e direcionam o cliente), clientes (nodos simples) e auditores (nodos completos).

Figura 4.5 - Arquitetura do sistema.

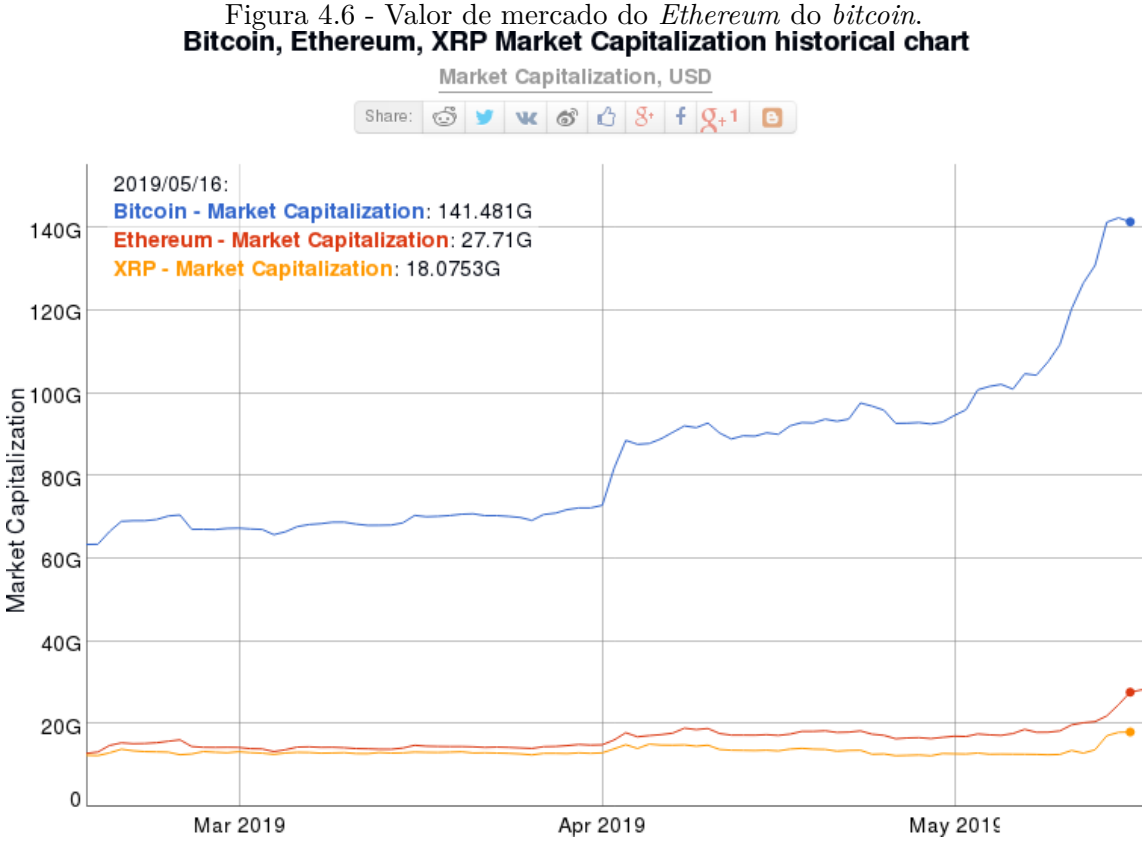


Fonte: Autor.

4.1 Justificativa da escolha do *blockchain* para este trabalho

A escolha deste trabalho foi inicialmente criar um código do zero, porém isto foi feito apenas de forma didática para aprender os conceitos, existem várias implementações disponíveis que demonstram os conceitos do *blockchain*(NASH, 2017). Em seguida se baseou no código do *bitcoin*, este projeto foi o que iniciou a revolução do *blockchain*, desde sua criação não houve nenhuma falha de segurança grave, é a criptomoeda com maior capitalização (Figura 4.6), valor individual (Figura 4.7), maior valor mediano de transferências (Figura 3.1) e maior número de endereços ativos (Figura 4.9). Essa importância do *bitcoin* significa que em qualquer momento existem dezenas de milhares de desenvolvedores e usuários procurando falhas em sua rede de modo a obter acesso ou criar *bitcoins* sem efetuar os cálculos necessários.

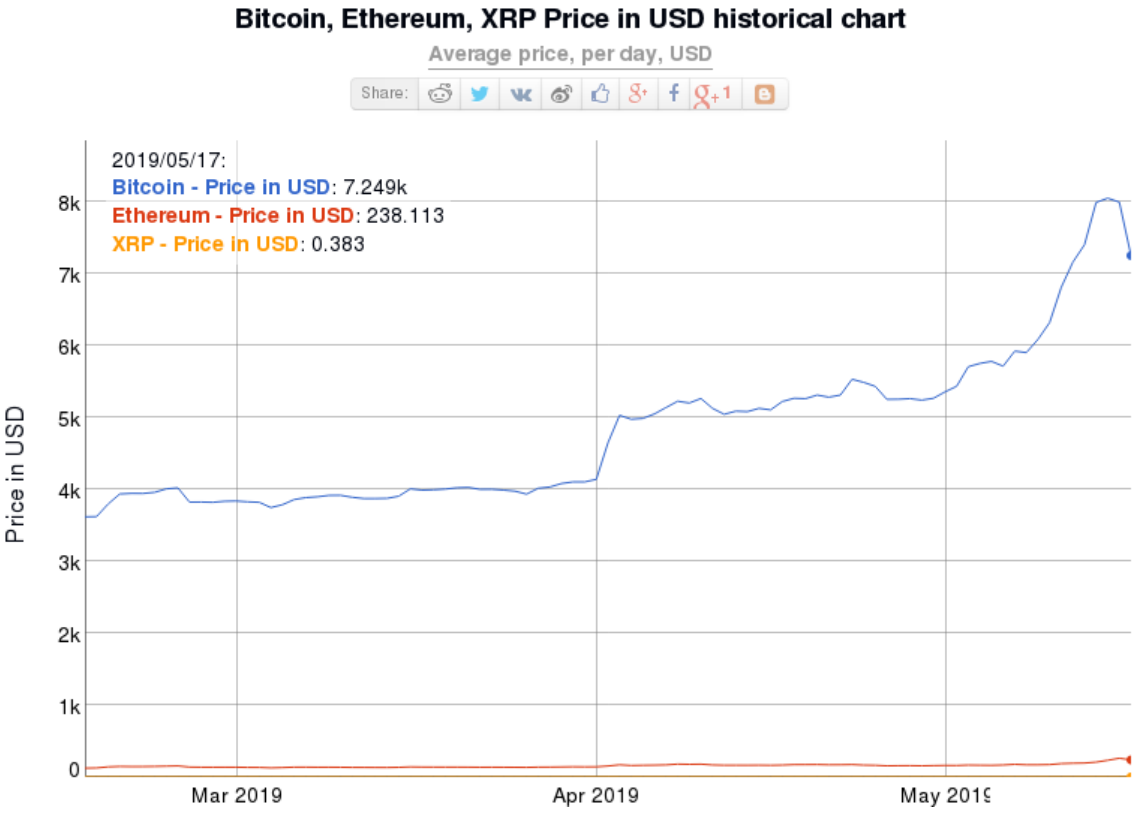
Figura 4.6 - Valor de mercado do *Ethereum* do *bitcoin*.



Fonte: BitInfoCharts (2019).

O desafio do trabalho foi alcançar os números de eleitores e tempo de votação semelhantes ao da população brasileira. A rede do *bitcoin* foi projetada desde o princípio

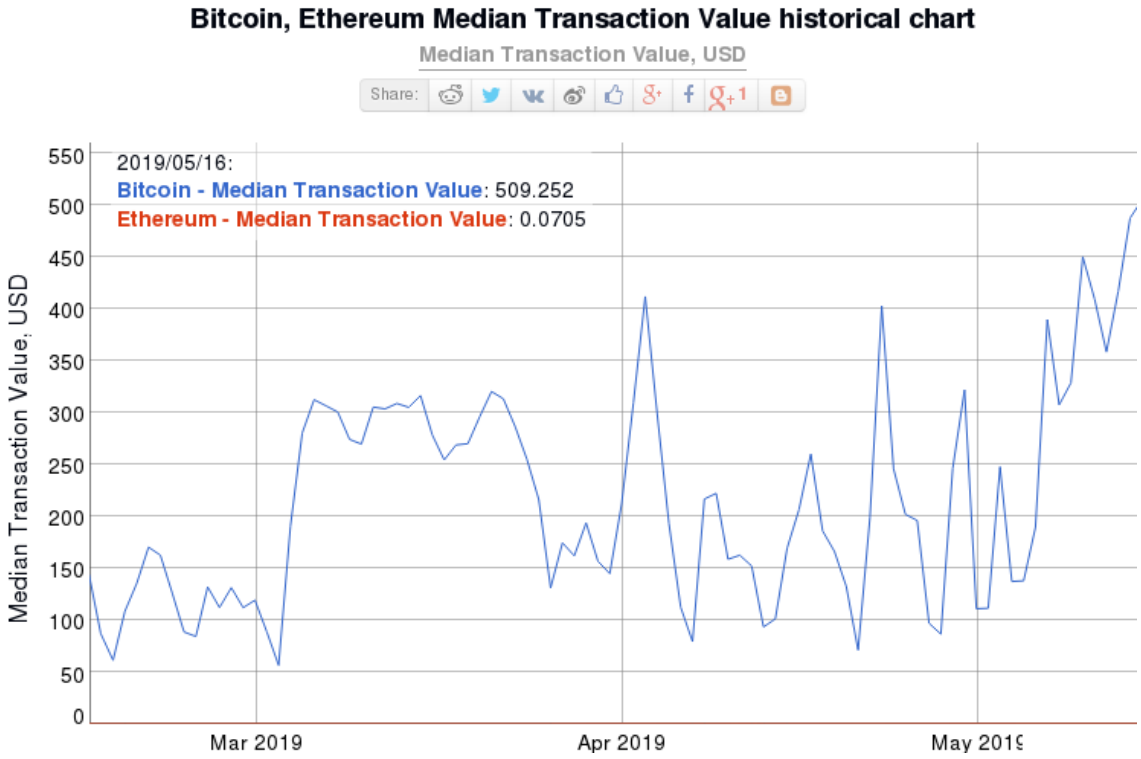
Figura 4.7 - Valor individual do *Ethereum* e do *bitcoin*.



Fonte: BitInfoCharts (2019).

para gerar novos blocos a cada 10 minutos (Figura 4.10), com cada bloco tendo em média 2000 transações, o algoritmo se adapta ao poder computacional da rede fazendo com que este intervalo se mantenha não importando o número de nodos da rede. Também conta com diversas funções, distribuição dos blocos, distribuição das transações, banco de dados otimizado para gravação e recuperação das informações, criação de uma rede independente, código criptográfico robusto. Também a cada novo desenvolvimento do código principal do *bitcoin*, as mudanças podem ser incorporadas retroativamente, não sendo necessária uma equipe de desenvolvimento própria e se utilizando do conhecimento da comunidade. A ideia original com o código do projeto *bitcoin* foi modificar a dificuldade mínima de cada bloco, permitindo uma geração mais rápida. Cada bloco gerado, ou minerado, recebe como prêmio pela participação na rede 50 *bitcoins*, onde cada moeda pode ser dividida em 100 milhões de unidades. Além do intervalo, também o tamanho dos blocos, permitindo uma geração mais rápida de blocos maiores. Os mineradores, ou nodos completos iriam distribuir frações dos *bitcoins* para cada eleitor, processo o qual

Figura 4.8 - Mediana do valor das transações diárias do *bitcoin* e *Ethereum*.



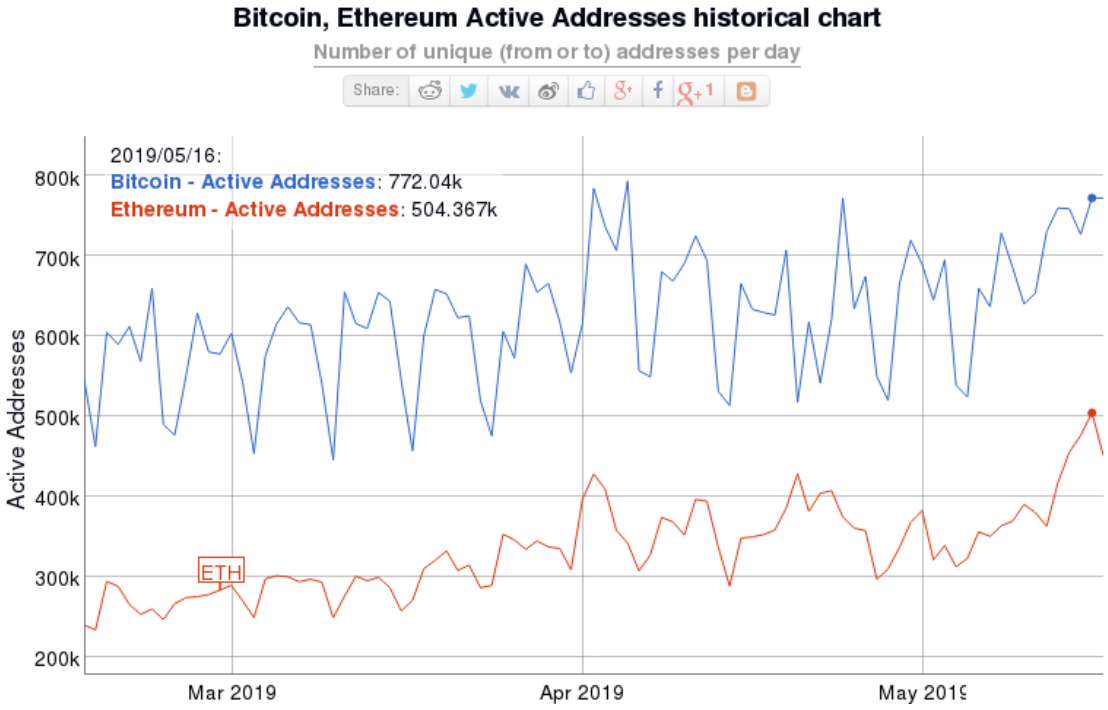
Fonte: BitInfoCharts (2019).

também gera transações, precisando de um período para criação destes novos blocos. Nesta ideia, qualquer nodo externo poderia se conectar à rede e incluir novos blocos, gerando para si *bitcoins* e podendo distribuí-los, isso seria o equivalente a forja de votos. Isto poderia ser mitigado pela criação de uma rede de computadores protegida por *firewall*. A seguir em 4.2 temos a solução adotada quanto aos nodos externos.

O conceito do sistema de voto com *blockchain* é equivalente a servidores funcionando como nodos completos, que gerariam blocos e receberiam transações. No período de preparação estes servidores gerariam um par de chaves e um endereço para cada eleitor, fariam a criação das cédulas distribuindo os *bitcoins* nestes endereços. Essa associação eleitor e endereço seria de responsabilidade do governo, para distribuição das chaves uma autenticação é necessária, devendo ser uma senha cadastrada previamente ou biometria(AKBARI et al., 2017).

O problema da utilização de prova de trabalho, são os recursos computacionais

Figura 4.9 - Endereços ativos diários do *bitcoin* e *Ethereum*.



Fonte: BitInfoCharts (2019).

exigidos, cada vez maiores, também há o fato de que a maioria dos cálculos seria desperdiçado, resultando em tempo e energia gastos sem utilidade.

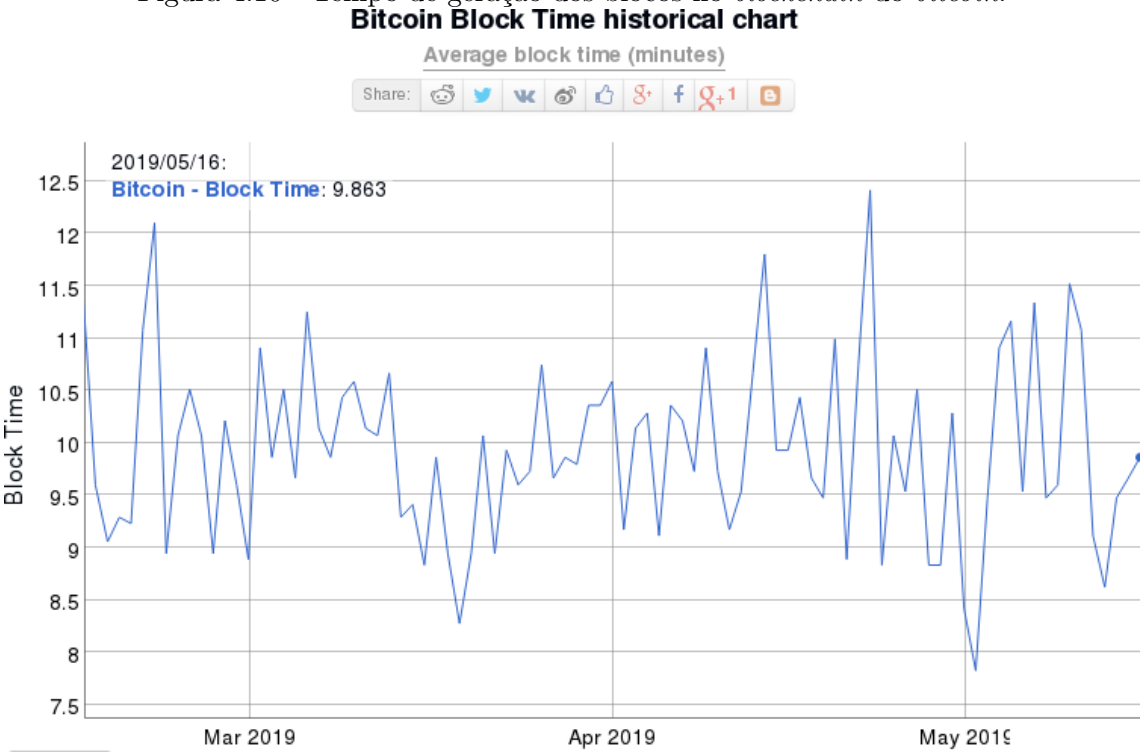
O projeto final trabalhou modificando o tamanho dos blocos, o tempo de processamento entre cada bloco, a criação de transações especiais que criam *tokens* utilizando-os como cédulas, permissões de acesso, paralelização da apuração e da simulação de milhares de votos simultâneos. Estas modificações foram feitas com base no código do *bitcoin* e do *multichain*. A arquitetura proposta também transforma a rede blockchain criada em federativa.

4.2 Utilizando permissões de acesso

Devido aos custos energéticos envolvidos ao se utilizar o método de prova de trabalho(ver 2.2.6) original do *bitcoin* (ver 2.2.8) foi adotado um modelo permissionado. Isto também impede que nodos aleatórios se conectem a rede.

Um *blockchain* com permissões possui um controle de acesso, que dita quem pode administrar, conectar, enviar, receber, minerar. Algumas soluções já existem para isso

Figura 4.10 - Tempo de geração dos blocos no *blockchain* do *bitcoin*.



Fonte: BitInfoCharts (2019).

como o Corda(BROWN et al., 2016), *Chain Core, Credits, HydraChain, BigchainDB*.

Para este projeto foi escolhida a implementação do *blockchain multichain* por ser baseado e manter compatibilidade com o código do *bitcoin*, implementando este controle de acesso. O *multichain* é um *fork* do *bitcoin*.

Esta escolha também se deve pela licença do código ser aberta e qualquer trabalho derivativo também ser; isto corrobora com os valores de que a pesquisa financiada com verba pública ser aberta a todos. O fato do código ser livre permite a verificação independente da corretude do modelo criptográfico.

O administrador possui as chaves iniciais, assim que se inicia uma *blockchain*, o software cria uma chave que assina os blocos e os identifica ao conectar com outros nodos. Esta chave concede permissões a qualquer outra chave. No sistema implementado as permissões adotadas são:

- Apenas nodos permitidos podem se conectar

- Apenas nodos permitidos podem criar blocos
- Todos os nodos podem enviar e receber transações
- Apenas transações permitidas podem ser enviadas
- Apenas nodos permitidos podem criar transações iniciais

Estas permissões são armazenadas dentro da própria *blockchain*, o nodo que tiver acesso e se conectar irá efetuar uma cópia de todos os blocos, transações e permissões.

O uso de uma *blockchain* permissionada permitiu uma grande economia energética, em relação ao modelo convencional, envolvendo prova de trabalho, porém também poderia ser adotado um modelo com distribuição predeterminada dos turnos de geração de blocos(HANIFATUNNISA; RAHARDJO, 2017).

4.3 Criando moedas não nativas dentro do blockchain

A criação de bens, tokens ou moedas não nativas dentro do *blockchain* é uma ideia que extende o conceito original. As transações não são mais das mesmas unidades, diferentes moedas podem ser trocadas. Um sistema de registro de uma casa de câmbio pode especificar que houve troca de dólares por euros entre dois usuários (identificados no *blockchain* pelas suas chaves). A cada eleição pode ser criada uma nova moeda e ser usada como cédula, essa cédula pode identificar o ano e local da eleição. Também permite que diferentes votações sejam realizadas ao mesmo tempo utilizando o mesmo *blockchain*.

Algumas implementações OpenAssets[1], OpenChain[2], de *blockchain* oferecem a criação destes tokens. Entre elas o *multichain*, facilitando assim o reuso de código e a integração, pois o protocolo utilizado é comum entre estes projetos e o projeto principal do bitcoin.

A criação de moedas não nativas herdada do *multichain* permitiu que os votos pudessem ser criados sem a limitação de quantidade. As moedas nativas no protocolo original do *bitcoin* se originam a partir da criação de cada novo bloco, inicialmente 50 moedas por bloco e tem um total de 21 milhões. Com a criação arbitrária de novas moedas/cédulas, pode-se criar a quantidade desejada e o fracionamento desejado. Ainda podem ser criadas diferentes cédulas na mesma *blockchain*, cédulas para presidente, governador, prefeito, etc. Dessa forma uma cédula para presidente não pode ser utilizada para votar em um governador.

[1] https://github.com/OpenAssets
[2] https://www.openchain.org/

4.4 Possíveis problemas

Dentro da arquitetura proposta podemos ter alguns problemas e abaixo algumas soluções.

4.4.1 Propaganda externa

Neste o sistema eletrônico de votação o sistema seria gerido pelo governo e auditores. Para evitar interferências externas as medidas seriam as mesmas do modelo atual de voto: policiamento das redes sociais para evitar a divulgação de *fake news* e aplicativos de mensagens eletrônicas monitorados para que não haja o repasse de mensagens políticas falsas. O problema de propaganda não é exclusivo da arquitetura proposta. Devido a possibilidade do eleitor votar de qualquer localidade, a proibição de distribuição de panfletos perto das zonas eleitorais deve ser estendida para a internet, controlando-se as mídias sociais. Os órgãos fiscalizadores podem ser organizações da própria sociedade, com membros de vários partidos e ideologias.

4.4.2 Distribuição dos clientes

Os aplicativos para uso devem ser distribuídos por plataformas seguras, como a App Store para iOS e Play Store para Android. Para desktops a distribuição poderia ser pela Microsoft Store. A segurança individual de cada dispositivo é responsabilidade do usuário, assim como acontece atualmente. Como o voto pode ser verificado, qualquer mudança no voto seria percebida na apuração do resultado, com o usuário conferindo se o voto foi para o candidato escolhido.

Caso o dispositivo do eleitor seja vítima de uma falha de segurança, o eleitor teria sua cédula comprometida, mas o restante da população não seria afetada.

4.4.3 Fraude da eleição pelo governo

Na arquitetura proposta todo o software é aberto e os resultados também divulgados, não apenas a contagem final dos votos mas como todos os votos individuais, com a identidade de cada eleitor mascarada.

Cada eleitor pode verificar seu próprio voto, através da *hash* de seu voto. Esta *hash* faz parte do *blockchain*, a modificação da mesma provocaria um efeito em cascata invalidando o restante dos votos.

Além de verificar que seu voto foi gravado de forma correta, qualquer usuário de posse da *blockchain* final pode fazer a contagem dos votos e comparar com o resultado

divulgado. Nesta *blockchain* constam todos os votos emitidos e votos realizados.

4.4.4 Sybil attack

No *bitcoin* quando uma entidade controla mais do que 50% da rede, ela pode decidir incluir blocos sem transações legítimas. No sistema proposto isto é evitado com o governo controlando a rede. E somente as transações criadas inicialmente podem ser transmitidas, somente o administrador tem autonomia pra criação das cédulas iniciais. Pela natureza aberta da apuração dos votos, qualquer manipulação feita pelo governo com os votos pode ser detectada, cada eleitor pode verificar se seu voto foi para o candidato correto e também a apuração dos votos. Em Yu et al. (2019) temos uma proposta que limita o poder de tais entidades, mesmo que controlem uma parcela acima de 50% da rede não conseguiriam incluir novos blocos, substituindo o poder computacional por um sistema de reputação, considerando a contribuição dos nodos ao longo do tempo, dessa forma novos nodos mesmo com mais capacidade de processamento não poderiam incluir blocos devido ao seu histórico.

4.4.5 Consenso bizantino

Este problema é sobre o consenso entre as informações transmitidas, cada transação recebe uma identificação que pode ser usada para verificá-la. Também há o consenso entre os nodos sobre os blocos inclusos, todos terão a maior cadeia de blocos, pelo sistema proposto ser controlado, não há competição para geração de cadeias diferentes. Em Meng et al. (2018) foi proposta uma melhoria no protocolo de consenso para o *bitcoin*, mas ainda não foi testado dentro da rede pública do *bitcoin*. Na literatura temos Wang et al. (2019) explorando o uso de créditos e checkpoints para aperfeiçoar o processo de consenso.

4.4.6 Transação duplicada

Um mesmo eleitor pode criar duas transações com saídas diferentes e a mesma entrada, mas somente a primeira será validada recebendo um id da transação dentro da *blockchain*. Isto é um problema no *bitcoin* pelo fato de que há várias cadeias de blocos diferentes, no sistema proposto só há uma.

4.4.7 Criação de votos aleatórios (interferência externa)

Na nossa implementação isto é evitado pois a criação de um novo bloco não tem recompensa e somente as transações permitidas são inclusas. A quantidade de cédulas é fixa e criada antes da distribuição.

4.4.8 Inclusão de blocos aleatórios

No *bitcoin* um nodo pode entrar na rede e gerar um bloco com a dificuldade mínima correta, a recompensa por essa geração é um número de *bitcoins* que pode ser usado. Na nossa implementação isto é evitado pois somente nodos autorizados podem fazer parte da rede.

4.4.9 Brute-force de votos

Uma transação precisa de uma transação de origem e uma chave privada, esta chave privada pode tentar ser adivinhada, utilizando valores aleatórios. O custo computacional é altíssimo e a recompensa seria um único voto.

4.4.10 Criptografia

A criptografia por trás das chaves *ECDSA* é definida pelo padrão americano ANSI X9.62, intitulado *"Public Key Cryptography for the Financial Services Industry: the Elliptic Curve Digital Signature Algorithm (ECDSA)"*, criptografia de chaves públicas para a indústria financeira.

> "Uma chave RSA de 2048 bits provê um nível de segurança de 112 bits. Entretando uma chave ECDSA requer somente uma chave de 224 bits para prover os mesmos 112 bits de segurança."(NAZIRIDIS, 2018)

No *blockchain* deste trabalho, as chaves possuem 256 bits, tendo sido extensivamente testada pela indústria e academia. Caso ela seja quebrada, o código pode ser trocado, esta é uma das vantagens de se ter utilizado dos projetos *bitcoin* e *multichain*, uma falha grave dessas teria resposta imediata e por ser um trabalho derivado o sistema de votação pode incorporar as mudanças facilmente.

Esta robustez é devido ao fato do *bitcoin* ter sido o primeiro *blockchain* criado, possuir centenas de desenvolvedores e milhares de usuários. Há uma recompensa financeira gigantesca para falhas no protocolo, ainda assim nenhuma foi descoberta. Todas as grandes falhas relacionadas foram nos sistemas de corretagens.

4.4.11 Tamanho do blockchain

A base de dados com 32 milhões de votos distribuídos e votados possui 28GB. Isto não é um problema para os servidores, chamados nodos, mas caso os eleitores

precisassem ter um nodo para votar isto seria impraticável. Cada cliente, que pode ser um computador desktop, celular, página web, precisará apenas das informações das chaves privada e id da transação (sua cédula), estas informações são palavras-chaves de 58 caracteres e 36 respectivamente, a criação e assinatura da transação serão locais, não sendo necessário guardar os blocos.

4.4.12 Segurança

Há vários níveis de segurança, onde cada um reforça o anterior. O primeiro deles é o acesso aos nodos através de uma requisição HTTP-RPC, esta possui uma senha, que pode ser divulgada aos clientes somente no dia desejado, isso evitaria que os nodos recebam requisições de agentes externos. Antes da eleição haveria uma camapanha de educação do novo sistema, inclusive como usar esta senha que seria distribuída no dia.

O segundo é a possibilidade de utilizar HTTPS, isto evitaria que os dados das transações sejam capturados; mesmo que haja captura, ela apenas divulgaria a informação de voto daquele cliente, não podendo ser modificado o voto. A transação é assinada com a chave privada do eleitor e qualquer modificação invalidaria o voto. Caso a rede de entrada dos nodos seja comprometida isto divulgaria as intenções de votos de todos, mas não mudaria o resultado da eleição. Quando houver fraude generalizada, a população pode descobrir através da auditoria dos votos.

O terceiro nível é a limitação das operações dos clientes com o nodo, que seriam apenas a autenticação e envio do voto. No código original do *bitcoin* podem ser requisitadas informações sobre transações, o tamanho da *blockchain*, desempenho do sistema, todos os blocos podem ser acessados por essa interface.

A quarta medida de segurança é a geração das cédulas pelo administrador, somente ele possui a chave com permissão para tanto, não podem haver transações utilizando outras cédulas.

O quinto nível é que os eleitores somente podem enviar para os candidatos, as permissões de acesso são modificadas para que os eleitores da lista possam somente enviar e os candidatos somente receber, o voto dos próprios candidatos teria de ser registrado de outra forma, até utilizando de voto em papel pelo número de candidatos ser muito menor que a da população.

O banco de dados com as informações de usuários e suas respectivas chaves é sigiloso, a quebra de seu sigilo implicaria na criação de votos maliciosos e associação entre

Listagem 4.1 - Exemplo de mensagem assinada digitalmente

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA256

This is a signed message.
-----BEGIN PGP SIGNATURE-----

iQEzBAEBCAAdFiEEKV3KrJyKwGsHctQVdRB3RdghPTgFAl1eKFcACgkQdRB3Rdgh
PTjmEwgA4OVj93j9VqaCAHmDJo+zkqv7kH9dOWUA9Zs/wV6j+9PJx/vi1zbuI2P8
yOizVSazbfCcrY3fcQTgqi29f3aN9qaPwXrSN68AwqhDSohxTtP52qvjD2e/uqIh
h1eiR83wJRNlrW+3/3A3mJ2AHXWI0DUG5YN+G4D98xQ6VpqblfeJSKl4tzgVIHk+
tNTIb+rgmdml6REYxdNWfsZpeGlSRI0DB7PpcqgC2tG817S93hLhvn7KuNY6KDoi
iyCrr36wfDIp5L54+N9+HDU+OP/4M3eNyAD3ESEkZ+t7vAEzsfTBQMM9qe0rnV4H
VI80YHhz90Rp/8UTNrEk8HsQxu6WJQ==
=6fpH
-----END PGP SIGNATURE-----
```

chaves públicas e usuários.

As transações são assinadas com o algoritmo de chaves público-privadas, assim podem ser trafegadas por uma rede insegura, mas qualquer modificação invalida sua assinatura. Seria utilizado o mesmo mecanismo usado em assinaturas de e-mail (Listagem 4.1). Esta assinatura garante que os votos realmente correspondam aos votos dos eleitores. A decodificação dos mesmos, através da confirmação, garante que o voto na tela é realmente o que foi enviado.

4.4.13 Verificação das transações

O total de verificações é feito na inclusão das transações na *blockchain*. O cliente recebe sua cédula eleitoral, o eleitor escolhe seus candidatos, assina com sua chave privada e os envia, localmente as verificações são quanto a quantidade de votos de cada candidato e se é válido. O nodo receptor irá efetuar as seguintes conferências:

- Transação de entrada utilizada é válida
- Índice da transação de entrada utilizada é válido
- Eleitor ainda não votou (A transação de entrada precisa fazer parte da lista de transações válidas UTXO)

- Cédula é válida
- Quantidade é válida
- Transação utilizada tem como recipiente o eleitor
- Assinatura da transação condiz com a chave do eleitor
- Destino da cédula é válido
- *Script* utilizado é válido

Todas estas conferências garantem que o voto seja transmitido corretamente.

Caso o eleitor não consiga efetuar seu voto, deve procurar o Tribunal Regional Eleitoral que pode investigar. Ou ainda no banco de dados final o usuário pode descobrir se seu voto foi feito sem seu consentimento.

4.5 Tecnologia

O código que implementa o sistema de voto eletrônico proposto neste trabalho está organizado em diversas funções lógicas, que se dividem da seguinte forma:

- Busca de novos nodos
- Troca de mensagens, transações e blocos completos entre os nodos
- Mineração/criação de novos blocos
- Checagem de *scripts* das transações
- Recepção de novas conexões *RPC*

Os clientes possuem apenas as funções de:

- Conexão e autorização à *blockchain*
- Recepção das chaves público privadas
- Recepção dos endereços dos candidatos
- Escolha dos candidatos
- Criação das transações com os candidatos escolhidos

- Assinatura das transações com a chave privada
- Envio da transação
- Recuperação das informações da transação pelo *txid* verificador

Algumas funções são executadas deliberadamente, tais como: criação das transações iniciais para as cédulas, distribuição das cédulas para os eleitores, contagem dos votos. Exigem participação do administrador. Qualquer participação do administrador fica registrada na *blockchain*, podendo ser investigada posteriormente.

A criação é uma transação especial que pode ser criada sem ter uma entrada.

A distribuição dos votos é sequencial, com o total de eleitores dividido pelo máximo de saídas em uma transação, sendo usado 4000. Na etapa de distribuição dos votos temos os seguintes passos: criação de uma transação com o total de votos a serem distribuídos; execução da rotina que distribui os votos seguindo a lista de eleitores, respeitando o limite de 4000 saídas por transação, por este motivo o processo é sequencial: as primeiras 3999 saídas vão para endereços de eleitores, a última com o restante das cédulas é utilizada como entrada da próxima redistribuição.

A contagem dos votos pode ser expressa através do pseudocódigo da Listagem 4.2. A leitura dos blocos pode ser feita em paralelo, assim a leitura em disco fica não sequencial, o que em discos mecânicos diminuiria o desempenho. Cada *thread* numa leitura paralela iria ler um bloco, o disco teria requisições simultâneas em diferentes arquivos. Em discos de memória de estado sólido, o acesso aleatório tem o mesmo tempo que leituras sequenciais. Em uma execução paralelizada divide-se o *blockchain* entre o número de *threads* desejado, assim cada *thread* faria a contagem de blocos diferentes.

Na paralelização tomou-se cuidado com a escrita concorrente das mesmas variáveis, isto foi evitado criando variáveis temporárias para cada *thread* e ao final uma rotina que fizesse a consolidação dos dados.

Nos testes a lista de eleitores foi criada previamente, com um banco de dados contendo a chave pública, privada e endereço. Esta lista fica gravada na *blockchain*, na forma das cédulas que cada eleitor tem direito. No final da eleição os resultados podem ser apurados, identificando-se quantos eleitores efetuaram seus votos. Esta lista deve estar de acordo com a estimativa no número de eleitores do TSE.

Em uma execução segura do sistema, cada eleitor cria sua chave público-privada

Listagem 4.2 - Pseudocódigo da apuração dos votos

```
int saida.candidato_votos
string saida.candidato
for bloco in BLOCKCHAIN:
    for tx in bloco:
        for saida in tx:
            mapa_votos[saida.candidato]=mapa_votos[saida.
               candidato] + saida.candidato_votos
for candidato in mapa_votos:
    imprime candidato.nome, candidato.votos
```

Fonte: Autor.

utilizando software e entropia local a seu dispositivo, depois enviando o endereço para uma central para entrar na lista da distribuição de votos. Dessa forma apenas o eleitor teria acesso à sua chave privada e a autenticação serviria apenas para enviar o endereço e receber o id da transação para ser usada como cédula. Na literatura também há propostas desvinculando eleitores das cédulas(BARTOLUCCI et al., 2018).

A mineração, recepção e checagem de transações podem se beneficiar de vários núcleos; a busca de novos blocos no modelo tradicional em que a dificuldade mínimo dos blocos inicia-se em sete números zero à esquerda da *hash* cria uma dificuldade artificial, que faz com que as *threads* de mineração de blocos concorram entre si para encontrar esta *hash* alvo. Neste trabalho foi adotada uma dificuldade inicial baixa, que não se altera com o aumento de capacidade do sistema, isto evita o desperdício de processamento e devido ao uso de permissões; somente as chaves designadas podem criar, enviar e receber novas transações. Logo o número de *threads* utilizado para a mineração/criação de novos blocos é único. Em testes utilizando números variados de *thread* não se perceberam ganhos, o gargalo ficou sendo o acesso concorrente às estruturas de dados que guardam o cache de transações.

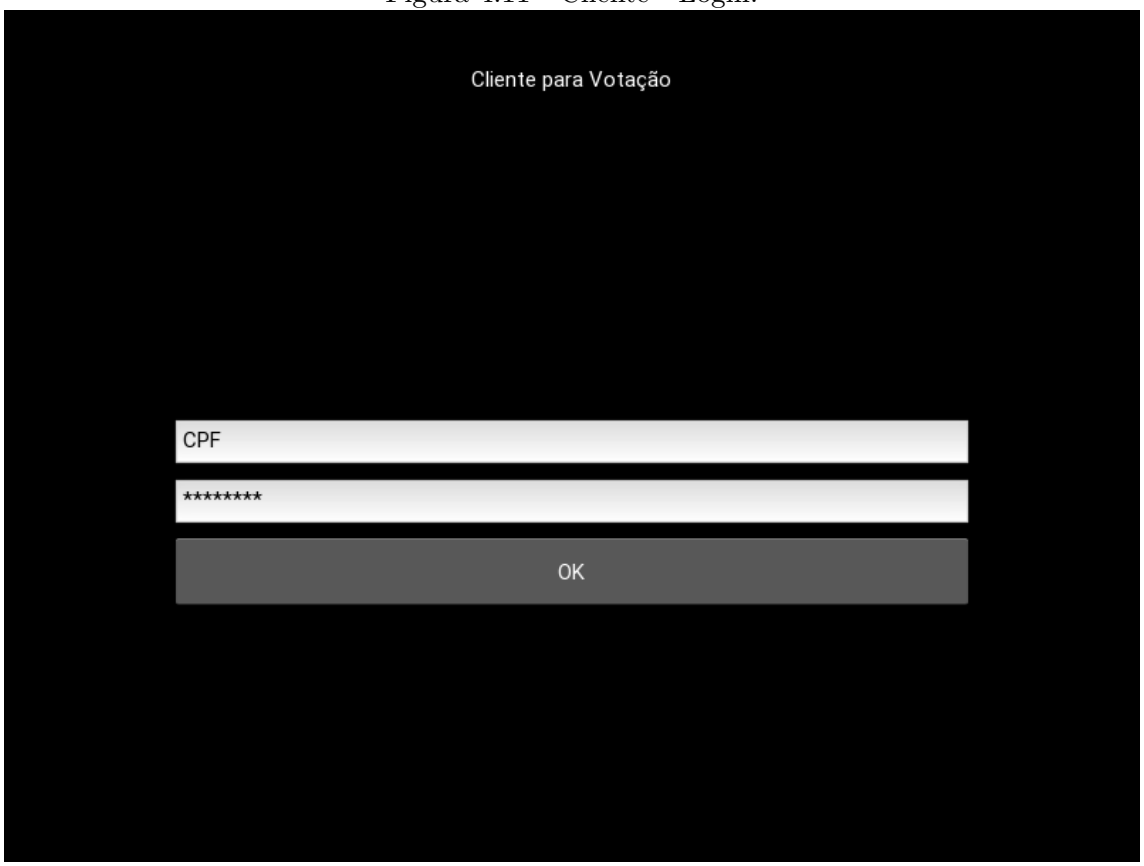
A recepção de novas transações foi definida de forma a se adequar ao número de conexões simultâneas esperadas, podendo chegar a 20.000, porém houve um consumo significante no uso de memória disponível, pois cada thread exige um gasto extra de memória no programa.

A checagem de scripts padrão utiliza todos os núcleos virtuais do processador, porém cada operação de checagem utiliza muito pouco tempo de processamento, sendo o overhead de criação de toda cadeia de funções um problema. Dessa forma não

houve o aproveitamento integral dos processadores. Para popular uma maior fila de processamento nos processadores utilizou-se um número maior de *threads* para a checagem, nos testes utilizou-se o número 320, como o processador utilizado possui 16 *threads*, criou-se uma fila de 20 *scripts* para cada um. Um dos conceitos utilizados no *blockchain* é a verificação de valores com base numa chave, a UTXO (ver 2.2.4). Esta lista de *hash* foi implementada através do *BerkleyDB*. Neste esquema há a consulta de *hash's* para verificar se já foi utilizada como transação ou não, caso já tenha sido utilizada como entrada de uma operação anterior a transação é inválida e não aceita dentro do *blockchain*. Isto evita o problema de gastos duplos, ou votos para diferentes candidatos utilizando a mesma cédula. Além do índice de transações esta biblioteca também guarda os índices dos blocos.

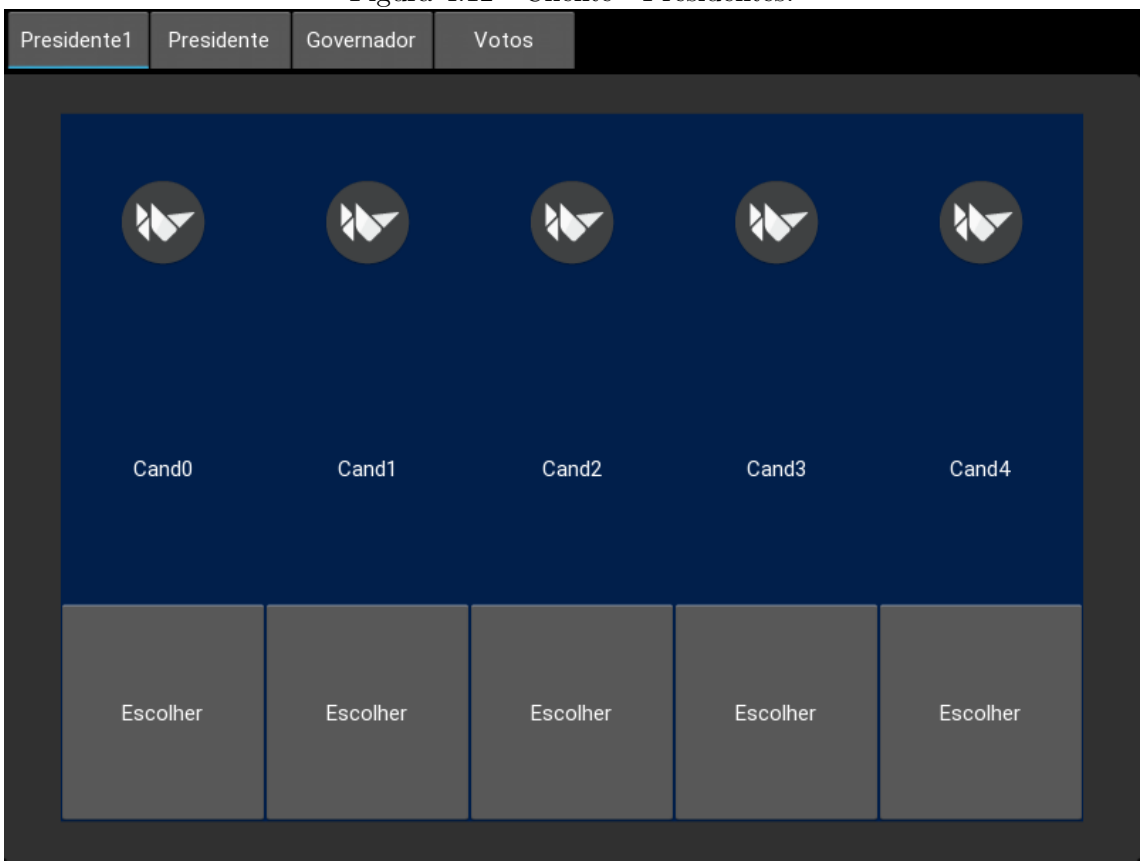
Foram criados clientes gráficos em python na plataforma kivy, para demonstrar a portabilidade do código (Figura 4.11 e Figura 4.12)

Figura 4.11 - Cliente - Login.



Fonte: Autor.

Figura 4.12 - Cliente - Presidentes.



Fonte: Autor.

Cada voto ou transação gera um identificador (Listagem 4.3). Ele pode ser usado para recuperar os dados de voto (Listagem 4.4) e assim validar se a escolha do eleitor foi mantida ou manipulada. Esse identificador pode ser convertido para um *QR code* (Figura 4.14) para facilidade de uso. O eleitor interessado em auditar seu próprio voto pode fazer o download do código, a base de dados, compilar ele mesmo e validar as informações, nessa transação haverá a transação origem da cédula dele, para quais endereços de candidatos ele enviou e a quantidade de votos. Existem interfaces gráficas (Figura 4.16,4.15,4.17) e web para explorar o *blockchain*, elas podem ser usadas neste sistema pois há compatibilidade entre os protocolos.

A conferência do voto é feita seguindo a estrutura da Figura 4.13, utilizando o código da Figura 4.14. Esta imagem é a codificação visual da id textual da Listagem 4.3, assim temos a informação sobre a transação como visto no Listagem 4.4. Como administrador além da transação, temos informações sobre os blocos como visto na

Listagem 4.3 - Identificador da transação validada

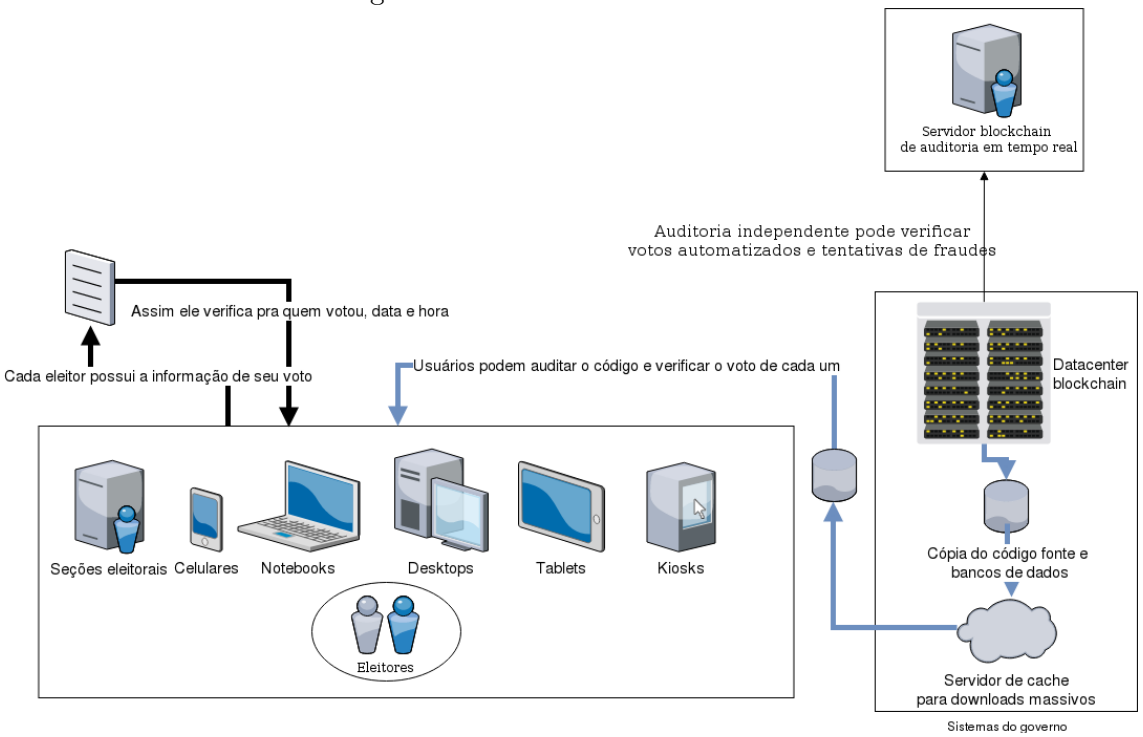
```
7d2572c5426faca62f37e6ab275fafd792869e9ee2f5d32b5aaa767e1e375f82
```

Fonte: Autor.

Listagem 4.5. O administrador assim como qualquer usuário, pode ver os votos da eleição posteriormente no processo de auditoria, porém apenas identificando a chave pública de cada um e não sua associação com uma identidade. Qualquer ação no *blockchain* fica registrado e pode ser auditado posteriormente.

A *blockchain* como um todo pode ser explorada através de visualizadores web (Figuras 4.16, 4.17, 4.15), o *blockchain* destes visualizadores é o do *bitcoin*, mas estas ferramentas poderiam ser adaptadas para o sistema de votos presente.

Figura 4.13 - Processo de auditoria.

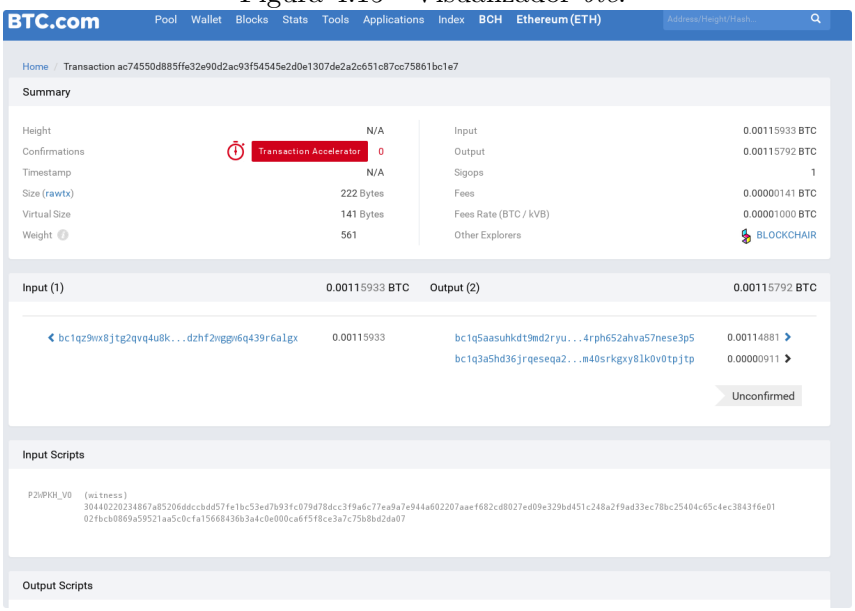


Fonte: Autor.

Figura 4.14 - QR Code verificador.

Fonte: Autor.

Figura 4.15 - Visualizador *btc*.



Fonte: BTC (2019).

4.6 Objetivos

Ao se implementar este sistema buscou-se os seguintes objetivos:

a) Anonimato

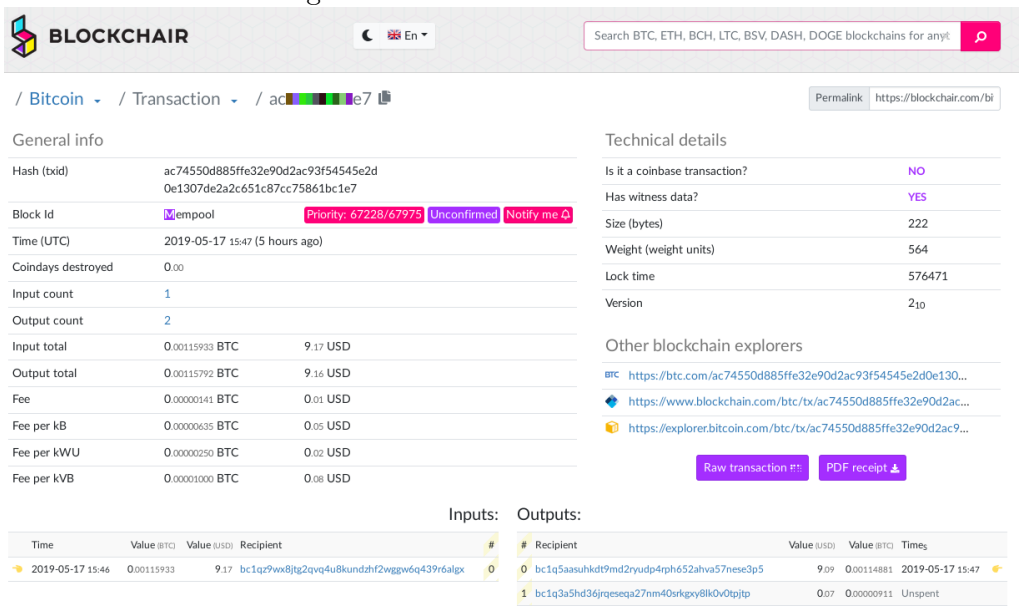
b) Segurança

c) Imutabilidade

d) Acesso Remoto além de zonas eleitorais

e) Verificação pelo usuário

Figura 4.16 - Visualizador *blockchair*.



Fonte: Blockchair (2019).

f) Auditabilidade

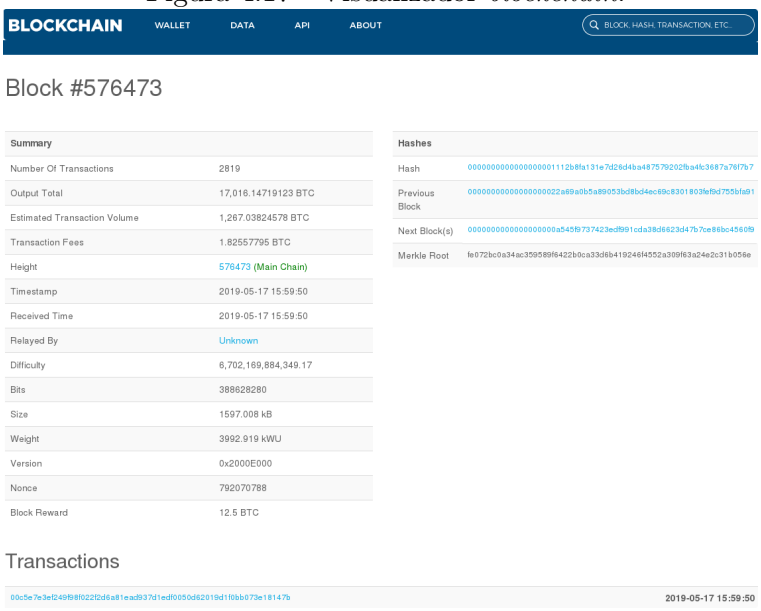
g) Código aberto

O anonimato foi garantido pelo fato dos usuários terem apenas informações das chaves públicas (através dos endereços no *blockchain*) registradas no banco de dados. A associação entre um eleitor e sua identidade nos registros só poderá ser feita por cada um, a menos que o mecanismo que associa as chaves públicas e a identidade dos eleitores seja divulgado.

A segurança se teve ao utilizar criptografia estabelecida e implementações já testadas.

Imutabilidade vem da própria natureza do *blockchain*, onde cada novo bloco aumenta ainda mais a segurança dos dados. Acesso remoto se teve pelo uso de uma requisição HTTP aos serviços, qualquer cliente pode ser escrito desde que possua uma biblioteca de requisições web. Podem ser criados clientes para computadores *desktop*, aplicativos para *Android* e *iOS*, navegadores web. Esta requisição utiliza *JSON* e poderia ser feita utilizando HTTPS assim evitando que os dados da rede sejam interceptados e se crie uma associação da chave pública com um endereço da internet e consequentemente com uma pessoa física. Isto é importante para a

Figura 4.17 - Visualizador *blockchain*.



Fonte: Blockchain (2019).

auditoria, os votos apurados não poderiam ser modificados depois de gravados.

Verificação foi implementada através do id da transação, cada voto irá retornar um identificador que irá permitir saber o destino do voto, quantidades e data. A verificação é importante para garantir a integridade do eleitor. Futuramente o voto sob coação ou venda de votos pode ser mitigado retirando a associação do usuário com seu voto, ou ainda criando um mecanismo que invalide o voto mas ainda assim seja gravado na *blockchain*.

A auditabilidade é completa, no final da divulgação dos resultados, é distribuído o banco de dados com todos os votos, todos poderão ver os votos, cada um pode validar seu próprio voto e contabilizar o voto do restante dos eleitores. Enquanto a votação estiver sendo realizada, um nodo completo pode ser gerido por uma auditoria, que poderá conferir em tempo real os votos.

O uso de código aberto foi primordial, pois a pesquisa utilizou de verba pública, também houve cuidado ao se escolher a licença de *copyright*, pois o código original do *bitcoin* é uma licença MIT, quer dizer que o código pode ser modificado e utilizado como quiser, sem necessitar de divulgação. A licença utilizada foi a GPLv3, ela dita que qualquer modificação deve ter seus fontes divulgados.

Foi utilizado *container docker* para a criação de ambientes de compilação e testes, isto inclusive esta de acordo com a licença, pois segundo a GPLv3 não se deve apenas divulgar o código fonte, mas também todo o necessário para que seja feita a compilação do mesmo.

Listagem 4.4 - Dados de uma transação pelo seu identificador


```
{
  "txid": "7d2572c5426faca62f37e6ab275fafd792869e9ee2f5...",
  "version": 1,
  "locktime": 0,
  "vin": [
      "txid": "0ecc9d88f8e78b210d907a14dd3b4a4f4ded2752948...",
      "vout": 502
  ],
  "vout": [
    {
      "n": 0,
      "scriptPubKey": {
        "reqSigs": 1,
        "type": "pubkey",
        "addresses": [
          "1WxbYxm7hJisS4kNnHnxCSnEsBywgspoywpWof"
        ]
      },
      "assets": [
          "name": "voto2019",
          "qty": 0.4,
          "type": "transfer"
        ]
    },
    {
      "n": 1,
        "reqSigs": 1,
        "type": "pubkey",
        "addresses": [
          "14ZGtgKWDaprxvD26JURkx8e1VWiemkqmPmLne"
        ]
      },
      "assets": [
          "name": "voto2019",
          "qty": 0.6,
          "type": "transfer"
        ]
    }
  ],
  "blockhash": "1e4624572f569c71f2e3800b380248a2b41b8f1858e52...",
  "confirmations": 1,
  "Date and time is": "Sun May 12 15:31:19 2019",
}
```


Fonte: Autor.

Listagem 4.5 - Dados de um bloco

```
{"method":"getblock","params":["3"],"id":1,"chain_name":"kk"}
{
    "hash" : "67d9bc0178e509a67ffe872daa24a602ec45a8b9...",
    "miner" : "1DWxicms4s6R3o6JTZCviJYpK3Z1EWoizzAYxj",
    "confirmations" : 3366,
    "size" : 838627985,
    "height" : 3,
    "version" : 3,
    "merkleroot" :
    ↪  "5646eba667e978cc11a187a6c4e86317f0c45b207d28572a00...",
    "tx number" : 2834,
    "time" : 1555469057,
    "Date: " : "Tue Apr 16 23:44:17 2019",
    "nonce" : 5,
    "bits" : "207fffff",
    "difficulty" : 4.656542374e-10,
    "chainwork" : "0000000000000000000000000000000000000008",
    "previousblockhash" :
    ↪  "42e1bdfd8e3eae207eea0e427aca0065f0a9ead0489c180a7e...",
    "nextblockhash" :
    ↪  "585df7ab48a9403d2d331220018b8c892afd6f6229ccee4792..."
}
```

Fonte: Autor.

5 RESULTADOS EXPERIMENTAIS

Os testes levaram em conta um número de eleitores de dezenas milhões de usuários e também o tempo de votação com as urnas eletrônicas atuais do Brasil, onde a apuração é feita no mesmo dia. Nos testes não foi possível passar da barreira de 45 milhões de transações por dia no hardware utilizado, melhores resultados são esperados com maior capacidade de processamento e velocidade de armazenamento. Por permitir o voto em qualquer dispositivo, além das zonas eleitorais, o processo de eleição pode se estender por vários dias sem encarecer os custos com infraestrutura. Também não há possibilidade de que a contagem seja divulgada pois cada eleitor tem acesso somente ao seu voto.

O processo de cada cliente se inicia ao conectar a um dos nodos principais através de uma chamada *HTTP-RPC*, autenticando com seu CPF e uma senha previamente estabelecida, em contrapartida recebe uma id de transação, indicando qual sua cédula e a localização da transferência para seu endereço dentro dessa transação, pois uma transação pode ter várias saídas pra destinatários diferentes, além disso receberá suas chaves público-privadas e o endereço. No cliente do eleitor há uma listagem de todos os candidatos e seus endereços associados.

5.1 Ambiente e testes realizados

Os testes foram feitos utilizando um servidor com:

- 32GB de memória RAM
- 2x Intel(R) Xeon(R) CPU E5620 @ 2.40GHz (8 núcleos físicos, 16 virtuais)
- Rede 100Mbit

A primeira máquina cliente:

- 12GB de memória RAM
- 2x Intel(R) Xeon(R) CPU E5620 @ 2.40GHz (8 núcleos físicos, 16 virtuais)
- Rede 100Mbit

A segunda máquina cliente:

- 4GB de memória RAM

- Intel® Core™ i7-4790 Processor @ 3.40GHz (4 núcleos físicos, 8 virtuais)
- Rede 100Mbit

O ambiente utilizado foi:

- Sistema operacional: ubuntu 18.10;
- Biblioteca: OpenSSL 1.0.8g;
- Biblioteca: BerkeleyDB 10.95;
- Bitcoin 10.5.3
- Multichain 2.0.1

Primeiramente foram distribuídos 1 milhão, 10 milhões e 100 milhões de votos, em três execuções sucessivas e retirada a média dos tempos. Depois foi feita a simulação de 50 milhões de votos em três execuções, na quarta execução foram incluídos mais 50 milhões de votos, de modo que se pudesse fazer a apuração com 100 milhões de votos. As apurações também foram feitas em 3 vezes sucessivamente e com os parâmetros do *OpenMP* modificados.

5.2 Testes com computação centralizada

Num sistema centralizado o cliente faz a requisição para o nodo para criar uma transação utilizando a transação de origem do voto e localização de seu próprio voto dentro dela, também deve indicar quais os destinos e quantidades. O retorno dessa requisição, que vem no formato de uma palavra-chave hexadecimal de comprimento de acordo com o número de destinatários, é assinado utilizando a chave-privada do eleitor. Isto garante a autenticidade da transferência. O próximo passo é outra requisição enviando essa transferência assinada. Os valores abaixo utilizaram os dados de 10 milhões de execuções de eleitores simulando votos para dois candidatos. Essas execuções foram divididas em 2000 instâncias simultâneas e 4 servidores em paralelo.

5.3 Testes com computação descentralizada

No sistema descentralizado, cada cliente cria a estrutura de sua transação, assina e somente utiliza o tempo dos nodos para envio. Como visto nas Tabelas 5.1 e 5.2, se essas operações forem feitas localmente, são 3 segundos a menos utilizando o nodo

Tabela 5.1 - Tempo de criação de transferência para dois destinatários

Mínimo	Máximo	Mediana	Média
0,00736s	8,79763s	1,62968s	1,73749s

Fonte: Autor.

Tabela 5.2 - Tempo de assinatura de transferência para dois destinatários

Mínimo	Máximo	Mediana	Média
0,00846s	8,35437s	1,37871s	1,51122s

Fonte: Autor.

(somando a mediana do tempo de assinatura (Tabela 5.2 e de criação (Tabela 5.1), para milhões de votos isto faz diferença na capacidade do sistema de transações simultâneas suportadas.

Os únicos tempos necessários para este modelo são os de envio dos clientes para os nodos completos. Os valores abaixo utilizaram os dados de 50 milhões de execuções de eleitores simulando votos para dois candidatos. Essas execuções foram divididas em 500 instâncias simultâneas em um único servidor. A Tabela 5.3 apresenta os tempos dessas execuções de envio.

Tabela 5.3 - Tempo de envio de uma transferência para dois destinatários

Mínimo	Máximo	Mediana	Média
0,007092s	35,231346s	0,799763s	1,064306s

Fonte: Autor.

5.4 Testes com compilação estática e com bibliotecas

Foram feitos testes compilando o código fonte de todas as dependências (OpenSSL[1], Boost++[2],BerkeleyDB[3]) e criando um executável estático. Mas não houve diferença

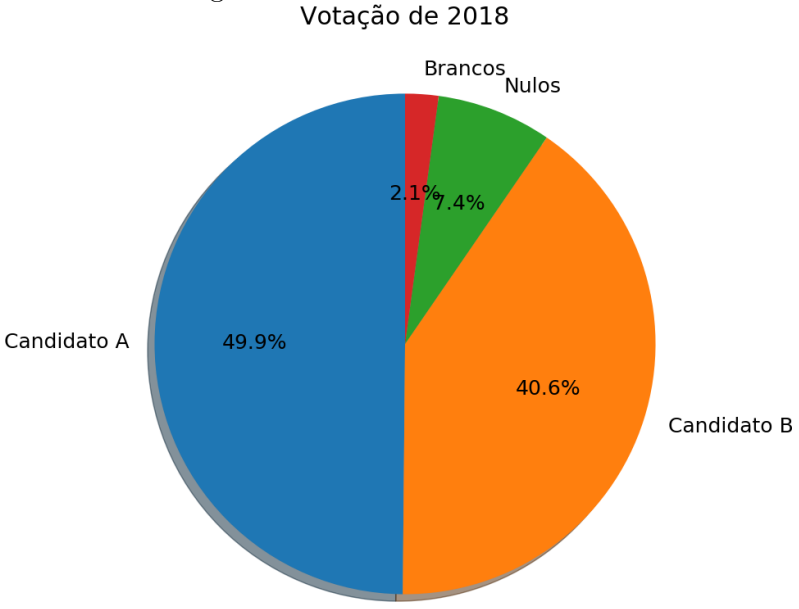
[1] https://www.openssl.org/
[2] https://www.boost.org
[3] https://www.oracle.com/technetwork/database/database-technologies/berkeleydb/downloads/index.html

significativa no desempenho, apenas é interessante para execução do software onde não haja permissão para instalar as bibliotecas necessárias.

5.5 Simulando processo eleitoral

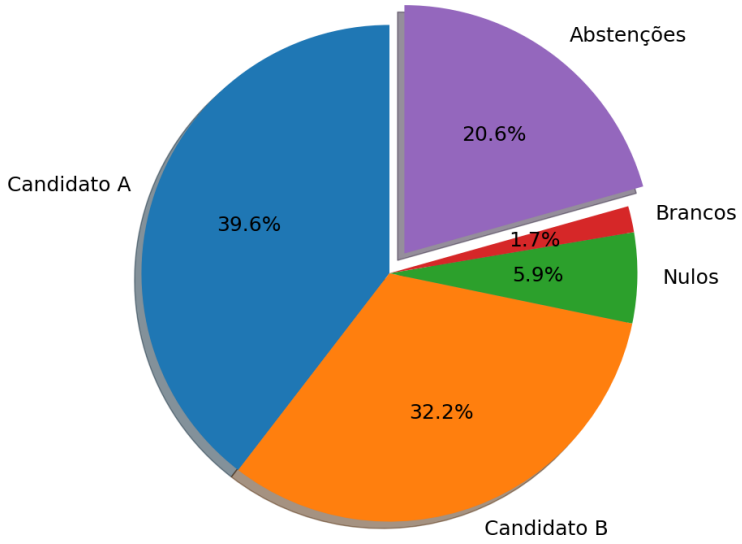
Para testes de desempenho do sistema foram levados em conta os números de eleitores do Brasil, Segundo Arrial (2018) em 2018 eram esperados 146 milhões de eleitores, porém na votação final foram contabilizados 116 milhões(Figura 5.1, havendo 11 milhões de brancos e nulos, portanto quase 20 milhões de abstenções. Em 2018 as eleições(MARTINS, 2018) ocorreram em 7 de outubro, o prazo para troca(CALEGARI, 2018) de zona eleitoral em 9 de maio e o pedido para voto em trânsito(RAMALHO, 2018) em 23 de agosto, ou seja a população teve menos de 1 mês e meio para poder requisitar o direito ao voto sem estar em sua zona eleitoral de registro ou ainda 5 meses para poder votar nos representantes de sua localidade atual. Com o sistema proposto nesta dissertação a localidade dos votos é irrelevante pois o eleitor poderá votar em qualquer dispositivo ou ainda em qualquer zona eleitoral. Essa desburocratização do voto poderia mudar o cenário das eleições, de acordo com a Figura 5.2 a parcela de eleitores que poderia ser beneficiada é significativa com 20% do total. Estudos sociais também são necessários para se criar melhores incentivos ao voto da população.

Figura 5.1 - Votos válidos de 2018.



Fonte: Jornal Gazeta do Povo (2018).

Figura 5.2 - Parcela de eleitores que poderiam votar no sistema proposto.



Fonte: Arrial (2018).

Os dois maiores colégios eleitorais são os estados de São Paulo e Minas Gerais, com 33 e 15 milhões de eleitores respectivamente com 32% e 14% do total do país. Logo a arquitetura proposta de um nodo *blockchain* por estado se torna viável.

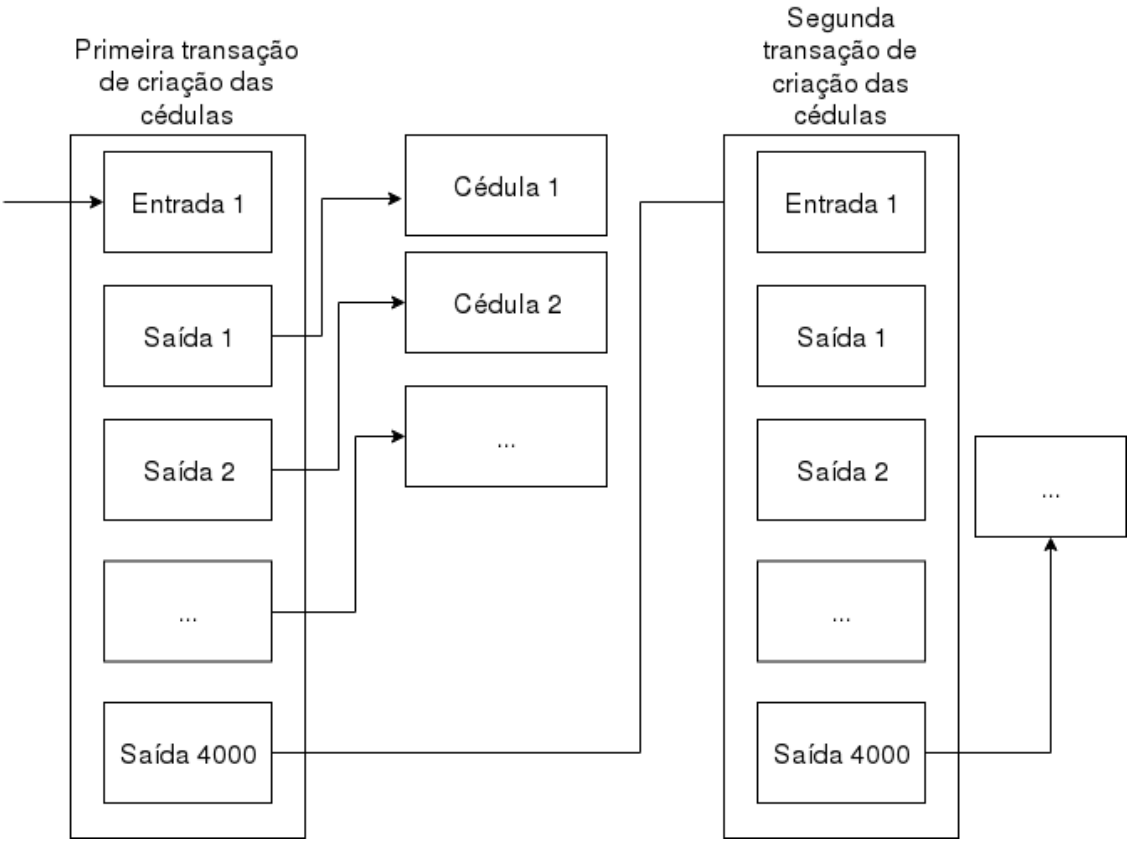
Os votos foram previamente preparados e assinados, trabalho que seria feito localmente por cada cliente, seja ele aplicativo de celular, página web ou programa de computador. A simulação foi a execução em paralelo de milhares de votos através de um cliente escrito em C++ em ambiente *linux*. O número de processos simultâneos foi afetado pela memória disponível na máquina de testes.

5.6 Distribuição

A distribuição dos votos consiste primeiramente da criação de uma transação especial com a quantidade total de votos a serem utilizados. Nenhum voto fora dessa transação pode ser computado pelos nodos. A partir de uma transação de origem, a próxima deve usar a quantidade exata de votos ou incluir uma saída extra com os votos restantes e um endereço. Há limitações no tamanho máximo do número de saídas da transação, nos testes o limite padrão de 4000 saídas foi utilizado. O processo de distribuição tem uma natureza obrigatoriamente sequencial, as primeiras 3999 cédulas são distribuídas entre os eleitores e a última saída para o endereço do

administrador com o total de votos menos esses 3999. Essa transação gera um identificador que é usado na próxima iteração. Isso se repete até que todos os eleitores tenham recebido e o endereço do administrador ficaria com o resto das cédulas não utilizadas. Na Figura 5.3 temos como se ocorre esta distribuição.

Figura 5.3 - Formato das transações para distribuição das cédulas.



Fonte: Autor.

Tabela 5.4 - Tempo para simulação da distribuição dos votos

Votos	Tempo	Número de transações
1 milhão	44,865s	251
10 milhões	509,859s	2501
100 milhões	7343,854s	25007

Fonte: Autor.

5.7 Simulando os votos

Tabela 5.5 - Tempo para simulação da execução dos votos

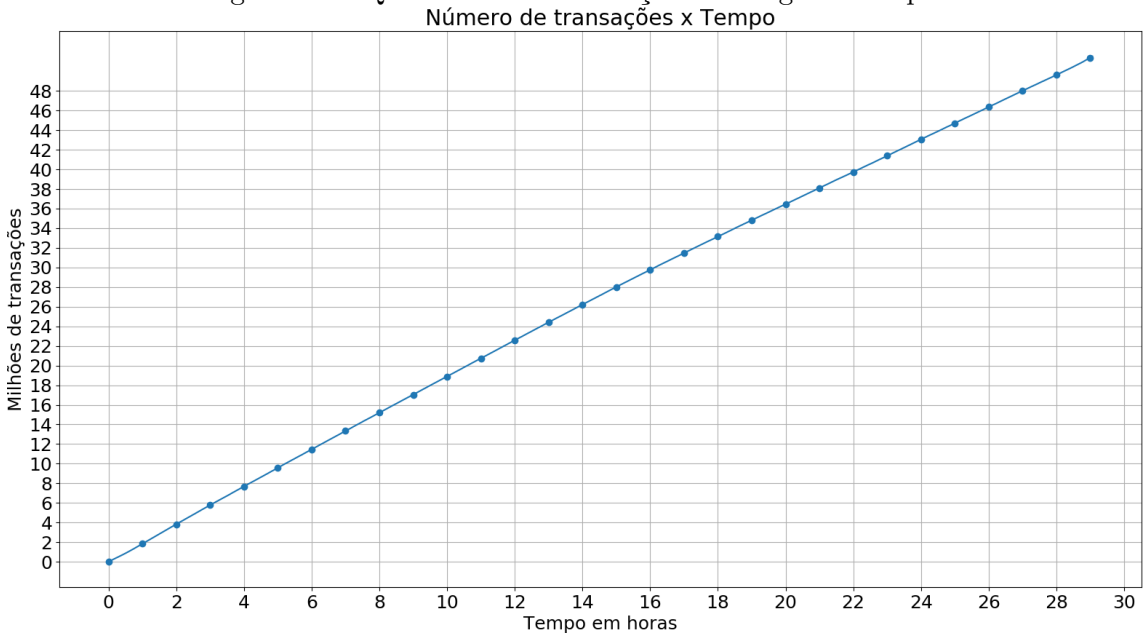
Milhões de votos	Tempo de execução
50	29h37m

Fonte: Autor.

Pela Tabela 5.5 temos que a simulação de 50 milhões de votos levou 29 horas.

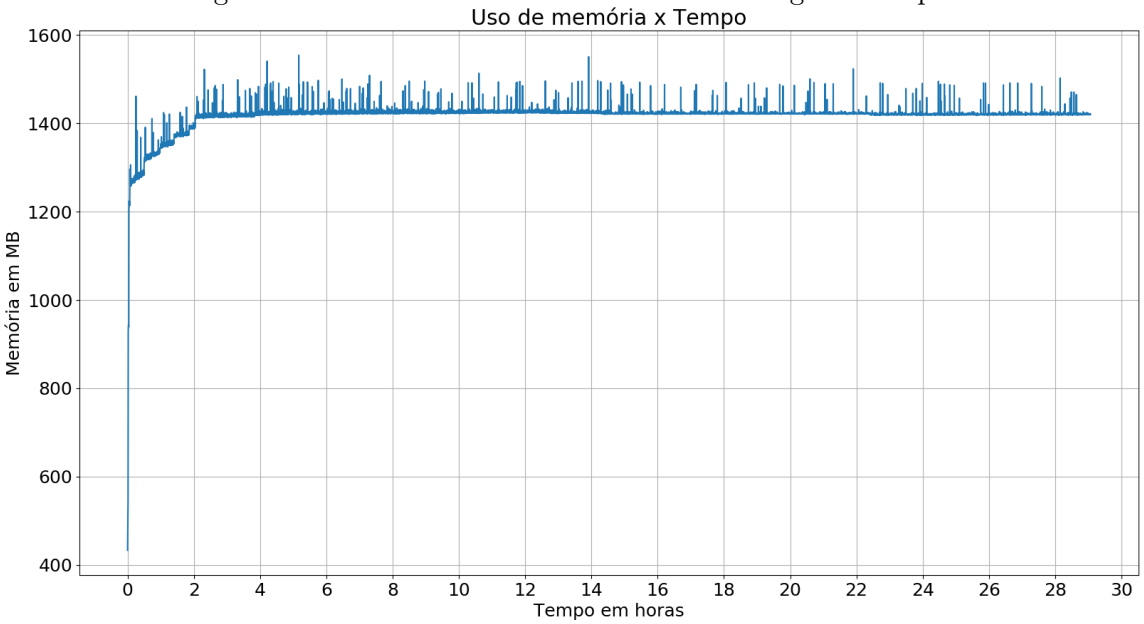
Pela Figura 5.4 temos que inicialmente o número de transações por hora era de 2 milhões, mas ao término de 24 horas haviam sido executados 43 milhões de votos, aproximadamente 10% a menos em relação ao esperado inicialmente. Isto se deve ao fato de que na criação dos blocos entrem travas de escrita de novas transações, as conexões entram em pausa. Somente ao final de 29h concluiu-se a simulação para 50 milhões de eleitores (Tabela 5.5).

Figura 5.4 - Quantidade de transações ao longo do tempo.



Fonte: Autor.

Figura 5.5 - Consumo de memória RAM ao longo do tempo.



Fonte: Autor.

Na Tabela 5.6 temos o tempo de execução individual de 50 milhões de votos, os quais foram divididos em 500 execuções simultâneas.

Tabela 5.6 - Tempo de execução de cada voto

Mínimo	Máximo	Mediana	Média
0,709s	35,231	0,799s	1,064s

Fonte: Autor.

5.8 Apuração

O processo de apuração é um problema massivamente paralelo, são milhões de operações independentes umas das outras, para tanto foram criadas funções otimizadas para processadores de vários núcleos. As informações das transações contém individualmente um volume de dados muito pequeno, são milhões de operações de curtíssimo tempo. Cada voto é assinado pela chave privada do eleitor, a apuração apenas faz a contagem dos mesmos. Pode realizar uma conferência completa dos

votos, ao calcular a hash das informações de cada voto e comparando com a hash gravada.

Este processo de apuração é paralelizado utilizando *OpenMP*, onde cada *thread* recebe um bloco com várias transações. Cada bloco pode conter um número variado de transações e cada transação tem um número variável de saídas, para cada uma dessas saídas teve de ser feita a contabilidade. O escalonamento do trabalho e do número das *threads* é dinâmico por conta dessa variação. Um histograma do número de transações por bloco pode ser visto na Figura 5.6, estes dados são da simulação de 50 milhões de votos. Pelo fato das informações gravadas no *blockchain* já terem sido extensiva e previamente verificadas (ver 4.4.13), os votos precisam ser apenas lidos e somados. Duas informações são necessárias: Endereço de destino e quantidade. Isto permite que a contagem seja feita rapidamente. Os tempos de apuração são proporcionais ao número de *threads*, na Tabela 5.7 temos que o tempo de execução ótimo foi alcançado com um número de threads do processo igual ao número de threads do processador, como se poderia esperar. O processador utilizado possuía 16 threads.

Tabela 5.7 - Tempo para apuração de 50 milhões de votos

Número de threads	Tempo de execução
1	1531,356s
2	770,941s
4	389,177s
8	205,703s
16	176,852s
32	177,400s

Fonte: Autor.

A Tabela 5.9 apresenta uma análise com um maior número de votos e testes com o controle do número de *threads* e se os *loops*(ver Listagem 4.2) são aninhados ou não. *Thread* dinâmica é a possibilidade do *OpenMP* definir se o número de *threads* é fixo ou varia com o trabalho disponível, o número de blocos restantes é proporcional ao número de *threads*. *Nested for* é o parâmetro que indica se os *loops for* internos serão estendidos.

Tabela 5.8 - Tempo para apuração com diferentes populações

Número de votos	Tempo de execução
10 milhões	31,293s
50 milhões	176,852s
100 milhões	334,332s

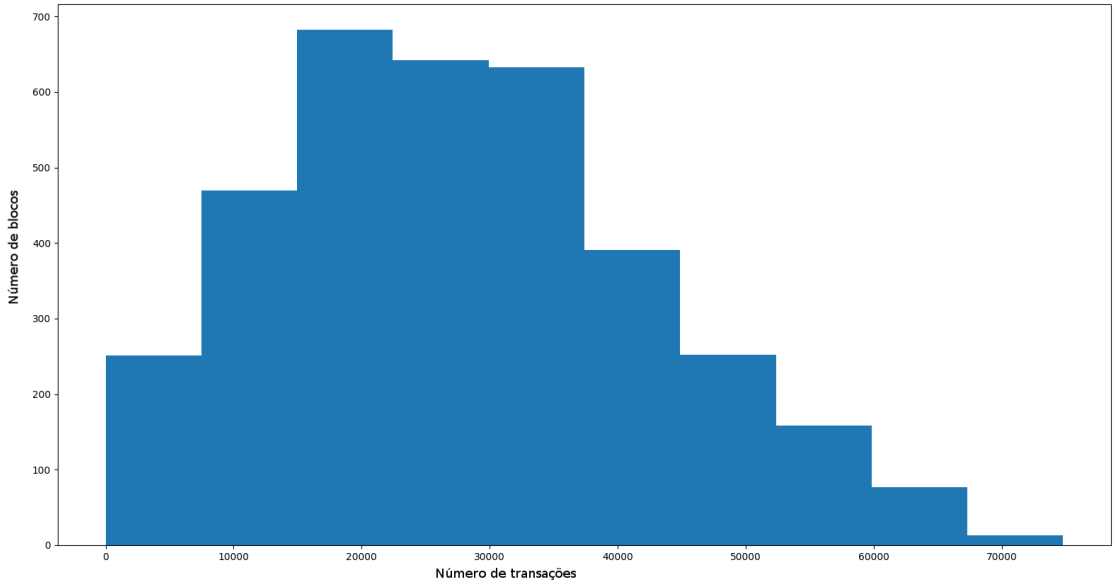
Fonte: Autor.

Tabela 5.9 - Tempo para apuração de 100 milhões de votos

Número de threads	Tempo de execução	*threads* dinâmicas	*nested for*
1	2929,786s	não	não
1	2974,318s	não	sim
1	2980,663s	sim	sim
1	3007,775s	sim	não
2	1480,545s	não	não
2	1479,847s	não	sim
2	1477,922s	sim	sim
2	1484,423s	sim	não
4	742,964s	não	não
4	742,745s	não	sim
4	744,624s	sim	sim
4	743,944s	sim	não
8	391,053s	não	não
8	390,864s	não	sim
8	391,197s	sim	sim
8	416,542s	sim	não
16	334,332s	não	não
16	349,854s	não	sim
16	340,334s	sim	sim
16	350,503s	sim	não
32	406,661s	não	não
32	353,165s	não	sim
32	338,059s	sim	sim
32	338,623s	sim	não

Fonte: Autor.

Figura 5.6 - Histograma do número de transações por bloco.(50 milhões de votos)



Fonte: Autor.

5.9 Auditoria

Segundo Rivest e Wack (2006), um dos autores da chave de criptografia RSA e autor do Princípio da Independência de Software em Sistemas Eleitorais: "Um sistema eleitoral é independente do software se uma modificação ou erro não-detectado no seu software não pode causar uma modificação ou erro indetectável no resultado da apuração." De acordo com este princípio, o presente trabalho atende, pois sua implementação pode ser feita em outra linguagem ou arquitetura desde que respeitando as regras do *blockchain*, usando as mesmas funções matemáticas, regras do protocolo e com as mesmas regras de transação. Temos um exemplo em (SHIRRIFF, 2014), onde os algoritmos do *bitcoin* são executados com papel e lápis.

Além das regras e protocolos poderem ser escritos em qualquer linguagem, a estrutura das transações e dos blocos gravados atende esta exigência. Isto garante que a modificação de algum voto, depois de enviado pelo usuário, invalide a *blockchain*. Pois cada transação faz parte de uma árvore Merkle (ver seção 2.2.3). Além disso, cada bloco leva em consideração o bloco anterior para calcular sua *hash* (Ver seção 2.2.2). Caso haja modificação em um único voto, todos os votos realizados após este serão invalidados. Como um castelo de cartas de baralho, onde uma carta retirada da base afetaria toda a estrutura.

Se todo o processo de voto e apuração fosse feito sem utilizar software, com apenas papel e caneta. Cada eleitor receberia uma transação de origem, sua cédula, nela escreveria seu voto. Respeitando o formato de origem, destino, quantidade e no final calculando a assinatura desse conteúdo utilizando sua chave privada. Este papel se assemelharia a um email assinado com PGP. Enviando estes votos para outras pessoas, órgãos do governo, auditores, estes iriam calcular as árvores *Merkle* destas transações e criar os blocos de votos, cada bloco referenciando o bloco anterior. Este processo seria dolorosamente lento, mas ao final teríamos o mesmo resultado que utilizando software. Caso um dos votos fosse retirado e manipulado, fazendo o processo de cálculo da *hash Merkle* de onde este voto faz parte teríamos que a *hash* não coincide com a *hash* armazenada, logo o voto seria inválido e não haveria garantias para os outros.

Para que a população possa efetuar a auditoria, as informações a serem transmitidas precisam ser compactas o suficiente para uma transmissão eficiente. O código fonte compactado de todo projeto fica abaixo de 10MB, isto inclui os fontes originais do *bitcoin* e *multichain* mais modificações e melhorias implementadas pelo autor. A imagem docker do ambiente de compilação com todas as bibliotecas instaladas ocupa 600MB instalado. A base de dados cresce com o número de transações como visto na Figura 5.10. Isto permite que os dados sejam transmitidos sem muita dificuldade, tanto o código fonte quanto as bases de dados de cada eleição podem ser auditados.

Tabela 5.10 - Tamanho das bases de dados

Número de votos em milhões	Tamanho em MB
1	630MB
10	5400MB
35	28000MB
50	35000MB
100	55000MB

Fonte: Autor.

5.10 Análise dos resultados

Pelo monitoramento da implementação temos que inicialmente o número de transações por hora era de 2 milhões, mas ao término de 24 horas haviam sido executados 43 milhões de votos, aproximadamente 10% a menos em relação ao esperado inicialmente. Esta limitação de 43 milhões de votos diários por servidor poderia ser

remediada utilizando de vários nodos, que trocariam entre si blocos e transações, havendo uma distribuição de carga. Também há a possibilidade de que se crie vários *blockchains* delimitados por região física, tal que o eleitor que vá votar para governador não precisaria ter acesso ao *blockchain* de outros estados. Para categorias mais abrangentes como presidente também é possível que os votos sejam feitos em separado e na apuração de cada região seja contabilizada no geral.

Também temos que o consumo de memória se manteve constante por volta de 1,5GB, conforme Figura 5.5. O tempo para recebimento e consolidação das transações em blocos pode ser melhorado, utilizando um particionamento dos eleitores em diferentes servidores. O o resultado foi com um único servidor, a vantagem de se utilizar o código do *bitcoin* é que já se tem a implementação para redistribuir transações e blocos, efetivamente criando uma rede distribuída. Também pode-se criar várias *blockchains* de acordo com o tamanho da população, uma por estado, ou cidade, até mesmo bairro. A indicação de para qual nodo o eleitor seria direcionado ficaria logo após a identificação e envio da chave privada e *txid* da cédula. Na apuração o processo é feito em paralelo entre as várias *blockchains* regionais, nelas os resultados serão ainda mais rápidos pelo menor número de blocos em cada servidor. Em comparação com as 5 horas(GLOBO, 2018), do sistema atual de voto brasileiro, o ganho foi considerável, onde um único servidor conseguiu apurar de forma correta 100 milhões de votos em pouco mais de 400 segundos. Os tempos para execução do código, que executa o voto, também são baixos, onde a mediana ficou em 0,8 segundos (Tabela 5.6, isto possibilitaria a execução em dispositivos com pouco poder de processamento. A distribuição de 100 milhões de cédulas levou pouco mais de 2 horas. Este tempo seria no pior caso, utilizando várias *blockchains* distribuídas em estados, cidades, bairros, o processo seria mais rápido ainda.

Uma comparação com o sistema atual brasileiro lista as melhorias alcançadas (Figura 5.11). Utilizamos os termos obscuro, ou obscurantismo, segundo Brunazo Filho (2001) pela segurança se basear no sigilo. A eficiência refere-se ao tempo de apuração. Manutenibilidade refere-se à necessidade de manutenção dos códigos-fonte, protocolos e algoritmos de criptografia; consideramos baixa a do sistema proposto pelo fato de ser código aberto com um grande número de usuários e desenvolvedores, fazendo com que novas mudanças e melhorias possam ser facilmente integradas, dispensando a necessidade de uma grande equipe própria para manter o sistema. Portabilidade refere-se à capacidade de funcionar em diferentes dispositivos. Auditabilidade refere-se à capacidade de ter seus processos, código-fonte, algoritmos e principalmente pelo resultado das eleições disponíveis para uma avaliação pública.

Transparência refere-se à possibilidade de apurações individuais e disponibilidade de todo código utilizado. Usabilidade é o fato de poderem ser criados clientes nas mais diferentes plataformas e permitir que o eleitor vote de forma prática.

Tabela 5.11 - Comparativo entre o sistema atual no Brasil e utilizando *blockchain*

Características	Sistema atual brasileiro	Sistema proposto utilizando *blockchain*
Eficiência	Eficiente	Mais eficiente
Segurança	Obscura	Alta
Manutenibilidade	Alta	Baixa
Portabilidade	Baixa	Alta
Auditabilidade	Obscura	Alta
Transparência	Obscura	Completa
Usabilidade	Baixa	Alta

Fonte: Autor.

6 OUTRAS APLICAÇÕES POSSÍVEIS

O *blockchain* tem diversas aplicações além da sua função original de moeda virtual, elas vão desde o mercado financeiro, até *IoT*, ou internet das coisas, onde micro clientes com acesso à internet se utilizam da natureza descentralizada e de registro permanente para enviar e receber instruções. São como se fosse um grande banco de dados distribuído e criptografado, novas informações podem ser incluídas apenas e não modificadas. Há também o uso de contratos inteligentes, são informações codificadas de um acordo entre duas partes, as identidades são preservadas através do uso de endereços. Quando o gatilho do contrato é atingido, seja um preço alcançado em uma ação, ou uma data determinada é alcançada, o contrato é executado. Entidades reguladoras podem auditar o registro deste contrato e entender o movimento do mercado com a privacidade dos indivíduos preservada.

Segundo Casino et al. (2019) temos o uso crescente de aplicações utilizando *blockchain* nas mais diversas indústrias (Figura 6.1).

6.1 Banco de dados Amazônia

Carlos Nobre pretende criar um banco de dados sobre a floresta amazônica utilizando *blockchain*, o sequenciamento genético da fauna e flora que resultarem em desenvolvimento econômico deve ser guardado dentro da *blockchain*.(DRONES. . . , 2018)

6.2 Blockchain CubeSat

Os satélites geoestacionarios estão localizados a 35.786 km acima da linha do Equador, seguindo a rotação da Terra. Os satélites de órbita baixam atuam entre 600 e 800 km. Entre estas duas categorias existem outros, como os que fazem parte do sistema GPS. O equipamento de transmissão precisa de uma potência adequada à sua distância da Terra, quanto mais longe mais necessariamente potente. A idéia de se utilizar uma rede *blockchain* para comunicação é diminuir a necessidade de potência do hardware de transmissão, utilizando uma rede de satélites comunicantes entre si. Os *CubeSats* são candidatos a este uso pelo fato de serem satélites de tamanho restritivo, onde espaço físico e peso contam. A ideia de se utilizar um *blockchain* no espaço, com CubeSats, é a de que o hardware necessário para comunicação satélite-satélite necessita ser menos potente do que o que faria a comunicação satélite-Terra. Logo uma constelação de satélites utilizaria a *blockchain* para propagar dados, de forma segura e imutável. Como todos os satélites teriam a mesma versão da *blockchain*, o satélite com hardware capaz de enviar os dados à estação de controle na

Figura 6.1 - Áreas com aplicações para o blockchain.

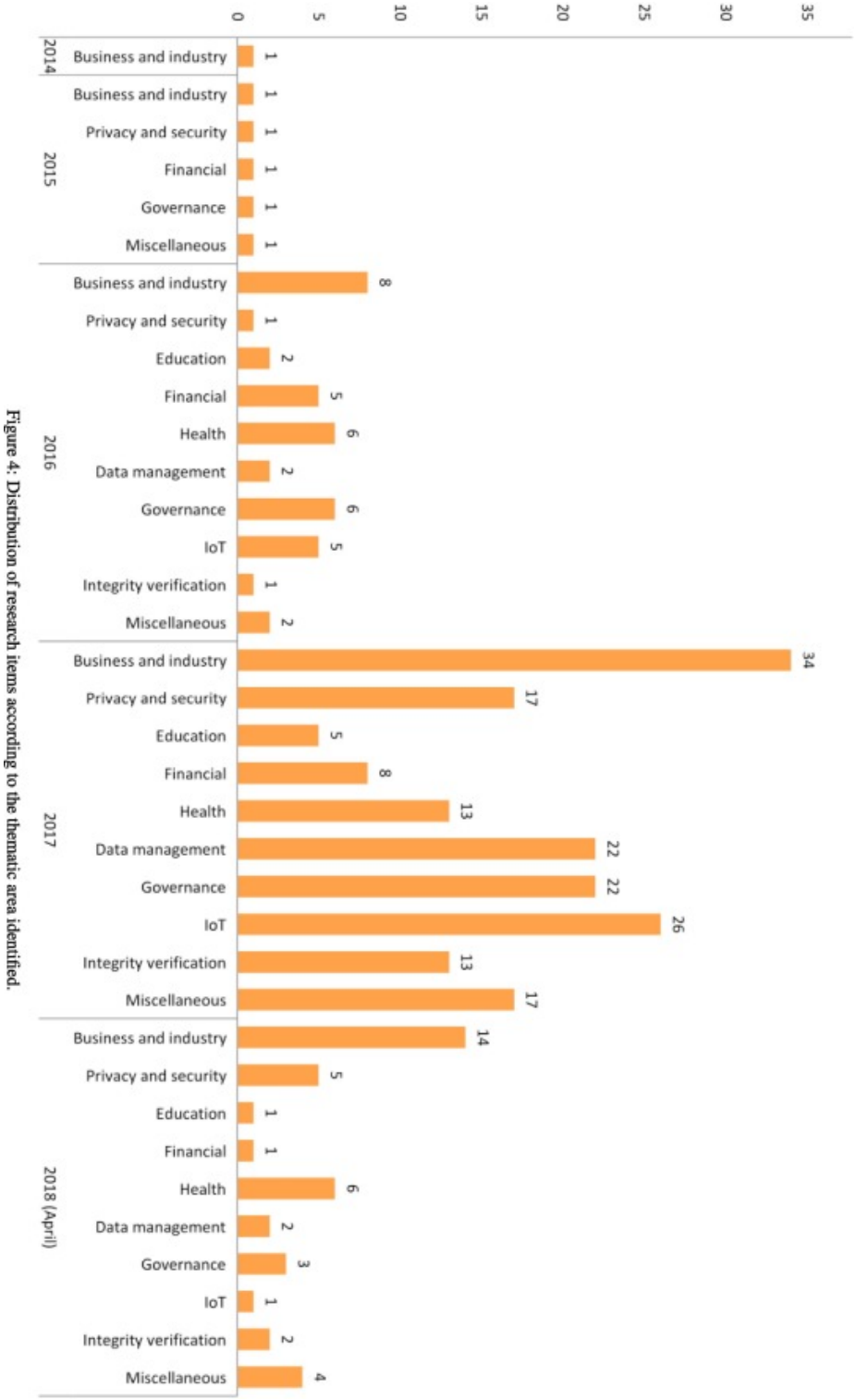


Figure 4: Distribution of research items according to the thematic area identified.

Fonte: Casino et al. (2019)

Terra faria a transmissão. Seria uma constelação de satélites com comunicação local e um ou mais que teriam comunicação com a Terra.

6.3 Blockchain Vant Swarm

O conceito de comunicação local também pode ser aplicado em um grupo de *Vant's* voando em conjunto, os dados seriam trocados localmente e enviados a uma estação de controle por um único aparelho, com hardware suficientemente potente. Mas todos teriam uma cópia completa dos dados em forma de *blockchain*. Isto já esta sendo explorado (MCNABB, 2018).

6.4 Blockchain IoT Sensors

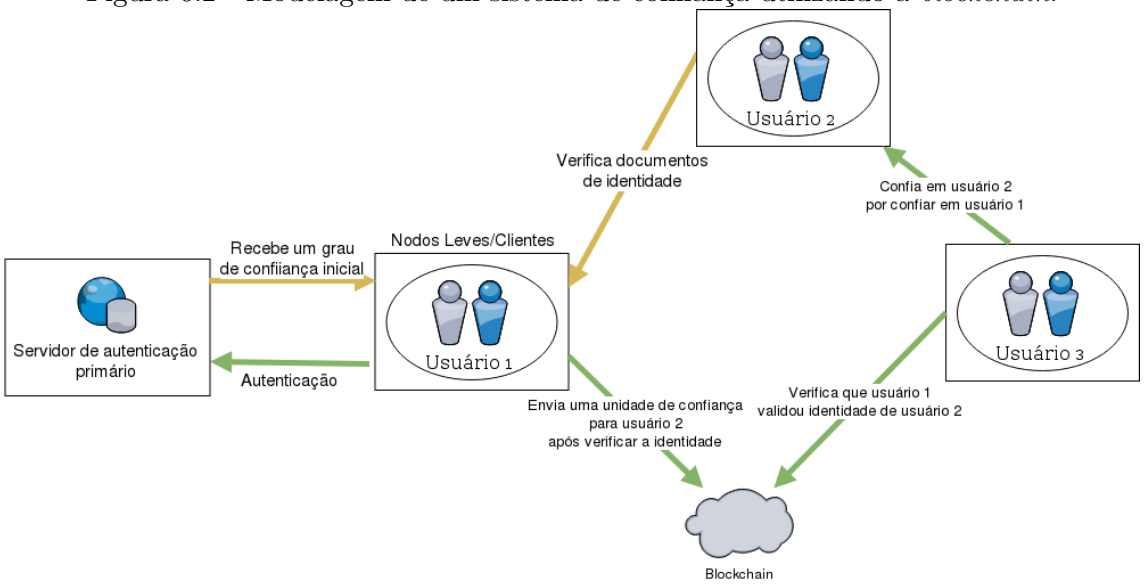
O conceito de uma rede de sensores que se comuniquem localmente, onde cada um guarda uma cópia de todo o histórico de dados e transações, pode ser aplicado na Internet das coisas *IoT*, onde há a comunicação de diferentes hardwares, com um denominador comum que permita a transmissão e armazenamento da *blockchain*. Um único nodo desta rede poderia transmitir toda a *blockchain* com hardware mais potente. A manutenção de um histórico distribuído e paralelo é proposto em Huang et al. (2018). Também existe a possibilidade de utilizar etiquetas de identificação *RFID* em equipamentos e objetos, atrelando esta identificação a um *token, asset* em uma rede *blockchain*; quando o objeto com a etiqueta passar por sensores instalados pelo caminho, haverá o registro da movimentação dentro da *blockchain*, copiando o movimento físico e criando um registro virtual.

6.5 Blockchain Web-Of-Trust

A ideia de uma web-of-trust é reunir e encontrar pessoas fisicamente no mesmo local, conferir identidade e chave-pública, assinar a chave do indivíduo e assim confirmar a pessoa. As pessoas que confiarem em quem assinou a chave podem confiar em quem recebeu a assinatura, não precisando necessariamente encontrar pessoalmente esta nova pessoa, criando assim uma rede de confiança. Com o uso de *blockchain*, cada pessoa recebe uma quantidade de confiança que pode ser distribuída, esta confiança é única e não pode ser copiada, enviada em duplicata ou criada de forma aleatória. O destinatário escolhe a quem enviar através do mesmo processo, conferindo documentos e encontrando pessoalmente. Esta confiança é modelada através da criação de uma transação com nome e quantidade. Cada transação que se envia para uma chave pública é assinada com a chave privada do remetente, assim o destinatário teria uma id de transação de várias pessoas as quais o verificaram; quanto mais

transações ele for recipiente, mais confiança de que ele é quem diz ser. A Figura 6.2 exemplifica esta aplicação.

Figura 6.2 - Modelagem de um sistema de confiança utilizando a *blockchain*.



Fonte: Autor.

7 CONCLUSÃO

O objetivo deste trabalho foi propor, implementar e testar um sistema de votação eletrônico, baseado em *blockchain*, com código livre e uma licença que obriga qualquer trabalho derivativo a disponibilizar suas modificações (AGPLv3). Isso faz com que tanto o código como a base de votos possam ser auditados por qualquer um.

O alvo desta implementação foi utilizar a *blockchain*. Esta tecnologia funciona da seguinte forma: cada transação ou voto é gravado de forma controlada e a cada novo voto a segurança do sistema como um todo é incrementada, pois para mudar uma transação passada, seria necessário refazer todas as transações subsequentes e isso resultaria não só em recalcular como ter controle dos nodos que consentem um novo bloco dentro da *blockchain*.

O sistema de urna eletrônica no Brasil tem as seguintes diferenças com o trabalho apresentado:

a) Na presente proposta, os votos são transmitidos por rede, podendo inclusive utilizar uma rede não segura como a internet.

b) Nesta implementação, os votos são transmitidos no mesmo instante em que são realizados, e após inclusão na *blockchain* não há chance alguma de modificação. Caso utilize-se a própria cédula do eleitor com sua chave privada, o eleitor descobriria após tentar realizar o voto e descobrir que este já foi feito.

c) Necessidade de comparecimento a uma zona eleitoral. Na presente proposta podemos ter softwares de voto funcionando em celulares e computadores pessoais, cada eleitor será responsável exclusivamente por seu voto, não sendo possível que manipule os votos de outra pessoa, pois não terá a chave privada nem a cédula do outro.

d) Atualmente, a auditoria é feita por grupos convidados pelos órgãos competentes, sendo feita exclusivamente no software e apenas no processo de voto, não na contabilização; uma auditoria em tempo real em época de eleição não é possível. Nesta proposta, qualquer pessoa que tenha acesso à *blockchain* pode conferir em tempo real a inclusão de todos votos, embora a divulgação desses dados alterasse o resultado. Por isto o acesso deve ser apenas de auditores autorizados.

e) No sistema brasileiro um eleitor não pode verificar se seu voto foi computado corretamente. Mesmo que haja a impressão de um comprovante da escolha feita, o eleitor não tem como verificar que seu voto foi incluso no total. Na nossa proposta, o voto fica exposto e pode ser verificado pelo usuário de forma anônima. Em casos de coerção ou compra de votos é necessário uma solução futura.

f) No processo de voto atual, a privacidade do eleitor é garantida apenas pela câmara de votação. Na nossa proposta, o eleitor poderia votar de qualquer lugar. Sua privacidade na auditoria é garantida pelo uso de chave pública identificando seu voto. Apenas quem distribuiu o voto sabe a relação entre chave pública e identidade.

As caracerísticas acima foram alcançados utilizando a tecnologia de *blockchain*, um banco de dados criado através de uma rede distribuída, transparente e auditável.

Os resultados realizados foram animadores, se comparado a trabalhos que utilizem a *blockchain* pública do *bitcoin*, que teve uma média diária de 380 mil transações e a *blockchain* do *Ethereum* com 900 mil (Figura 3.1). O protótipo criado neste trabalho obteve 43 milhões utilizando a mesma estrutura de segurança do *bitcoin* com um processador lançado em 2010[1].

7.1 Trabalhos futuros

Alguns dos trabalhos que podem ser implementados com base neste, são testes em hardware mais eficiente, com maior número de instruções por *clock*(IPC) e em maior escala. Criação de novas interfaces gráficas e novos clientes em diferentes plataformas. Caso os algoritmos de segurança utilizados se tornem obsoletos, a troca seria necessária, mas isto acompanharia o desenvolvimento do *bitcoin*, sendo implementado por especialistas em segurança. Outros trabalhos relacionados a voto e blockchain podem ser desenvolvidos em diferentes plataformas, de acordo com Dinh et al. (2017) a *blockchain Hyperledger Fabric* apresenta as melhores características para a realização de trabalhos com voto; como a velocidade de processamento das transações, contratos inteligentes e propagação dos blocos. Embora não tenha sido testada como o *bitcoin*.

Esta arquitetura, alvo deste trabalho, buscou principalmente aumentar a transpa-

[1] https://ark.intel.com/content/www/us/en/ark/products/47925/intel-xeon-processor-e5620-12m-cache-2-40-ghz-5-86-gt-s-intel-qpi.html

rência no processo de voto. Problemas com os dispositivos de cada eleitor, voto sob coerção e venda de votos ficaram de fora do escopo, há medidas para mitigar, mas não para eliminá-las.

REFERÊNCIAS BIBLIOGRÁFICAS

ADIDA, B. Helios: web-based open-audit voting. In: CONFERENCE ON SECURITY SYMPOSIUM (SS'08). USENIX ASSOCIATION. **Proceedings...** [S.l.], 2008. p. 335–348. 9, 11

ADIPUTRA, C. K.; HJORT, R.; SATO, H. A proposal of blockchain-based electronic voting system. In: WORLD CONFERENCE ON SMART TRENDS IN SYSTEMS, SECURITY AND SUSTAINABILITY (WorldS4). **Proceedings...** [S.l.], 2018. p. 22–27. 23

AKBARI, E.; WU, Q.; ZHAO, W.; ARABNIA, H. R.; YANG, M. Q. From blockchain to internet-based voting. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE AND COMPUTATIONAL INTELLIGENCE (CSCI). **Proceedings...** [S.l.], 2017. p. 218–221. 33

ALENCAR, M. F. de; CORREA, M. **Ethervoltz: um sistema de votação auditável baseado no blockchain ethereum**. Monografia (Graduação em Engenharia da Computação) — ETEP Faculdades, São José Dos Campos, 2017. Disponível em: <http://www.inicepg.univap.br/cd/INIC_2017/anais/arquivos/RE_0475_1017_03.pdf>. 24

ALONSO, L. P.; GASCO, M.; BLANCO, D. Y. M.; ALONSO, J. A. H.; BARRAT, J.; MORETON, H. A. E-voting system evaluation based on the council of europe recommendations: Helios voting. **IEEE Transactions on Emerging Topics in Computing**, p. 1–1, 11 2018. 9

ALSUNAIDI, S. J.; ALHAIDARI, F. A. A survey of consensus algorithms for blockchain technology. In: INTERNATIONAL CONFERENCE ON COMPUTER AND INFORMATION SCIENCES (ICCIS). **Proceedings...** [S.l.], 2019. p. 1–6. 24

ANANDARAJ, S.; ANISH, R.; DEVAKUMAR, P. Secured electronic voting machine using biometric. In: INTERNATIONAL CONFERENCE ON INNOVATIONS IN INFORMATION, EMBEDDED AND COMMUNICATION SYSTEMS (ICIIECS). **Proceedings...** 2015. p. 1–5. Disponível em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7192976&isnumber=7192777>. 12

ARAUJO, R.; FOULLE, S.; TRAORé, J. A practical and secure coercion-resistant scheme for remote elections. In: INTERNATIONALES BEGEGNUNGS- UND

FORSCHUNGSZENTRUM FüR INFORMATIK. **Frontiers of Electronic Voting**. 2008. Disponível em: <http://drops.dagstuhl.de/opus/volltexte/2008/1295>. 24

ARRIAL, T. **Eleitorado de 2018**. Confederação Nacional de Municípios, 2018. Disponível em: <https://www.cnm.org.br/cms/biblioteca/Eleitorado-2018.pdf>. 58, 59

BACK, A. **Hashcash - a denial of service counter-measure**. 2002. Disponível em: <http://www.hashcash.org/papers/hashcash.pdf>. 13

BARTOLUCCI, S.; BERNAT, P.; JOSEPH, D. SHARVOT: secret SHARe-based VOTing on the blockchain. In: INTERNATIONAL WORKSHOP ON EMERGING TRENDS IN SOFTWARE ENGINEERING FOR BLOCKCHAIN (WETSEB). **Proceedings...** [S.l.]: IEEE/ACM, 2018. p. 30–34. 44

BATISTA, V. **Peritos criminais da PF avaliam código fonte da urna eletrônica nesta semana - blog do servidor**. 2018. Disponível em: <http://blogs.correiobraziliense.com.br/servidor/peritos-criminais-da-pf-avaliam-codigo-fonte-da-urna-eletronica-nesta-semana/>. 4

BENALOH, J. Simple verifiable elections. In: ACCURATE ELECTRONIC VOTING TECHNOLOGY. **Proceedings...** USENIX Association, 2006. (EVT'06), p. 5–5. Disponível em: <http://dl.acm.org/citation.cfm?id=1251003.1251008>. 12

BISTARELLI, S.; MANTILACCI, M.; SANTANCINI, P.; SANTINI, F. An end-to-end voting-system based on bitcoin. In: SYMPOSIUM ON APPLIED COMPUTING. **Proceedings...** New York, NY, USA: ACM, 2017. (SAC '17), p. 1836–1841. ISBN 978-1-4503-4486-9. Disponível em: <http://doi.acm.org/10.1145/3019612.3019841>. 23

BITINFOCHARTS. **Bitcoin, Ethereum Transactions chart**. 2019. Disponível em: <https://bitinfocharts.com/>. 24, 25, 31, 32, 33, 34, 35

BLOCKCHAIN. **Blockchain - the most trusted crypto company**. 2019. Disponível em: <https://www.blockchain.com/>. 50

BLOCKCHAIR. **Blockchair — universal blockchain explorer and search engine**. 2019. Disponível em: <https://blockchair.com/>. 49

BROWN, R. G.; CARLYLE, J.; GRIGG, I.; HEARN, M. **Corda: an introduction**. 2016. Disponível em: <https://docs.corda.net/_static/corda-introductory-whitepaper.pdf>. 35

BRUNAZO FILHO, A. Critérios para avaliação da segurança do voto eletrônico. In: WORKSHOP EM SEGURANÇA DE SISTEMAS COMPUTACIONAIS, WSEG. **Anais...** 2001. (Wseg'2001). Disponível em: <http://www.brunazo.eng.br/voto-e/textos/Wseg2001.htm>. 2, 3, 4, 67

_____. **Modelos e gerações dos equipamentos de votação eletrônica**. 2014. Disponível em: <http://www.brunazo.eng.br/voto-e/textos/modelosUE.htm>. 1, 2

BTC. **Bitcoin block explorer - btc.com**. 2019. Disponível em: <https://btc.com/>. 48

BUTERIN, V. **Ethereum : a next-generation smart contract and decentralized application platform**. 2015. Disponível em: <https://www.weusecoins.com/assets/pdf/library/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf>. 20, 22

CALEGARI, L. **Prazo para agendar pedido ou transferência de título eleitoral acaba sexta**. 2018. Disponível em: <https://exame.abril.com.br/brasil/prazo-para-pedir-ou-transferir-titulo-eleitoral-termina-nesta-sexta/>. 58

CASINO, F.; DASAKLIS, T. K.; PATSAKIS, C. A systematic literature review of blockchain-based applications: Current status, classification and open issues. **Telematics and Informatics**, v. 36, p. 55–81, mar. 2019. ISSN 07365853. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0736585318306324>. 22, 69, 70

CLARKSON, M. R.; CHONG, S.; MYERS, A. C. Civitas: toward a secure voting system. In: SYMPOSIUM ON SECURITY AND PRIVACY (SP 2008). **Proceedings...** [S.l.]: IEEE, 2008. p. 354–368. 9, 10, 11

COOLEY, R.; WOLF, S.; BOROWCZAK, M. Blockchain-based election infrastructures. In: INTERNATIONAL SMART CITIES CONFERENCE (ISC2). **Proceedings...** [S.l.]: IEEE, 2018. p. 1–4. 23

CUNHA, S. S. d.; MARCACINI, A. T. R.; CORTIZ, M. A.; FERNANDES, C. T.; STOLFI, J.; REZENDE, P. A. D. d.; BRUNAZO FILHO, A.; MOURA, F. V. d.; CARVALHO, M. A. M. d.; TEIXEIRA, M. C. **Relatório sobre o Sistema Brasileiro de Votação Eletrônica**. Comitê Multidisciplinar Independente, 2014. Disponível em: <http://www.brunazo.eng.br/voto-e/textos/CMind-1-Brasil-2010.pdf>. 5

DIAS, S. P. **Proposta de sistema de votação eletrônica auditável para instituições de ensino superior**. Dissertação (Mestrado em Sistemas de informação e gestão do conhecimento) — Universidade FUMEC - Fundação Mineira de Educação e Cultura, 2016. Disponível em: <http://www.fumec.br/revistas/sigc/article/view/4070/0>. 25

DINH, T. T. A.; WANG, J.; CHEN, G.; LIU, R.; OOI, B. C.; TAN, K.-L. **BLOCKBENCH: a framework for analyzing private blockchains**. 2017. Disponível em: <https://arxiv.org/abs/1703.04057>. 25, 74

DRONES + Blockchain = Combining Two Exciting Technologies. 2018. Disponível em: <https://dronesonvideo.com/drones-and-blockchain-technology/>. 69

FERNANDES, C. **Estudo e avaliação tecnológica dos dados oficiais da eleição de alagoas 2006 1o turno**. 2006. Disponível em: <http://www.brunazo.eng.br/voto-e/arquivos/AL06-laudoFerITA.zip>. 2

GLOBO. **Resultado da apuração de 2º turno para presidente das Eleições 2018.** 2018. Disponível em: <https://g1.globo.com/politica/eleicoes/2018/apuracao/presidente.ghtml>. 67

GRAAF, J. V. d. O mito da urna. 2017. Disponível em: <https://inscrypt.dcc.ufmg.br/wp-content/uploads/2017/11/o-mito-da-urna.pdf>. 9

GREENSPAN, G. **MultiChain private blockchain — white paper**. 2015. Disponível em: <https://www.multichain.com/download/MultiChain-White-Paper.pdf>. 22

GREWAL, G. S.; RYAN, M. D.; CHEN, L.; CLARKSON, M. R. Du-vote: remote electronic voting with untrusted computers. In: COMPUTER SECURITY FOUNDATIONS SYMPOSIUM. **Proceedings...** IEEE, 2015. (CSF '15), p.

155–169. ISBN 978-1-4673-7538-2. Disponível em: <https://doi.org/10.1109/CSF.2015.18>. 12

HANIFATUNNISA, R.; RAHARDJO, B. Blockchain based e-voting recording system design. In: INTERNATIONAL CONFERENCE ON TELECOMMUNICATION SYSTEMS SERVICES AND APPLICATIONS (TSSA). **Proceedings...** [S.l.]: IEEE, 2017. p. 1–6. 36

HJALMARSSON, F.; HREIOARSSON, G. K.; HAMDAQA, M.; HJALMTYSSON, G. Blockchain-based e-voting system. In: INTERNATIONAL CONFERENCE ON CLOUD COMPUTING (CLOUD). **Proceedings...** [S.l.]: IEEE, 2018. p. 983–986. 23

HUANG, J.; LI, H.; ZHANG, J. Blockchain based log system. In: INTERNATIONAL CONFERENCE ON BIG DATA (BIG DATA). **Proceedings...** [S.l.]: IEEE, 2018. p. 3033–3038. 71

Jornal Gazeta do Povo. **Presidente: resultado segundo turno**. 2018. Disponível em: <https://especiais.gazetadopovo.com.br/eleicoes/2018/resultados/brasil-2turno-presidente/>. 58

KHOURY, D.; KFOURY, E. F.; KASSEM, A.; HARB, H. Decentralized voting platform based on ethereum blockchain. In: INTERNATIONAL MULTIDISCIPLINARY CONFERENCE ON ENGINEERING TECHNOLOGY (IMCET). **Proceedings...** [S.l.]: IEEE, 2018. p. 1–6. 24

KULYK, O.; NEUMANN, S.; VOLKAMER, M.; FEIER, C.; KOSTER, T. Electronic voting with fully distributed trust and maximized flexibility regarding ballot design. In: INTERNATIONAL CONFERENCE ON ELECTRONIC VOTING: VERIFYING THE VOTE (EVOTE). **Proceedings...** [S.l.], 2014. p. 1–10. 11

LAI, W.; HSIEH, Y.; HSUEH, C.; WU, J. DATE: A decentralized, anonymous, and transparent e-voting system. In: INTERNATIONAL CONFERENCE ON HOT INFORMATION-CENTRIC NETWORKING (HotICN). **Proceedings...** [S.l.]: IEEE, 2018. p. 24–29. 24

LEE, Y.; PARK, Y. B. A proposal of iterative consensus process for group decision making. In: INTERNATIONAL CONFERENCE ON PLATFORM TECHNOLOGY AND SERVICE (PlatCon). **Proceedings...** [S.l.], 2019. p. 1–4. 24

LEETARU, K. **How Estonia's e-voting system could be the future**. 2017. Disponível em: <https://www.forbes.com/sites/kalevleetaru/2017/06/07/how-estonias-e-voting-system-could-be-the-future>. 12

LUO, Y.; CHEN, Y.; CHEN, Q.; LIANG, Q. A new election algorithm for DPos consensus mechanism in blockchain. In: INTERNATIONAL CONFERENCE ON DIGITAL HOME (ICDH). **Proceedings...** [S.l.], 2018. p. 116–120. 24

MARTINS, R. **Voto impresso: desconfiança nas urnas eletrônicas vale 2 bilhões?** 2018. Disponível em: <https://exame.abril.com.br/brasil/a-desconfianca-das-urnas-eletronicas-vale-2-bilhoes-de-reais/>. 2, 58

MCNABB, M. **How blockchain and drones go together**. ago. 2018. Disponível em: <https://dronelife.com/2018/08/13/how-blockchain-and-drones-go-together/>. 71

MENG, Y.; CAO, Z.; QU, D. A committee-based byzantine consensus protocol for blockchain. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING AND SERVICE SCIENCE (ICSESS). **Proceedings...** [S.l.]: IEEE, 2018. p. 1–6. 38

MERCURI, R. T. **Electronic vote tabulation checks and balances**. Thesis (PhD) — University of Pennsylvania, 2001. Disponível em: <https://repository.upenn.edu/dissertations/AAI3003665>. 1

MUDLIAR, K.; PAREKH, H.; BHAVATHANKAR, P. A comprehensive integration of national identity with blockchain technology. In: INTERNATIONAL CONFERENCE ON COMMUNICATION INFORMATION AND COMPUTING TECHNOLOGY (ICCICT). **Proceedings...** [S.l.], 2018. p. 1–6. 24

NAKAMOTO, S. **Bitcoin: a peer-to-peer electronic cash system**. 2008. Disponível em: <https://bitcoin.org/bitcoin.pdf>. 12, 22, 23

NASH, G. **Let's build the tiniest blockchain**. 2017. Disponível em: <https://medium.com/crypto-currently/lets-build-the-tiniest-blockchain-e70965a248b>. 31

NAZIRIDIS, N. **Comparing ECDSA vs RSA**. 2018. Disponível em: <https://www.ssl.com/article/comparing-ecdsa-vs-rsa/>. 39

RAMALHO, R. **Prazo para eleitor pedir voto em trânsito termina nesta quinta**. 2018. Disponível em: <https://g1.globo.com/politica/eleicoes/2018/noticia/2018/08/22/termina-nesta-quinta-feira-prazo-para-eleitor-pedir-voto-em-transito.ghtml>. 58

RIVEST, R. L.; WACK, J. P. **On the notion of "software independence" in voting systems**. 2006. Disponível em: <https://people.csail.mit.edu/rivest/pubs/RW06.pdf>. 65

SARAF, C.; SABADRA, S. Blockchain platforms: a compendium. In: INTERNATIONAL CONFERENCE ON INNOVATIVE RESEARCH AND DEVELOPMENT (ICIRD). **Proceedings...** [S.l.]: IEEE, 2018. p. 1–6. 26

SHAHEEN, S. H.; YOUSAF, M.; JALIL, M. Temper proof data distribution for universal verifiability and accuracy in electoral process using blockchain. In: INTERNATIONAL CONFERENCE ON EMERGING TECHNOLOGIES (ICET). **Proceedings...** [S.l.], 2017. (13), p. 1–6. 27

SHAHZAD, B.; CROWCROFT, J. Trustworthy electronic voting using adjusted blockchain technology. **IEEE Access**, v. 7, p. 24477–24488, 2019. ISSN 2169-3536. Disponível em: <https://ieeexplore.ieee.org/document/8651451/>. 23

SHELKOVNIKOV, A. **Blockchain. enigma. paradox. opportunity**. 2016. Disponível em: <https://www2.deloitte.com/content/dam/Deloitte/uk/Documents/Innovation/deloitte-uk-blockchain-full-report.pdf>. 14

SHIRRIFF, K. **Mining Bitcoin with pencil and paper: 0.67 hashes per day**. 2014. Disponível em: <http://www.righto.com/2014/09/mining-bitcoin-with-pencil-and-paper.html>. 65

SINGH, A.; CHATTERJEE, K. SecEVS : Secure electronic voting system using blockchain technology. In: INTERNATIONAL CONFERENCE ON COMPUTING, POWER AND COMMUNICATION TECHNOLOGIES (GUCON). **Proceedings...** [S.l.], 2018. p. 863–867. 23

SOLVAK, K. V. M. **E-voting in Estonia: technological diffusion and other developments over ten years (2005 - 2015)**. University of Tartu, 2016. Disponível em: <http://skytte.ut.ee/sites/default/files/skytte/e_voting_in_estonia_vassil_solvak_a5_web.pdf>. 12

SULTAN, K.; RUHI, U.; LAKHANI, R. Conceptualizing blockchains: characteristics & applications. 2018. Disponível em: <https://arxiv.org/abs/1806.03693v1>. 15

VRIES, A. d. **Bitcoin energy consumption index**. 2019. Disponível em: <https://digiconomist.net/bitcoin-energy-consumption>. 21

WANG, Y.; CAI, S.; LIN, C.; CHEN, Z.; WANG, T.; GAO, Z.; ZHOU, C. Study of blockchains's consensus mechanism based on credit. **IEEE Access**, v. 7, p. 10224–10231, 2019. ISSN 2169-3536. Disponível em: <https://ieeexplore.ieee.org/document/8605507/>. 38

WRAY, S. **Blockchain e-voting: Backed by US candidate, hacked in Moscow - Smart Cities World**. 2019. Disponível em: <https://www.smartcitiesworld.net/special-reports/special-reports/blockchain-e-voting-backed-by-us-candidate-hacked-in-moscow>. 12

WU, H.; YANG, C. A blockchain-based network security mechanism for voting systems. In: INTERNATIONAL COGNITIVE CITIES CONFERENCE (IC3). **Proceedings...** [S.l.], 2018. p. 227–230. 23

YAVUZ, E.; KOÇ, A. K.; ÇABUK, U. C.; DALKILIÇ, G. Towards secure e-voting using ethereum blockchain. In: INTERNATIONAL SYMPOSIUM ON DIGITAL FORENSIC AND SECURITY (ISDFS). **Proceedings...** [S.l.], 2018. p. 1–7. 24

YU, J.; KOZHAYA, D.; DECOUCHANT, J.; ESTEVES-VERISSIMO, P. RepuCoin: Your reputation is your power. **IEEE Transactions on Computers**, v. 68, n. 8, p. 1225–1237, ago. 2019. ISSN 0018-9340, 1557-9956, 2326-3814. Disponível em: <https://ieeexplore.ieee.org/document/8645706/>. 38

ZHANG, W.; YUAN, Y.; HU, Y.; HUANG, S.; CAO, S.; CHOPRA, A.; HUANG, S. A privacy-preserving voting protocol on blockchain. In: INTERNATIONAL CONFERENCE ON CLOUD COMPUTING (CLOUD). **Proceedings...** [S.l.]: IEEE, 2018. p. 401–408. 24

PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE

Teses e Dissertações (TDI)

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

Manuais Técnicos (MAN)

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

Notas Técnico-Científicas (NTC)

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programas de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

Relatórios de Pesquisa (RPQ)

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

Propostas e Relatórios de Projetos (PRP)

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

Publicações Didáticas (PUD)

Incluem apostilas, notas de aula e manuais didáticos.

Publicações Seriadas

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Constam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

Programas de Computador (PDC)

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. Aceitam-se tanto programas fonte quanto os executáveis.

Pré-publicações (PRE)

Todos os artigos publicados em periódicos, anais e como capítulos de livros.