BLOCKCHAIN: uma solução para transparência e integridade

BLOCKCHAIN: a solution for transparency and integrity

Caio Augusto Barreto do Amaral
Graduando em Tecnologia em Banco de Dados
caio.amaral2@fatec.sp.gov.br

Gustavo Bruschi
Professor do curso de Tecnologia em Banco de Dados
gustavo@bruschi.net

RESUMO:

Existem muitas situações hoje em que a integridade e transparência de dados é fundamental. Uma vez que a arquitetura de banco de dados comumente utilizada concebe problemas nestes aspectos, esse trabalho propõe utilizar a tecnologia Blockchain para resolver os problemas de transparência e integridade por meio da replicação e validação dos dados que ocorre durante o processo de inserção. Quando um dado é inserido, ele é validado no processo chamado proof-of-work, que garante que somente os dados presentes na maioria dos nós permaneçam. Utilizando um banco de dados chamado BigchainDB, que replica o comportamento de uma blockchain, foi feita uma simulação de eleição e votos. Um dos nós teve os dados manipulados e, conforme esperado, a alteração não foi replicada para os demais nós. Conclui-se que a estrutura Blockchain cumpre seu papel em integridade neste tipo de cenário.

Palavras-chave: Blockchain. Transparência. Integridade.

ABSTRACT:

There are many situations today where data integrity and transparency are fundamental. Since the commonly used database architecture conceives problems in these respects, this work proposes to use blockchain to solve the problems of transparency and integrity through the data replication and validation that occurs during the insertion process. When a data is entered, it is validated in the process called proof-of-work, which ensures that only the data present on most nodes remain. Using database called BigchainDB, which replicates the behavior of a blockchain, a simulation of a election and voting was performed. One of the node had the data manipulated and, as expected, the chand wasn't replicated to the other nodes. It is concluded that the blockchain structure fulfills its role in integrity in this type of cenario.

Keywords: *Blockchain. Transparency. Integrity.*

1 INTRODUÇÃO

Existem hoje diversos problemas na tecnologia, dentre eles a dificuldade de distribuir informações e garantir sua autenticidade. O conceito de armazenamento

de dados mais usado hoje implica que os dados sejam guardados em um ou mais pontos, na maioria das vezes gerenciados por um grupo pequeno de pessoas.

Guardar todos os dados em um ou mais lugares, gerenciados somente por algumas pessoas, torna mais fácil o seu gerenciamento e em muitos casos também mais rápido o seu acesso. Também permite maior segurança dos dados e evita desperdício de recursos combatendo a redundância. Esse modelo é justificável e muito bem aplicável na maioria dos casos. No entanto existem situações em que os dados precisam ser conhecidos e acessíveis por todos os usuários para permitir transparência e autenticidade nas operações.

Informações centralizadas impedem, por exemplo, que um sistema de eleições pela internet seja confiável, uma vez que todos os dados estariam nas mãos de uma única pessoa ou organização. Por estarem sob domínio de um grupo restrito de indivíduos, essas informações poderiam ser manipuladas e disponibilizadas segundo o interesse deste grupo.

Para resolver os problemas da centralização foi criado uma arquitetura para compartilhamento de dados chamada blockchain. Nessa rede os dados são replicados em diversos pontos e a autenticidade de uma transação é garantida pela presença de dados iguais em todos os pontos. Em uma blockchain pública, os usuários podem se apresentar para serem pontos da rede e hospedar todos os dados, descentralizando assim a informação.

Esse trabalho propõe avaliar a transparência e integridade de uma rede blockchain para compartilhamento de dados em uma situação de eleições, como exemplificada anteriormente. Tem por objetivo mostrar se é possível garantir a veracidade dos dados em situações em que sua manipulação é possivelmente reprovável.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Transparência e Distribuição da Informação

Transparência é a qualidade daquilo que permite ver através de si. Segundo a explicação de Gomes, Amorim e Almada (2018), é um princípio que se relaciona com a moral e auxilia a democracia. Segundo eles, a transparência e a publicidade promovem um comportamento mais virtuoso daqueles que são responsáveis por aquilo que será exposto, uma vez que isso poderá ser conhecido por outras pessoas.

Além do conceito apresentado, a transparência também pode demonstrar confiança naquele que recebe a informação e no que está sendo executado, o que pode ser muito benéfico em determinadas circunstâncias. Uma empresa que permita o cliente ver o que está sendo feito pode trazer maior segurança a aquele que faz uso de seus serviços ou produtos, por exemplo.

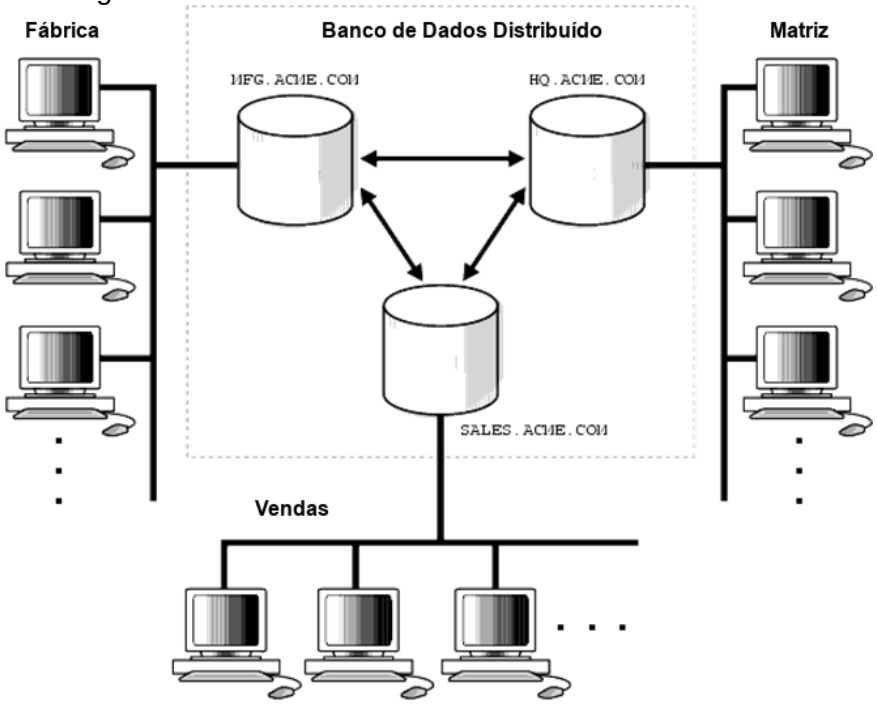
2.2 Banco de Dados Distribuído

Segundo Elmasri e Navathe (2011), temos um banco de dados distribuído quando vários bancos de dados se relacionam de forma lógica, distribuídos através de uma rede de computadores.

Na figura 1 é possível observar três bases de dados que são representadas separadamente, estão ligadas a diversos clientes e se comunicam entre si. A separação entre as bases de dados deve ser transparente para o usuário.

Através da evolução dos sistemas em redes também se tornou possível distribuir o armazenamento e processamento das informações. A distribuição dessas tarefas pode trazer grandes benefícios no desempenho do sistema e segurança contra possíveis falhas no processo. Özsu e Valduriez (2011) mencionam que um banco de dados distribuído pode exibir transparência entre os equipamentos, confiabilidade das transações, performance melhorada e facilidade de expansão.

Figura 1 – Modelo de banco de dados distribuído



Fonte: Oracle (2019, adaptado)

Elmasri e Navathe (2011) também explicam como vantagens a facilidade e flexibilidade de desenvolvimento de aplicações, maior confiabilidade e disponibilidade, maior desempenho e expansão mais fácil. A facilidade e flexibilidade ocorre devido à transparência dos dados. Confiabilidade e desempenho são um reflexo do isolamento dos pontos da rede. Caso um ponto apresente falha, os demais continuam funcionando e o impacto ao usuário é reduzido. O maior desempenho ocorre pela possibilidade de alocar os dados de forma geograficamente mais próxima aos usuários e em porções menores. Por fim, a facilidade de expansão existe devido ao banco de dados distribuído ser naturalmente um conjunto de pontos separados, o que facilita a inserção de novos nós.

2.3 Blockchain

A blockchain ainda possui muitas definições dadas por diferentes autores. Pires (2016, p. 14) expõe blockchain como “um livro razão validado por diversos nós de uma rede P2P”. Arruda (2017, p. 141) define blockchain como “um tipo de banco de dados distribuído que armazena o registro de transações de forma permanente [...], inviolável, "inderrubável" e extremamente eficiente”. Já Kiyomoto, Rahman e Basu (2017) dizem que a blockchain é um sistema distribuído de gerenciamento de registros de transação.

Nakamoto (2008) explica que em uma rede blockchain os dados são armazenados em blocos, que são ligados entre si através de hashes. O hash é criado usando os itens do bloco, o timestamp do momento e o timestamp anterior. Como cada bloco possui uma referência do bloco anterior, forma-se assim uma “corrente”. O mesmo autor retrata que após o processo de geração de hash a máquina precisa executar um algoritmo de verificação. Na blockchain do bitcoin o algoritmo usado é chamado de proof-of-work. Esse algoritmo somente aceita como válido o hash que iniciar com uma quantidade específica de zeros. Essa quantidade é chamada de nounce. O usuário precisa gerar hashes de diversas combinações até conseguir o hash com o número de zeros necessários.

O proof-of-work também assegura que os dados do bloco não serão mudados, uma vez que, ao alterar os dados em qualquer bloco, seu hash também é alterado juntamente com todos os hashes dos blocos subsequentes.

Após gerar um bloco, o nó continuará o processo de geração do próximo bloco. Caso blocos diferentes tenham sido gerados ao resolver o proof-of-work, os nós aceitarão como verdadeiro o que pertencer a uma corrente maior de blocos.

Glaser (2017) explica que existem 3 tipos de blockchain: públicas, privadas e híbridas. As públicas possuem código aberto e permitem a participação de quaisquer usuários, como por exemplo a rede do Bitcoin e do Ethereum. Blockchains privadas são acessíveis somente por um número restrito de usuários e o código está nas mãos de uma empresa ou organização. As blockchains híbridas misturam ambas as características.

Nakamoto (2008) também expõe que manter e disponibilizar uma blockchain exige um esforço de CPU e eletricidade. Para que seja viável o apoio dos nós na rede, existe um sistema de recompensas. Na rede do bitcoin, para cada primeira transação de bloco que registre na rede, o nó recebe um montante que representa um valor ou moeda dentro dessa rede. Também é possível recompensar o nó com uma taxa pela transação executada. O processo de criar estes blocos é comparado ao processo de mineração de ouro.

2.3.1 Blockchain e Banco de Dados

Blockchain, conforme detalhado no ponto 2.3 e dito por Yli-Huumo et al (2016, p. 2), é uma “solução de banco de dados distribuído que mantém uma lista de crescimento constante de registros que são confirmados pelos nós participantes nela”.

Vale ressaltar que, devido às regras seguidas em uma blockchain, conforme indicado no trabalho de McConaghy et al (2016), a blockchain do bitcoin possui uma série de pontos negativos em relação aos bancos de dados tradicionais:

a) Execução de poucas transações por segundo;
b) Latência de 10 minutos após uma transação de escrita confirmada;
c) Capacidade de poucos GB de armazenamento.

E ainda, em sua proposta de criação de um banco de dados específico para blockchain (chamado de BigchainDB), McConaghy et al apontam os seguintes elementos presentes em bancos de dados distribuídos tradicionais, porém ausentes na rede do Bitcoin:

a) Execução de mais de 1 milhão de transações por segundo;
b) Latência de fração de segundos;
c) Capacidades de armazenamento de petabytes ou mais.

2.3.2 Arquitetura da blockchain

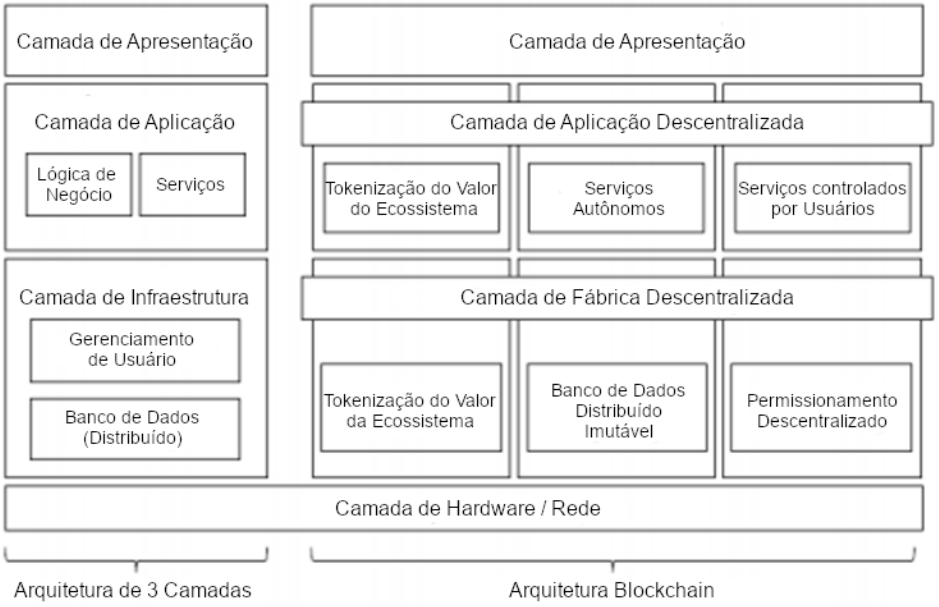
De acordo com Glaser (2017), quanto à estrutura, podemos dividir uma blockchain em 2 camadas principais: Camada de fábrica e camada de aplicação. As camadas podem ser visualizadas na figura 2, que compara a arquitetura de uma blockchain com a de uma base de dados distribuída de três camadas.

A camada de fábrica é o alicerce da blockchain, onde são gerenciadas permissões de usuário, a comunicação dos nós, a infraestrutura de chave privada e toda a infraestrutura para gestão do banco de dados e contratos inteligentes.

A situação em que a blockchain é utilizada determina quem serão os detentores dessa camada, uma vez que o acesso a essa estrutura pode gerar riscos de segurança a depender da informação contida.

A segunda camada, chamada de camada de aplicação, engloba o ecossistema criado pelos diversos desenvolvedores do sistema. Na rede Ethereum é onde estão armazenados os contratos inteligentes. Pode-se dizer que essa camada está sob o controle dos usuários que desenvolveram esses códigos.

Figura 2 – Camadas de um blockchain



Fonte: GLASER (2017, adaptado)

Essa descentralização de controle proporcionada pelas duas camadas cria um balanço no ecossistema, possibilitando o que é chamado de sistema “trustless” ou sistema “sem confiança”.

2.3.3 Case

De acordo com Kelly (2018), no estado da Virgínia Ocidental foi utilizado um aplicativo baseado em blockchain para se realizar duas eleições em 2018. O aplicativo foi desenvolvido por uma empresa chamada Voatz em versões para Android e iOS.

Segundo a empresa Voatz, após o registro, o aplicativo permite ao usuário se autenticar através do celular em 3 etapas: Primeiro o usuário escaneia a licença de motorista ou o passaporte, depois filma o próprio rosto em um vídeo curto e por fim

escaneia a impressão digital no celular. Após isso o dispositivo é vinculado ao usuário, que pode então fazer seu voto. O aplicativo faz a comparação do rosto do vídeo com o do documento para realizar a validação do usuário.

Miller (2018) informa que o método através do aplicativo da Voatz estaria disponível para militares em atividade no exterior.

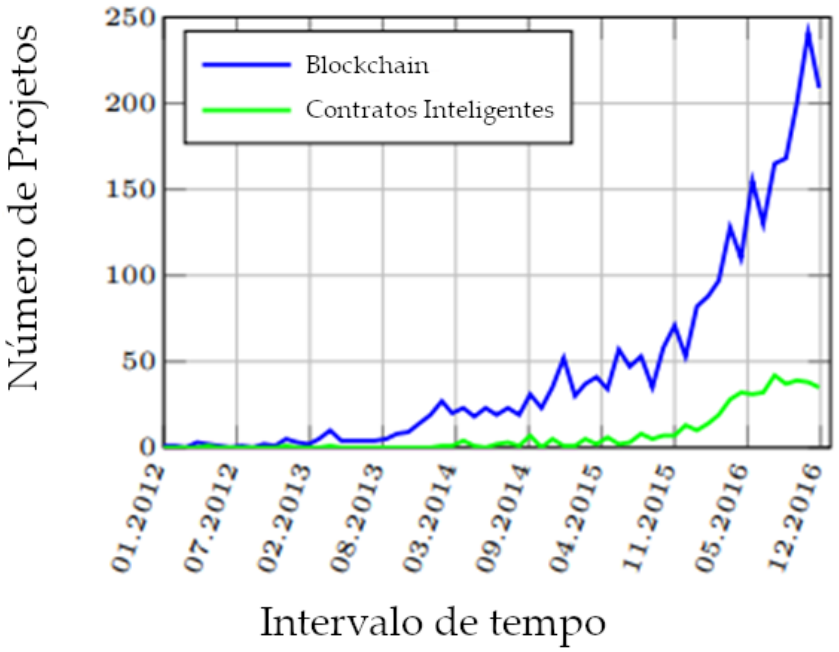
2.4 Contratos Inteligentes

Gatteschi et al (2018) explicam que contratos inteligentes são pedaços que códigos que se comportam de determinadas formas, de acordo com as condições em que são executados. São programações hospedadas dentro de uma blockchain, que fazem uso dos dados deste e seguem regras específicas. Por ser implantado dentro de uma blockchain, o código de um contrato inteligente não pode ser alterado.

Em uma blockchain pública qualquer pessoa com algum conhecimento de programação pode inspecionar o contrato, que, já que não pode ser alterado, proporciona segurança de que as regras foram seguidas da forma em que estão programadas.

Bartoletti e Pompianu (2017) mostraram que o número de projetos no Github envolvendo blockchain e contratos inteligentes cresceu muito entre 2012 e o final de 2016 (ver figura 3).

Figura 3 – Número de projetos relacionados a blockchain no Github



Fonte: BARTOLETTI e POMPIANU (2017, adaptado)

Por meio dos contratos inteligentes é possível realizar uma série de tarefas dentro de uma blockchain. Bartoletti e Pompianu (2017) classificaram os códigos existentes na rede Ethereum em 5 categorias principais:

a) Financial: Códigos que lidam com moedas como o principal foco. Fazem transações bancárias, crowdfunding, trocas, investimentos etc.;
b) Notary: Códigos para armazenar dados, tirando proveito da característica de imutabilidade da blockchain;

c) Games: Códigos que servem como jogos de azar e habilidade;
d) Wallet: Códigos que tem o intuito de facilitar o uso da blockchain, gerenciando chaves, carteira, transações etc.;
e) Library: Códigos de uso geral, como conversões, que são utilizados por outros contratos.

2.5 BigchainDB

Conforme explicado por BigchainDB GmbH (2018) em seu Whitepaper, o BigchainDB é um software que possui propriedades de blockchain e de banco de dados distribuídos. Ele une características de ambos em uma só interface.

Esse software consiste em uma interface de comunicação entre diversos nós em uma rede, implementando protocolos Tendermint e armazenando as informações em bases de dados MongoDB independentes.

Segundo os criadores, o uso do Tendermint permite uma comunicação satisfatoriamente rápida entre dezenas de nós localizados em diferentes continentes, proporcionando uma alta taxa de transações. A interface do BigchainDB por sua vez fornece segurança, uma vez que todas as transações são assinadas com criptografia e não há recursos implementados para alteração ou exclusão de dados.

3 MATERIAIS E MÉTODOS

Os testes foram realizados em uma máquina física executando 3 containers Docker. Os containers são conjuntos de aplicações e configurações pré-definidas, necessários para a execução de um serviço, no caso o BigchainDB.

A máquina física usada foi um notebook Acer Aspire E14, com um Core i5-7200U 2.5GHz, 12GB de RAM DDR4, SSD Kingston 128GB e o S.O. Linux Fedora Workstation 34. A imagem base usada para os containers foi fornecida pela BigchainDB em seu repositório oficial no Github, site onde é possível hospedar e compartilhar o código-fonte de aplicações. A imagem usada foi a All-in-one.

Todos os containers possuíam sua própria base de dados MongoDB e estavam conectados na mesma rede. A comunicação entre eles foi feita usando o Tendermint, ferramenta para consenso de rede através de protocolos pré-definidos. Os dados foram inseridos através de uma aplicação web desenvolvida em JavaScript.

As aplicações escritas em JavaScript consistiam em um formulário web que enviava os dados para um servidor Express, que por sua vez encaminhava as transações para um dos nós de BigchainDB disponíveis.

Na página web era possível utilizar dois formulários: um para candidatura e outro para voto. No primeiro era possível cadastrar o nome e número, e no segundo o nome, número de eleitor e número do candidato. Além dos formulários, também foi criado um endpoint onde era possível consultar a quantidade de votos de cada candidato.

4 RESULTADOS E DISCUSSÃO

A primeira etapa do experimento consistiu em construir a rede de nós BigchainDB para armazenamento das informações. Em primeiro momento, foram criados containers Ubuntu, onde seriam instalados todos os componentes da rede passo a passo, porém havia problemas de comunicação entre os containers. Para

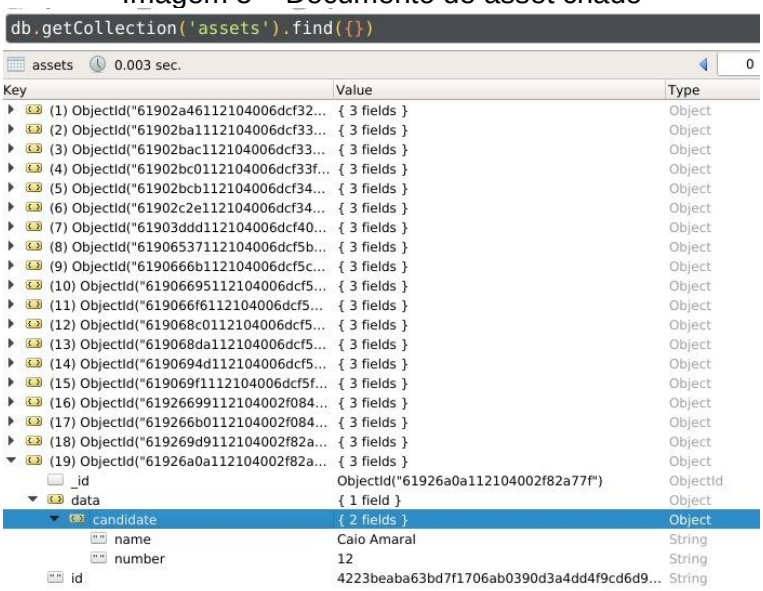
evitar maiores atrasos, optou-se então pelo uso das imagens prontas, que são fornecidas no repositório do BigchainDB. A criação dos containers foi feita utilizando a ferramenta Docker Compose. Para o experimento foram criados 3 containers iguais.

Após a criação da rede, foi feita a interface web para votação, apresentada na imagem 4. Para simplificação do experimento foi feita somente uma tela que exibia a lista de candidatos, um formulário para candidatura e um formulário para votação. Ao se colocar um novo candidato, a aplicação gerava um conjunto único de chaves Ed25519 para este usuário, representando suas chaves de acesso à rede. Ed25519 é um sistema de assinatura digital, que utiliza um algoritmo de criptografia chamado SHA-512 e a curva 25519 na criação das chaves. O formulário de votação, por sua vez, também gerava um conjunto de chaves Ed25519 e usava essas chaves para realizar uma transação CREATE no BigchainDB, conforme mostra a imagem 5. O asset continha as informações do candidato votado, enquanto as informações do eleitor foram colocadas nos metadados (ver a imagem 6). Essa transação criava um asset na base de dados, que era então replicado para toda a rede.

Imagem 4 – Tela de votação

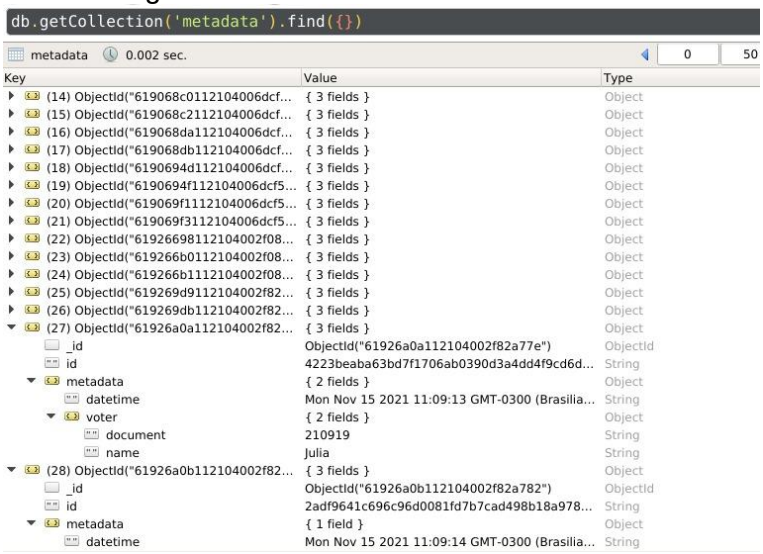
Fonte: O autor (2021)

Imagem 5 – Documento de asset criado



Fonte: O autor (2021)

Imagem 6 – Documento de metadata criado



Fonte: O autor (2021)

Na imagem 7 é possível observar um endpoint de consulta dos resultados nesta aplicação web. A aplicação fazia uma consulta a um endpoint fornecido pelo BigchainDB, buscando por todos os assets que continham os números de eleitores e então agrupando estes registros. Com isso foi possível verificar os resultados da votação.

Imagem 7 – Resultados obtidos na votação


```
// 20211115111001
// http://localhost:3000/result

[
  {
    "name": "Rafael Silas",
    "number": "7",
    "votes": 7
  },
  {
    "name": "Caio Amaral",
    "number": "12",
    "votes": 12
  }
]
```

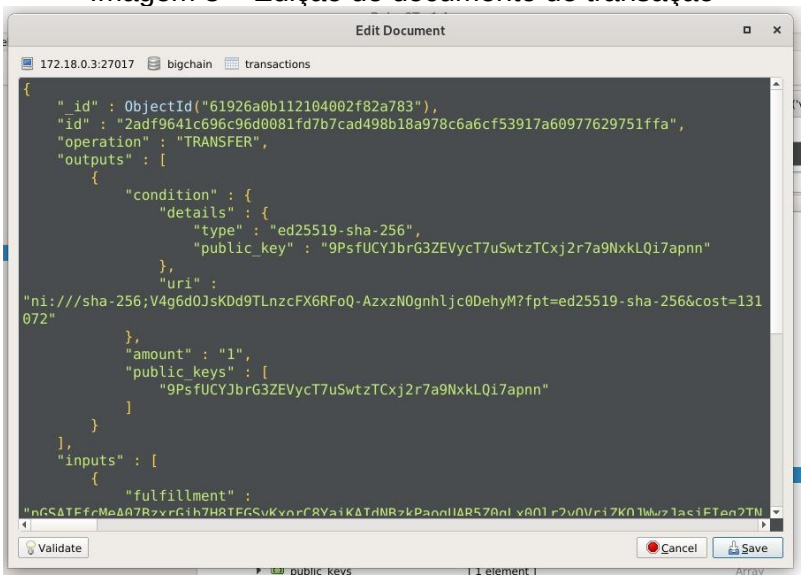

Fonte: O autor (2021)

Tendo feito isso, usando a ferramenta Robo3T, a base foi acessada e alguns assets modificados, para que fossem computados para um outro candidato. Com isso percebeu-se uma alteração nos resultados, ao consultar-se o mesmo nó. Essa alteração, contudo, não foi replicada para os demais nós, que ficaram idênticos. Apesar disso novos votos puderam ser adicionados normalmente.

Após este primeiro teste, foi aplicada uma segunda ideia. Desta vez, além das operações CREATE, os eleitores também realizavam uma transação TRANSFER, passando a posse do asset, que representava o seu voto, para o candidato escolhido. Dessa forma haveria uma segunda maneira de verificar a integridade da rede, que seria consultando a validade das transações.

Novamente os dados foram alterados, desta vez passando a posse de um asset para outra chave. A imagem 8 apresenta este procedimento. Apesar disso ainda assim foi possível fazer uma nova operação após a alteração, conforme apresentado na imagem 9. As informações alteradas não foram replicadas para outros nós.

Imagem 8 – Edição de documento de transação



Fonte: O autor (2021)

Imagem 9 – Novo voto computado mesmo após alteração


```
// 20211115111718
// http://localhost:3000/result

[
  {
    "name": "Rafael Silas",
    "number": "7",
    "votes": 7
  },
  {
    "name": "Caio Amaral",
    "number": "12",
    "votes": 13
  }
]
```


Fonte: O autor (2021)

Da mesma forma que na criação do asset, o BigchainDB cria um Hash ID para cada transação usando os dados desta como base. Esse ID é criado usando o algoritmo SHA3 (256). Este algoritmo não permite reversão do hash, ou seja, o hash pode ser criado, mas não desconstruído. A única maneira de se obter o mesmo hash é utilizando os mesmos dados anteriores.

Para validar as transações gravadas seria necessário recriar cada ID utilizando os dados que compõem o registro, porém o BigchainDB não fornecia nenhum recurso para fazer essa operação. Devido à complexidade esse procedimento não foi implementado.

5 CONCLUSÃO

Houve limitações no experimento devido a ferramenta utilizada, o BigchainDB, que ainda deixa de apresentar alguns recursos importantes neste cenário. Alguns destes recursos incluem a funcionalidade de validar a integridade das transações, recriando o hash de todos os dados e comparando os seus resultados, e também o recurso de comparar os dados aos serem recebidos em outros nós, evitando que um nó corrompido insira novos dados na rede. Dentro do escopo também não foi possível aplicar os contratos inteligentes, como os apresentados dentro da rede Ethereum.

Contudo pode-se observar que a estrutura de blockchain proporcionou maior confiabilidade das informações, trazendo camadas adicionais de segurança e colocando processos de difícil replicação. Para uma manipulação efetiva da base de dados em uma rede completamente configurada, seria necessário que o atacante adulterasse de forma quase simultânea uma quantidade significativa de nós. Essa alteração precisaria conter as novas informações, incluindo hashes concisos que pudessem ser validados posteriormente. Em uma rede com muitos nós de diferentes hosts isso se torna um esforço extremamente grande.

Conclui-se assim que, utilizando uma ferramenta mais madura e aplicando todos os protocolos e validações sugeridas em uma rede blockchain, pode-se obter um resultado satisfatório de integridade de dados.

Como trabalhos futuros sugere-se a reaplicação do cenário, porém com a elaboração de um recurso de validação de hashes. Esse recurso poderia ser utilizado na adição de novos registros, validando o bloco anterior, ou poderia ser usado na criação de um recurso que validasse todas as transações registradas na BigchainDB, retornando as inconsistências e apontando qual nó teve maior sucesso nos testes. Outro experimento que poderia ser elaborado é a execução do mesmo cenário em um cluster MongoDB. Após a criação e uso do cluster, comparar as dificuldades encontradas ao manipular os dados na rede BigchainDB e no cluster MongoDB.

6 REFERÊNCIAS

ARRUDA, G. O. A Tecnologia a Serviço da Democracia: O Processo Eleitoral na Era da Informação. *Revista da Advocacia Pública Federal*, Brasília, v. 1, n. 1, p. 139-148, 2017. Disponível em: <http://anafenacional.org.br/seer/revista/article/view/9/9>. Acesso em: 01 nov. 2018.

BARTOLETTI, M.; POMPIANU, L. *An empirical analysis of smart contracts: platforms, applications, and design patterns*. 2017. Disponível em: <https://arxiv.org/pdf/1703.06322.pdf>. Acesso em: 31 maio 2019.

BIGCHAINDB GMDH. *BigchainDB 2.0: The Blockchain Database*. Maio 2018. Disponível em: < https://www.bigchaindb.com/whitepaper/bigchaindb-whitepaper.pdf>. Acesso em: 22 nov. 2021.

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados*. 6. ed. São Paulo: Pearson Education, 2011.

GATTESCHI, V. et al. *Blockchain and Smart Contracts for Insurance:* Is The Technology Mature Enough? 20 fev. 2018. Disponível em: <https://www.mdpi.com/1999-5903/10/2/20/pdf>. Acesso em: 30 maio 2019.

GLASER, F. *Pervasive Decentralization of Digital Infrastructures:* A Framework for Blockchain enabled System and Use Case Analysis. 04 jan. 2017. Disponível em: <http://hdl.handle.net/10125/41339>. Acesso em: 06 maio 2019.

GOMES, W.; AMORIM, P. K. D.; ALMADA, M. P. *Novos desafios para a ideia de transparência pública*. 04 abr. 2018. Disponível em: <http://www.e-compos.org.br/e-compos/article/view/1446/1847>. Acesso em: 15 nov. 2018.

KELLY, M. *Nearly 150 West Viriginians voted with a mobile blockchain app*. 10 nov. 2018. Disponível em: <https://www.theverge.com/2018/11/10/18080518/blockchain-voting-mobile-app-west-virginia-voatz>. Acesso em: 31 maio 2018.

KIYOMOTO, S.; RAHMAN, M. S.; BASU, A. *On Blockchain-Based Anonymized Dataset Distribuition Platform*. Londres, 2017. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7965711>. Acesso em: 01 nov. 2018.

MCCONAGHY, T.; MARQUES, R; MÜLLER, A.; et al; *BigchainDB:* A Scalable Blockchain Database. Berlim, 8 jun. 2016. Disponível em: <https://mycourses.aalto.fi/pluginfile.php/378362/mod_resource/content/1/bigchaindb-whitepaper.pdf>. Acesso em: 31 maio 2019.

MILLER, B. *West Virginia Becomes First State to Test Mobile Voting by Blockchain in a Federal Election*. 28 mar. 2018. Disponível em: <https://www.govtech.com/biz/West-Virginia-Becomes-First-State-to-Test-Mobile-Voting-by-Blockchain-in-a-Federal-Election.html>. Acesso em: 31 maio 2019.

NAKAMOTO, S. *Bitcoin:* A Peer-to-Peer Electronic Cash System. 2008. Disponível em: <https://bitcoin.org/bitcoin.pdf>. Acesso em: 14 mar. 2019.

PIRES, T. P. *Tecnologia Blockchain e Suas Aplicações Para Provimento de Transparência em Transações Eletrônicas*. Brasília, 2016. Disponível em: <http://bdm.unb.br/bitstream/10483/16252/1/2016_TimoteoPimentaPires_tcc.pdf>. Acesso em: 18 set. 2018.

ÖZSU, M. T.; VALDURIEZ, P. *Principles of Distributed Database Systems*. 3. ed. New York: Springer, 2011. Disponível em: <https://books.google.com.br/books?id=TOBaLQMuNV4C&printsec=frontcover#v=onepage&q&f=false>. Acesso em: 19 nov. 2018.

TARASOV, P.; TEWARI, H. *The Future of E-Voting*. IADIS Internacional Journal on Computer Science and Information Systems. Disponível em: <http://www.iadisportal.org/ijcsis/papers/2017210210.pdf>. Acesso em: 17 maio 2019.

VOATZ. *Frequently Asked Questions (FAQ)*. Disponível em: <https://voatz.com/faq.html>. Acesso em: 31 maio 2019.

YLI-HUUMO, J. et al. *Where Is Current Research on Blockchain Technology?* - A Systematic Review. 3 out. 2016. Disponível em: <https://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0163477&type=printable>. Acesso em: 31 maio 2019.