UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
ENGENHARIA DE COMPUTAÇÃO

LUCAS ANDRADE DE OLIVEIRA REIS

SISTEMA DE CRIPTOMOEDA CENTRALIZADO

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2020

LUCAS ANDRADE DE OLIVEIRA REIS

SISTEMA DE CRIPTOMOEDA CENTRALIZADO

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná, câmpus Cornélio Procópio, como requisito parcial para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Lucas D. H. Sampaio

CORNÉLIO PROCÓPIO

2020

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Cornélio Procópio
Diretoria de Graduação
Coordenação de Engenharia de Computação
Bacharelado em Engenharia de Computação

TERMO DE APROVAÇÃO

Sistema de Criptomoeda Centralizado

por

Lucas Andrade de Oliveira Reis

Este Trabalho de Conclusão de Curso de graduação foi julgado adequado para obtenção do Título de Bacharel em Engenharia de Computação e aprovado em sua forma final pelo Programa de Graduação em Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Cornélio Procópio, 30 de Julho de 2020

_____

Prof. Dr. Lucas Dias Hiera Sampaio

_____

Prof. Francisco Pereira Junior

_____

Prof. Dr. Rogério Santos Pozza

"A Folha de Aprovação assinada encontra-se na Coordenação do Curso"

AGRADECIMENTOS

Agradeço ao Prof. Dr. Lucas Dias Hiera Sampaio pela oportunidade, apoio e prestatividade na elaboração deste trabalho. Agradeço também a meus pais, Luiz e Rosemary e meus irmãos, Leonardo e Larissa que sempre me apoiaram em minhas conquistas.

RESUMO

REIS, Lucas Andrade de Oliveira. SISTEMA DE CRIPTOMOEDA CENTRALIZADO. 40 f. Trabalho de Conclusão de Curso – Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2020.

As criptomoedas trazem, pela primeira vez na história da humanidade, uma moeda e um sistema de transações em uma só plataforma, conquistando assim a atenção de investidores e pesquisadores, e resultando em milhares de estudos e novas aplicações desta tecnologia. Este trabalho traz informações sobre a implementação de um modelo de Blockchain com a criação de uma nova criptomoeda centralizada apresentando detalhes e implicações de sua implementação.

Palavras-chave: Blockchain, Criptomoeda, Banco, Central, Digital, Bitcoin, Centralizada

ABSTRACT

REIS, Lucas Andrade de Oliveira. CENTRALIZED CRYPTOCURRENCY SYSTEM. 40 f. Trabalho de Conclusão de Curso – Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2020.

The cryptocurrencies bring, for the first time in human history, a currency and a transaction system on the same single platform, thus achieving the attention of investors and researchers, and also resulting in thousands of studies and new applications of this technology. This work provides information on the implementation of a Blockchain model with the creation of a new centralized cryptocurrency presenting details and implications of its implementation.

Keywords: Blockchain, Cryptocurrency, Central, Bank, Digital, Bitcoin, Centralized

SUMÁRIO

1 INTRODUÇÃO

As criptomoedas tem atraído centenas de bilhões de dólares para sua economia (COINMARKETCAP, 2020) e experienciado um rápido aumento em sua popularidade (JUDMAYER et al., 2017), induzindo um crescente interesse científico em sua tecnologia.

Neste capítulo serão apresentadas motivações para o desenvolvimento deste trabalho, além de explicitar o objetivo e indicar os trabalhos relacionados.

1.1 MOTIVAÇÃO

É inegável que a mais popular de todas as criptomoedas existentes hoje, tanto em termos econômicos como de fama, é a Bitcoin, a qual pela primeira vez na história trouxe uma moeda virtual e um mecanismo de transferências em um só sistema. Sua capitalização de mercado nos últimos cinco anos passou de U$3,4 bilhões para U$181,7 bilhões (aumento de 5344%) e o valor por unidade da moeda aumentou de U$236,15 para U$8670,16 (aumento de 3671%) segundo a CoinMarketCap (2020), evidenciando o otimismo e a confiança de seus investidores.

Como destacadas por Mirzayi e Mehrzad (2017), apesar de desvantagens como custo operacional elevado, alta volatilidade e uso em cenários criminosos, diversas vantagens podem ser notadas com a utilização de criptomoedas, tais como: segurança nas transações, privacidade financeira, resistência a fraudes, taxa previsível de geração de novas moedas e ausência de gastos como impressão e distribuição de papel-moeda.

Tais pontos fortes são possíveis graças ao caráter seguro e distribuído dos sistemas individuais de cada criptomoeda, que possuem todos seus registros criptografados e guardados em um livro imutável, a Blockchain (MIRZAYI; MEHRZAD, 2017). Esses registros são visíveis para todos os participantes ativos no sistema da criptomoeda, que analisam e verificam se as informações nele armazenadas são verdadeiras. Deste modo, a descentralização acaba por garantir a autenticidade dos dados presentes na Blockchain.

Apesar da grande maioria das criptomoedas ser validada de forma descentralizada (com nós da rede espalhados pelo mundo), há ainda a possibilidade dessa rede ser centralizada e continuar sendo distribuída. É o caso de Blockchains utilizadas em indústrias privadas (LI et al., 2017), onde a rede de computadores interna valida os blocos, porém ninguém de fora da empresa tem acesso aos dados e nem é capaz de validar as informações lá armazenadas.

A Bitcoin se tornou um ícone do liberalismo econômico, como analisado por Santos (2016), por meio da participação dos mineradores e indivíduos que realizam movimentações financeiras virtuais sem a regulação do Estado. Logo, pouco se aplicou o conceito de Blockchain em um cenário centralizado, como uma moeda digital emitida por banco central.

A implementação de uma moeda virtual centralizada, caso implementada para funcionar em um banco central, traz diversas vantagens para uma nação, como a ausência de gastos com emissão de papel-moeda e transparência nas movimentações de capital para todos os cidadãos.

1.2 OBJETIVO

Acreditando no potencial disruptivo de uma moeda virtual a um baixo custo de manutenção, é apresentado neste trabalho a implementação de um sistema de criptomoeda centralizado, podendo ser aplicado em um cenário nacional, sendo assim regulada por um banco central. São objetivos deste trabalho de conclusão de curso:

- Construção de uma interface gráfica para inserção e visualização de dados na Blockchain;
- Desenvolvimento de uma central de cálculos que verifica a legitimidade de cada uma das transações inclusas;
- Implementação de um mecanismo de validação dos blocos com as transações armazenadas.

1.3 LEVANTAMENTO BIBLIOGRÁFICO

Ao pesquisar nas bases de dados IEEE e ACM, nota-se que muito pouco foi desenvolvido na área de Blockchain voltada para sistemas centralizados. A imensa maioria

desses sistemas foca em centralizar um livro digital em indústrias, garantindo segurança e transparência em suas aplicações.

Para efeitos de comparação, a busca das palavras-chave “central bank cryptocurrency” na plataforma IEEXplore retorna apenas doze resultados, os quais nenhum de fato tratam sobre o desenvolvimento de uma criptomoeda emitida por um banco central. Já na plataforma ACM Digital Library, uma busca avançada pelos artigos os quais o abstract contenha as mesmas palavras-chave (“central bank cryptocurrency”) retorna apenas dois resultado, os quais também não tratam a aplicação deste trabalho de conclusão de curso.

Apenas um trabalho que compartilha objetivos com este foi encontrado, a Fedcoin, publicada e encontrada fora de bases de dados científicas oficiais. Koning (2016) propõe em sua publicação uma moeda estável emitida por uma unidade federativa. O autor também promove análises estruturais pertinentes sobre alguns detalhes de seu funcionamento, tais como até que ponto os cidadãos devem ter acesso a uma moeda não-tangível emitida por banco central, até que ponto a moeda ofereceria anonimato aos usuários, se deveria ou não ter limite de depósito.

A Fedcoin é apenas um estudo de viabilidade de uma moeda emitida por banco central, onde o autor traz um modelo a ser seguido e algumas diretrizes que a moeda seguiria para garantir sua eficácia. Tal estudo foi útil para o desenvolvimento deste trabalho de conclusão de curso, visto que alguns pontos estruturais importantes para a construção de uma criptomoeda centralizada em um banco central foram discutidos pelo autor.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão evidenciados os fundamentos teóricos necessários para a realização deste trabalho. Seguindo uma ordem lógica, será explicado o funcionamento de uma Blockchain, uma visão geral sobre as criptomoedas e a Bitcoin (incluindo comparações entre alguns algoritmos de consenso), sistemas baseados na tecnologia da Blockchain e aplicações de Blockchain para sistemas centralizados.

2.1 FUNCIONAMENTO DA BLOCKCHAIN

Segundo Tapscott e Tapscott (2016) *"a Blockchain é um livro digital incorruptível de transações econômicas que podem ser programados para registrar não apenas transações financeiras, mas virtualmente tudo que possui valor"*.

O funcionamento de uma Blockchain foi idealizado por Satoshi Nakamoto, por meio de seu agora famoso artigo, publicado em 2008. O documento trazia detalhes sobre o funcionamento de um sistema de pagamentos ponto-a-ponto, chamado de Bitcoin (cuja tecnologia por trás de seu mecanismo fora chamada posteriormente de Blockchain) (NAKAMOTO, 2008).

Uma Blockchain é, como em sua tradução literal, uma cadeia de blocos, onde cada bloco é formado por um conjunto de dados (no caso das criptomoedas, transações) que são armazenados em um livro de registros distribuído. Uma vez registrados em um bloco, os dados se tornam imutáveis, de modo que uma simples alteração invalide todos os blocos subsequentes (TASATANATTAKOOL, 2018).

Um bloco genérico é formado por uma lista de *hashes* de transações independentes. Quando o bloco atinge sua completude, um algoritmo de *hashing* utiliza então o *hash* do bloco anterior, mais todos os *hashes* das transações do bloco corrente para gerar o *hash* do bloco atual. O próximo bloco, quando gerado, utilizará novamente o *hash* do bloco anterior a ele para formar seu próprio *hash* (NAKAMOTO, 2008).

Devido a cadeia de blocos validada pelos *hashes*, uma simples alteração em um carácter de uma transação de um dos blocos faria com que a *hash* do bloco em questão mudasse totalmente, invalidando o *hash* daquele bloco e, por consequência, todos os outros blocos subsequentes (IANSITI; LAKHANI, 2017).

Uma vez que o bloco esteja completo, inicia-se o processo popularmente conhecido como mineração. Trata-se na verdade, de um algoritmo de prova de trabalho (do inglês, proof-of-work, PoW) no qual os "mineradores" buscam adivinhar um elemento do bloco chamado nonce (number used once), que nada mais é que um número aleatório que entra como fator na determinação do *hash* do bloco (NAKAMOTO, 2008).

Os mineradores trabalham procurando o valor do nonce que faz com que o *hash* daquele bloco tenha uma característica pré-definida (no caso da rede Bitcoin, os mineradores tentam encontrar um nonce que resulte em um *hash* de bloco que inicie com x zeros, sendo x chamado de "dificuldade" da mineração) (VUJICIC et al., 2018). A Figura 1 ilustra o processo de busca pelo nonce do bloco responsável por gerar o hash-alvo.

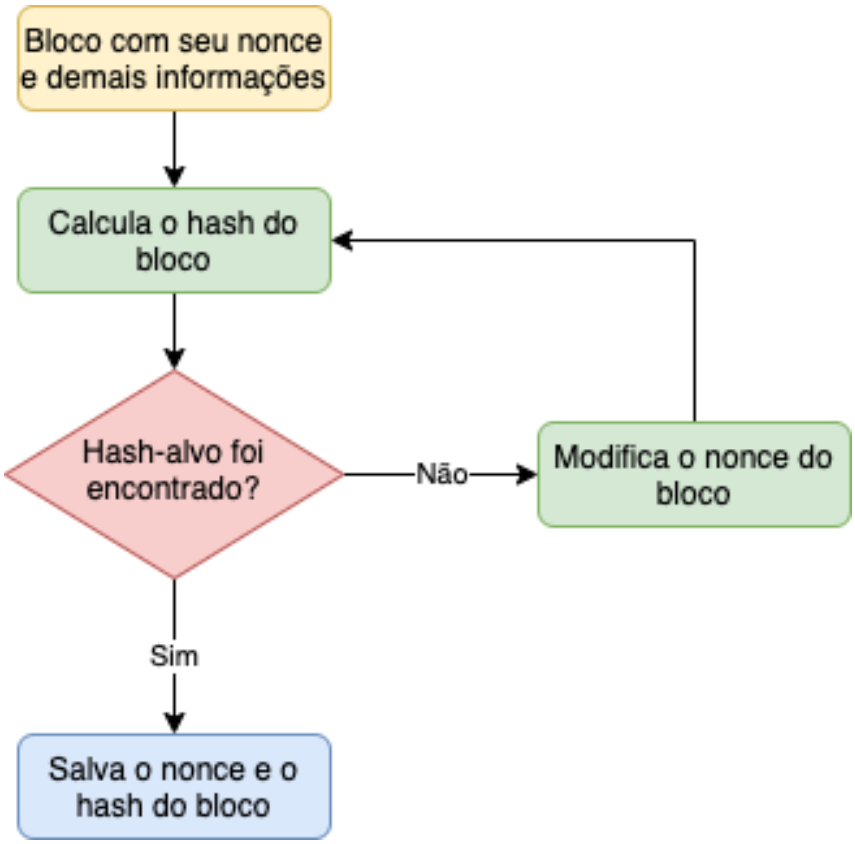

Figura 1: Fluxo da busca pelo nonce do bloco que resulta no hash-alvo. Fonte: Autoria própria.

Quando o nonce-objetivo é encontrado, toda a rede é avisada e os outros nós verificam se o bloco é realmente válido. Uma vez que o bloco é considerado válido por 51%

da rede, ele é então adicionado à Blockchain e outro bloco se inicia, trazendo a recompensa aos envolvidos na validação dos dados, de acordo com as políticas de consenso do sistema (JUDMAYER et al., 2017).

Todos esses elementos fazem com que a Blockchain seja um sistema muito seguro por seu caráter distribuído e que seja perfeitamente aplicável a diversos setores da tecnologia.

2.2 CRIPTOMOEDAS

As criptomoedas são aplicações da Blockchain para sistemas financeiros. São moedas digitais com seus próprios sistemas de transferência e armazenamento, com suas regras regidas por seu código-fonte. São moedas virtuais que podem ser transferidas de pessoa para pessoa sem a regulação de uma entidade central, seguindo as diretrizes P2P.

2.2.1 BITCOIN

A primeira e mais famosa criptomoeda que surgiu foi a Bitcoin, onde sua criação se deu com a publicação do artigo de Satoshi Nakamoto. Ele propôs um sistema com servidor timestamp P2P distribuído, que serve como um gerador da prova computacional da ordem cronológica de transações (NAKAMOTO, 2008). Seu artigo também evidenciava a solução do problema do gasto-duplo (possibilidade de se gastar uma mesma moeda duas vezes), o qual fora resolvido pela implementação do algoritmo de prova de trabalho.

2.2.1.1 FUNCIONAMENTO GERAL

Seguindo os princípios da Blockchain, uma moeda eletrônica é definida como uma cadeia de assinaturas digitais (VUJICIC et al., 2018). Cada transação é definida usando o *hash* digitalmente assinado da transação anterior juntamente com a chave pública do próximo dono. A chave privada é usada para assinar a transação, enquanto a chave pública é usada para verificação da transação, como mostra a Figura 2. A chave pública é mantida em uma carteira, que pode ser implementada em software, hardware ou online.

O livro de registros do Bitcoin é definido como um sistema de transição de estados, consistindo de um estado que mostra o status de propriedade de todos os Bitcoins e de uma função de transição de estado, sob a forma de transação (VUJICIC et al., 2018). A saída dessa função é um novo estado. Os resultados desse processo são mudanças de

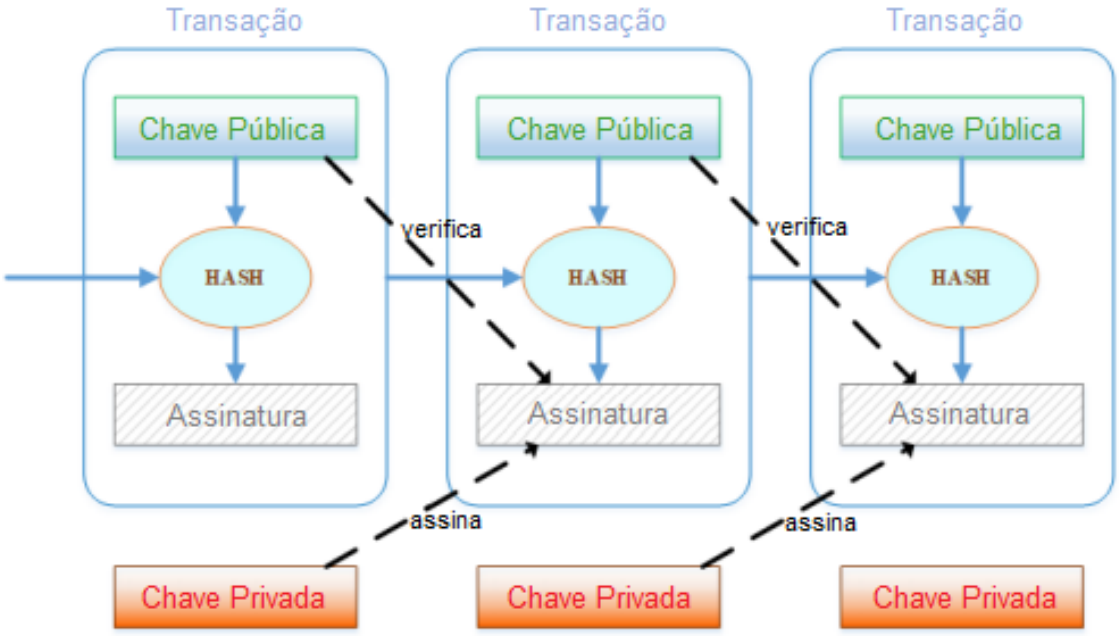

Figura 2: Uma transação na rede Bitcoin. Fonte: Vujicic et al. (2018)

estado do remetente e destinatário (se o remetente possui Bitcoins suficientes para fazer a transação), ou um erro.

Todas as transações e posses de Bitcoins da rede estão descritas de forma distribuída no livro de registros Bitcoin. Todos os nós da rede P2P detém uma cópia destes registros (Ethereum Community, 2013). Se um usuário quer enviar uma certa quantidade de moedas para outro, ele pode fazê-lo anunciando publicamente essa transação. Cabe à rede verificar a veracidade desta operação. Contudo, um usuário poderia tentar manipular a rede e emitir mais que uma transação da mesma Bitcoin para usuários diferentes (problema do gasto duplo), o que é impedido pelo mecanismo "proof-of-work".

2.2.1.2 PROOF-OF-WORK

O problema do gasto duplo é previnido na rede Bitcoin por exigir uma prova de trabalho (proof-of-work) de cada nó que verifica uma transação. Os nós têm que fazer cálculos para provar que são membros válidos da rede. Enquanto o poder computacional dos nós honestos for maior que o poder computacional dos atacantes, o sistema continuará consistente e todas as transações legítimas irão ocorrer sem nenhum problema (TSCHORSCH; SCHEUERMANN, 2016).

Um conjunto de transações, juntamente com o *hash* do bloco anterior, mais um nonce, constitui um bloco. Um servidor timestamp faz a *hash* do bloco e o anuncia

publicamente, provando que os dados dentro do bloco tem que ter existido no momento em que o algoritmo de *hashing* foi executado. O servidor timestamp tem que verificar que o timestamp do bloco é maior que o timestamp do bloco anterior na cadeia e menor que duas horas à frente. Essas *hashes* são ligadas em cadeia, formando o que é chamado de Blockchain, como na Figura 3. A grande importância da Blockchain é que todas as transações podem ser consultadas a qualquer momento, sendo publicamente disponível.

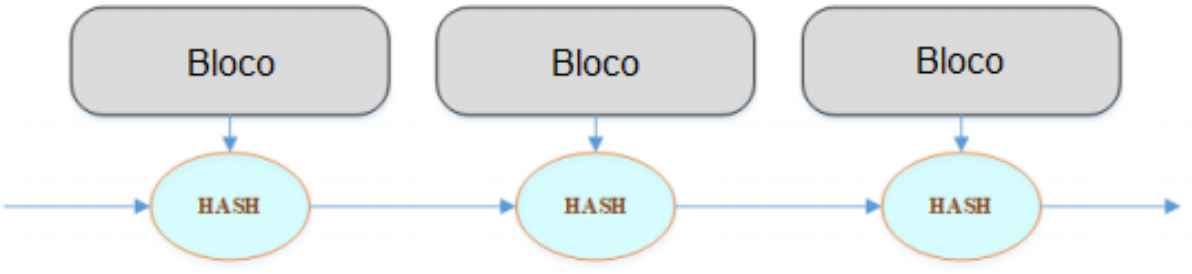

Figura 3: A cadeia de blocos. Fonte: Vujicic et al. (2018)

O sistema de prova de trabalho que a rede Bitcoin utiliza é similar ao Hashcash (BACK, 2002) e baseado no algoritmo de *hashing* SHA-256. A prova de trabalho é realizada incrementando o nonce presente no bloco até que o valor do *hash* produzido seja menor que um número x. Uma vez que essa condição é satisfeita, não pode ser desfeita sem repetir os cálculos.

Na Tabela 1 é exemplificado o papel do nonce em um bloco. Em um cenário de criptomoedas, o texto "Hello world" seria substituído pela lista de transações presentes no bloco. Ainda no exemplo, com o nonce 614 pode ser observado que a saída do *hash* se inicia com 000, sendo este o chamado *hash*-alvo (com dificuldade de três zeros) (DRESCHER, 2018).

Se por algum motivo qualquer dado for modificado por um atacante malicioso, então todos os blocos subsequentes terão *hashes* inválidos. A regra é que a maior cadeia que possuir a maioria do consenso da rede é a correta. Logo, se um atacante desejar modificar um bloco, ele terá que ter poder computacional para superar a votação da maioria dos nós honestos (VUJICIC et al., 2018).

As transações em um bloco passam pelo algoritmo de *hash* e são estruturadas como uma árvore de Merkle (Merkle Tree). A árvore de Merkle é um tipo de árvore onde a raíz dos nós-folha é uma *hash* de seus filhos. A Figura 4 mostra um bloco que consiste na árvore de Merkle resultante dos *hashes* das transações. Qualquer inconsistência na árvore será refletida na cadeia. A árvore é utilizada para liberar espaço de armazenamento ao

Nonce	Texto cujo hash será gerado	Saída
0	Hello World! 0	4EE4B774
1	Hello World! 1	3345B9A3
2	Hello World! 2	72040842
3	Hello World! 3	02307D5F
	...	
613	Hello World! 613	E861901E
614	Hello World! 614	00068A3C
615	Hello World! 615	5EB7483F

Tabela 1: Exemplo da relação entre o nonce e o *hash*. Fonte: Drescher (2018)

gravar a Blockchain nos nós. Após as transações serem incorporados em um bloco e esse bloco verificado, a rede descarta todos os *hashes* da árvore exceto o nó-raíz, incluso no cabeçalho do bloco. Isso ocorre pois a rede Bitcoin implementa uma "verificação de pagamento simplificada" (Simplified Payment Verification, SPV), que não requer que os nós da rede guardem um registro completo das transações, mas só a cópia dos cabeçalhos dos blocos da cadeia mais longa (NAKAMOTO, 2008).

A primeira transação em um bloco cria uma nova moeda, que se torna propriedade do criador do bloco (NAKAMOTO, 2008). Essa é a característica que estimula os nós da rede a verificar as transações e permanecerem honestos, além de colocar novas moedas em circulação (já que não existe uma autoridade central emissora de Bitcoins). Essa transação é chamada de "coinbase transaction" (VUJICIC et al., 2018). À tarefa de verificar a autenticidade das operações em troca de recompensa, é atribuída o nome de "mineração".

2.2.2 ALTCOINS

Após o surgimento e consolidação da Bitcoin, outras criptomeodas surgiram ao longo do tempo, cada uma com seu próprio sistema, políticas e particularidades. Por não serem a principal moeda digital, passaram a fazer parte do grupo das moedas alternativas, as "altcoins".

A Ethereum, por exemplo, é uma altcoin que não tem o mercado financeiro

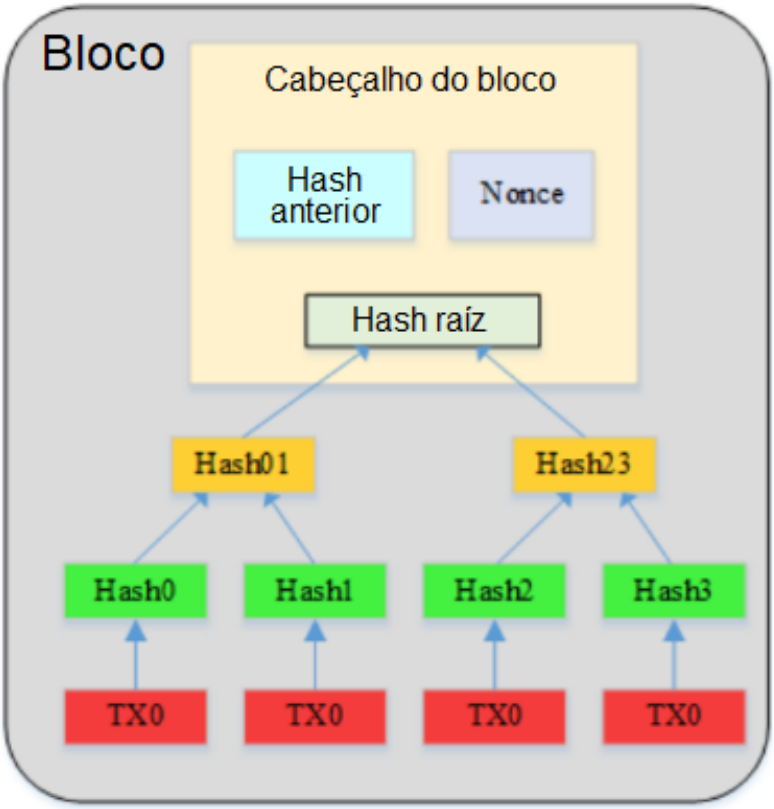

Figura 4: A árvore de Merkle. Fonte: Vujicic et al. (2018)

como objetivo principal. Ela é uma plataforma de aplicação descentralizada de "smart contracts" (contratos inteligentes), desenvolvida pelo pesquisador e programador Vitalik Buterin. Ela utiliza computação distribuída baseada em Blockchain que permite o processamento de contratos inteligentes em sua cadeia de blocos (TASATANATTAKOOL, 2018).

Na rede Ethereum, o estado é formado por objetos chamados de contas, onde cada transição de estado é uma transferência direta de valor e informação entre as contas (Ethereum Community, 2013). Sendo assim, além de transferência de valores, as contas podem transferir também posses (como casas ou carros) em forma de código, utilizando estruturas condicionais.

Um outro exemplo de criptomoeda é a Ripple que, segundo Lewis (2014), é descrita como "um banco medieval digitalizado". É tanto uma moeda digital quanto um protocolo de pagamentos.

Em um banco medieval, Alex transferiria dinheiro para Beth (em outra localidade) da seguinte forma: Alex deposita o dinheiro em um banco e fala uma senha; a agência de Alex então telefona para a agência de Beth e diz a senha; o agente da agência

de Beth então aguarda alguém que diga a senha correta; Beth vai até a agência, diz a senha e recebe o dinheiro.

Dessa forma, os fundos foram transferidos de forma que a moeda não tenha se movido fisicamente. Pode ser observado também que fez-se necessária a presença de um agente de confiança para que a transação fosse completada.

De forma análoga a esse sistema, no sistema da Ripple, lojas e websites fazem o papel de agentes bancários e a ligação telefônica é substituída por mensagens eletrônicas. Alex realiza login em uma gateway Ripple de sua preferência, deposita dinheiro em sua gateway, instrui a liberar fundos para a gateway de Beth, que coleta os fundos por sua própria gateway (não só dinheiro, mas qualquer bem de valor, no caso da Ripple) (LEWIS, 2014).

2.2.3 MECANISMOS DE CONSENSO

Em um cenário distribuído é extremamente necessária a presença de regras que regem o comportamento da rede e determinam o papel de cada nó. Os participantes acordam quanto à existência, valores e histórico dos estados (CHALAEMWONGWAN, 2018). Este acordo mútuo é chamado de mecanismo/algoritmo de consenso.

Nesta seção serão explicitados os dois principais mecanismos de consenso que podem ser aproveitados para realização deste trabalho, os quais surgiram após problemas encontrados em seu precursor (proof-of-work).

2.2.3.1 PROBLEMAS COM O PROOF-OF-WORK

Um grande problema de criptomoedas que fazem uso do algoritmo de consenso de prova de trabalho (proof-of-work) é o alto custo operacional para sustentar a rede. De acordo com o site Digiconomist (2018), o gasto de energia elétrica com mineração na rede Bitcoin chega a 65,8TWh, o necessário para fornecer energia para 6 milhões de casas estadunidenses. Isso ocorre pois o algoritmo de prova de trabalho faz com que os mineradores compitam uns contra os outros, resultando em muitos cálculos desnecessários, em troca da recompensa da mineração.

O proof-of-work também acarretou no surgimento das “pools de mineração”, uma espécie de rede virtual de computadores que distribui o trabalho entre computadores ao redor do mundo e, quando encontram o nonce do bloco, distribuem a recompensa da mineração entre os computadores da pool. Essa prática acaba levando à centralização da

rede (BEIKVERDI; SONG, 2015), podendo acarretar no chamado “ataque 51%” (TOSH et al., 2017).

2.2.3.2 PROOF-OF-STAKE

Para tentar solucionar esses problemas, surgiu o algoritmo de consenso de prova de participação (proof-of-stake, PoS). O sistema usa um processo de eleição onde um nó da rede é escolhido aleatoriamente para validar o bloco seguinte.

Para se tornar um validador, um nó precisa fazer um depósito de garantia na rede, chamada de “participação” e, só então, entra o algoritmo de eleição (TOSH et al., 2017). O tamanho da participação determina as chances de um validador ser escolhido para validar o próximo bloco. Em um cenário fictício, Bob faz um depósito de 1000 moedas e Alice faz um depósito de 100 moedas. Com o mecanismo de prova de participação, Bob passa a ter 10 vezes mais chances que Alice de ser escolhido pela rede para validar o bloco corrente.

Quando um bloco é validado, ele é então adicionado à Blockchain e as taxas de transferências cobradas em cada transação são repassadas ao validador do bloco (que fora escolhido pelo algoritmo de eleição), como forma de recompensa pelo esforço computacional. O que impede um validador de aprovar transações inválidas e fraudar a rede, é o depósito feito por ele anteriormente, antes de ser escolhido pelo algoritmo de eleição. Se um validador agir de forma fraudulenta, parte de seu depósito de segurança será confiscado pela rede, de forma que a recompensa pela validação seja menor que o valor perdido no depósito.

Apesar de todas as vantagens que a prova de participação oferece, alguns pontos fracos podem ser observados. Uma das fraquezas do algoritmo é que a validação dos blocos tende a ser centralizada nas mãos dos nós que possuem maior riqueza, podendo gerar um ciclo vicioso, já que os nós com maior posse continuarão aumentando sua riqueza por receberem as recompensas das validações.

Outra desvantagem deste mecanismo é que, mesmo que um validador valide um bloco fraudulento, apenas terá parte de seu depósito tomado pela rede, podendo voltar a ser escolhido para validar outros blocos no futuro. Para tentar driblar essas e outras falhas, um novo algoritmo de consenso foi desenvolvido, o proof-of-authority.

2.2.3.3 PROOF-OF-AUTHORITY

O algoritmo de consenso de prova de autoridade (proof-of-authority, PoA) é uma forma modificada do proof-of-stake, onde ao invés da participação se dar por valor monetário, se dá na verdade pela identidade de um validador. Nesse caso, a identidade significa a correspondência entre a identificação pessoal de um validador na plataforma com os documentos oficiais desta mesma pessoa.

Disponibilizar identidade como participação significa mostrar voluntariamente quem a pessoa é em troca do direito de validar blocos. Isso significa que tanto os benefícios que um participante obtém, quanto atitudes negativas que ele possa tomar, são públicas.

Para que o conceito funcione da forma correta, três condições devem ser satisfeitas (POA Network, 2017):

1. A identidade tem que ser legítima: precisa haver um padrão e um robusto processo de verificação para garantir que os validadores sejam realmente quem eles dizem que são.

2. Elegibilidade para utilizar identidade como participação deve ser difícil de ser obtida: para que o direito de ser um validador se torne merecido, valioso e desagradável de ser perdido.

3. O procedimento para obtenção de autoridade deve ser o mesmo para todos os validadores: para garantir que toda a rede entenda o processo e possa confiar em sua integridade.

Uma grande vantagem do uso do algoritmo de prova de autoridade sobre o algoritmo de prova de trabalho é a inexistência do processo de mineração. Assim como na prova de participação os nós não precisam competir entre si em troca de recompensa, diminuindo drasticamente o custo de energia para manter a rede (NAUMOFF, 2017).

2.3 SISTEMAS BASEADOS EM BLOCKCHAIN

Aplicações diferentes utilizam a Blockchain de maneiras diferentes, de acordo com a necessidade de cada uma. É dever do projetista escolher o melhor modelo para trabalhar, levando em consideração os algoritmos de consenso existentes, fazendo adaptações e criando novas formas de se implementar uma Blockchain.

Contudo, nem sempre é indicado que uma aplicação seja armazenada em um livro de registros distribuídos e imutável, como mostra o artigo de Lo et al. (2017). O trabalho estuda quando é viável aplicar os conceitos de Blockchain em um sistema ou se um banco de dados convencional já seria suficiente. Utilizando seu estudo e observando o fluxograma da Figura 5, é notável que este trabalho de conclusão de curso poderia ser implementado perfeitamente em uma Blockchain.

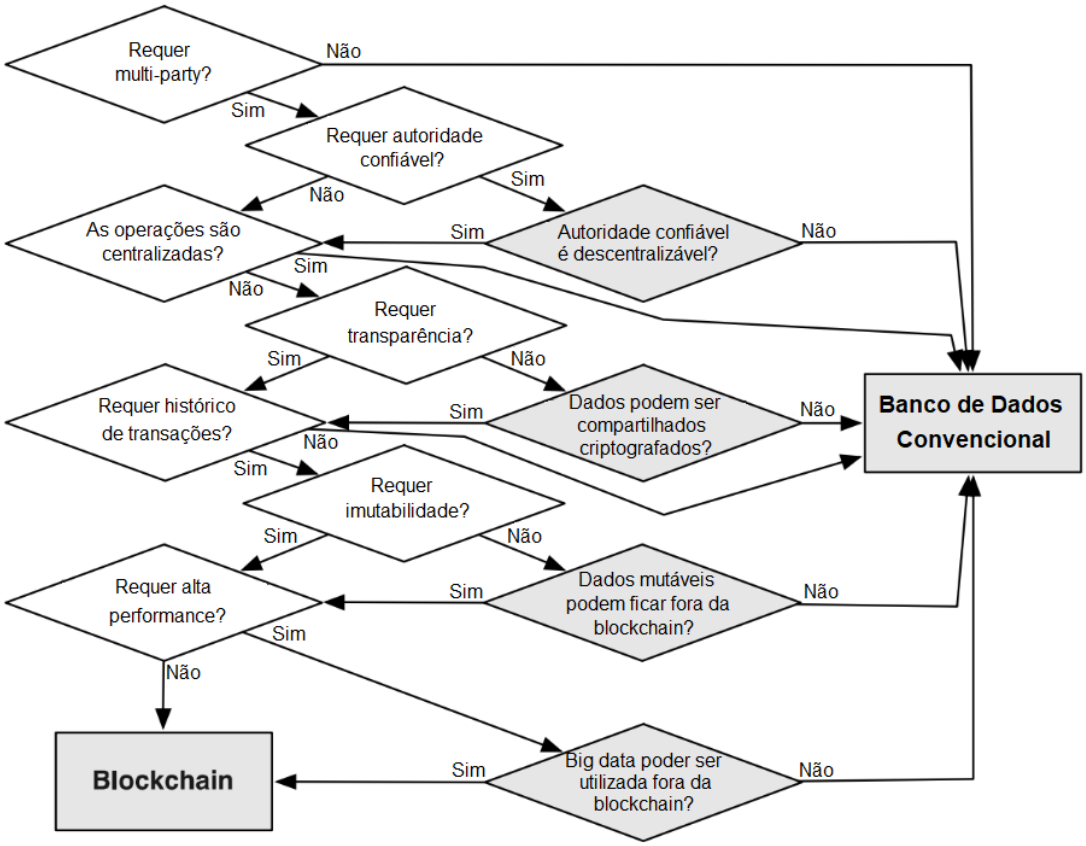

Figura 5: Blockchain ou banco de dados convencional? Fonte: Lo et al. (2017)

Como evidenciado por Gatteschi et al. (2018), os princípios da Blockchain podem ser aplicados em diversas áreas para armazenar vários tipos de dados, como transações financeiras, registros públicos, registros privados, identificação, atestados, chaves de bens físicos, bens intangíveis, entre outros.

Uma das aplicações encontradas em pesquisas às bases de dados foi a implementação de um sistema de Autenticação de Arquivos pela Blockchain (Notarizing Files over the Blockchain, NFB). O NFB é um protocolo que garante a comunicação entre uma solução centralizada de arquivamento de documentos e a Blockchain, com o intuito de validar arquivos em uma Blockchain (MAGRAHI; SENOT, 2018). A tecnologia provê três serviços principais: arquivamento de documentos, recuperação de documentos e prova de

existência de documentos.

Utilizando o protocolo NFB, os usuários podem provar publicamente que um documento existe (ou não), retornando seus metadados indexados na solução de arquivamento seguro e com base nas informações traçadas no Blockchain, durante um período.

Há também uma aplicação que propõe a criação uma plataforma voltada para a indústria de seguros. A ideia por trás do modelo de Raikwar et al. (2018) é implementar os processos de uma seguradora em forma de contratos inteligentes (smart contracts), colocando os contratos em uma plataforma distribuída habilitada para Blockchain, tanto para execução dos contratos quanto para armazenamento dos resultados.

Existe também a iniciativa de Koç et al. (2018), que sugere mudanças em sistemas de eleição. Seu trabalho resulta na tecnologia de voto virtual (e-voting), agindo como um contrato inteligente na rede Ethereum, utilizando as carteiras Ethereum e a linguagem Solidity.

Como observado existem várias formas de se utilizar a Blockchain para armazenar informações de forma distribuída e segura, basta o projetista saber a real necessidade de sua utilização e saber aplicar corretamente.

2.4 APLICAÇÃO DE BLOCKCHAIN EM SISTEMAS CENTRALIZADOS

Ao pesquisar nas bases de dados é nítido que a quantidade de estudos a cerca de aplicações de Blockchain em sistemas centralizados é muito menor que a de sistemas descentralizados, o que remete a ideia de que o uso da Blockchain de forma centralizada não foi tão explorada.

A maior recorrência de aplicações centralizadas de Blockchain são provenientes de companhias privadas/indústrias, onde é necessário o armazenamento permanente de determinados registros e, ao mesmo tempo, que haja disponibilidade de infraestrutura para manter o sistema ativo distribuidamente dentro da organização.

O trabalho de Li et al. (2017), por exemplo, propõe uma arquitetura de Blockchain projetada especificamente para atender os padrões da indústria. A proposta utiliza a ideia de “cadeias de satélites”, que podem executar diferentes protocolos de consenso em paralelo, melhorando as premissas de escalabilidade do sistema.

Na mesma proposta é apresentado um modelo interessante, onde os nós da rede podem assumir diferentes papéis: clientes, validadores, auditores ou reguladores. Cada

um dos papéis possuindo suas particularidades e finalidades bem definidas (LI et al., 2017).

3 METODOLOGIA E DESENVOLVIMENTO

Neste capítulo serão explicitados detalhes do desenvolvimento deste trabalho trazendo, além de uma visão geral do sistema implementado, uma apresentação de como se dão os usuários, transações, blocos e a mineração na Blockchain desenvolvida.

3.1 VISÃO GERAL

Esta seção tem como objetivo explicitar de forma geral o funcionamento do sistema, apresentar seus agentes e citar quais as ferramentas utilizadas em seu desenvolvimento.

3.1.1 MODELO

A dinâmica do sistema desenvolvido é apresentada na Figura 6. Como pode ser observado na mesma, estão presentes três agentes: a interface gráfica da aplicação, a central de controle e o validador ou minerador.

A interface gráfica foi criada com o intuito de inserir dados no sistema, através de usuários (que possuem contas) que poderiam se autenticar no sistema e transferir dinheiro para outros usuários, tendo em mãos apenas o número de identificação (*idn*) do destinatário. Porém, como o desenvolvimento de um sistema de login/autenticação não é um objetivo do trabalho, as inserções de transações no sistema são realizadas de forma genérica: o manipulador da interface precisa apenas digitar de qual conta o dinheiro está saindo e para qual conta está indo. Contudo, mesmo sem a presença de uma aplicação de autenticação, o sistema realiza todas as verificações de chaves e assinaturas necessárias, simulando como seria se houvesse um sistema de autenticação funcional.

A cada inserção de transação adicionada pelo manipulador da interface, a central verifica se aquela transação é possível realizando diversas checagens, tais como se as contas existem, se possuem saldo, entre outras.

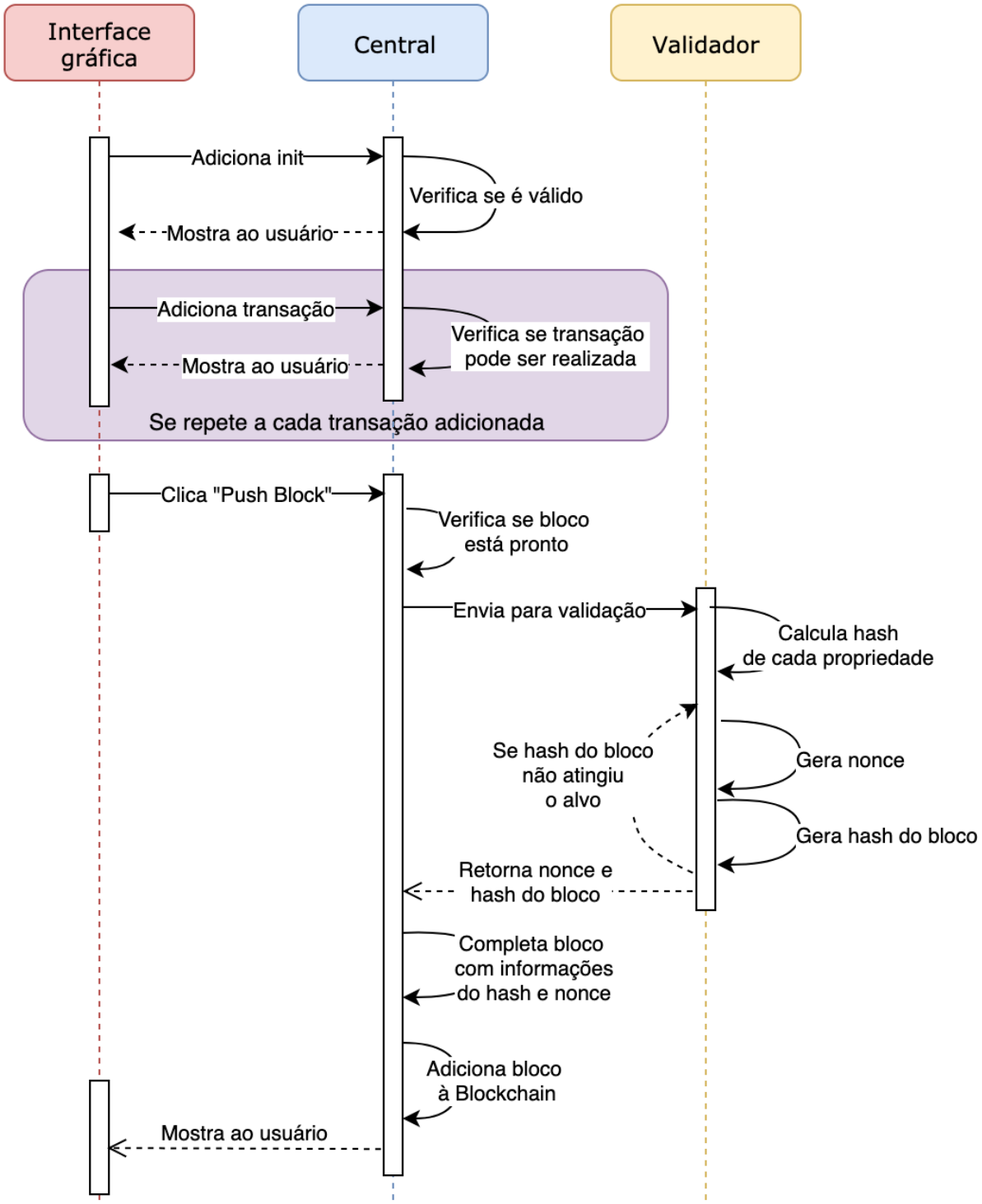

Figura 6: Diagrama de sequência do sistema para inserção de blocos na rede. Fonte: Autoria própria.

Quando a central verifica que todas as transações são possíveis e o manipulador da interface gráfica solicita a inserção do bloco, a central envia todos os dados para a unidade validadora, que valida todos os dados recebidos, calculando todos os *hashes* necessários das propriedades dos blocos. Com a validação concluída, o bloco pode ser inserido na rede pela central.

3.1.2 CODIFICAÇÃO E FERRAMENTAS

O sistema foi desenvolvido utilizando Python 3.7.3 (PYTHON.ORG, 2019). A utilização dessa tecnologia se deve à fluência do autor na linguagem e também à existência prévia de bibliotecas que tiveram papel fundamental no desenvolvimento do sistema.

Exemplos de bibliotecas utilizadas são a PyCryptodome (2020) para diversas tarefas relacionadas a criptografia e a PyQt5 (2020) para construção das interfaces gráficas do sistema. É válido dizer que o uso da biblioteca PyCryptodome se fez necessário pois a mesma, após descobertas relevantes de falhas de segurança (CVE, 2018), substituiu a biblioteca PyCrypto.

Como IDE para programação e execução do projeto, por ser uma ferramenta de fácil manipulação, atual e bastante popular entre desenvolvedores, foi utilizado o Microsoft Visual Studio Code durante todo o desenvolvimento (MICROSOFT, 2020). Além disso, todo o versionamento do projeto foi controlado pelo Github e o código-fonte encontra-se aberto (REIS, 2020).

Nas próximas seções serão explicitados de forma detalhada cada uma das fases do processo de inserção de novos blocos na rede, com o intuito de esclarecer o funcionamento do sistema.

3.2 USUÁRIOS

Os integrantes da Blockchain que movimentam fundos por meio de transações são as contas dos usuários. São essas contas as responsáveis por guardar dados dos usuários e possibilitar a transferência de moedas entre eles. No sistema, as contas se apresentam em um arquivo de formato json, nomeado *accounts.json*. Uma vez que o sistema não possui uma funcionalidade para criação de novas contas via interface gráfica, todas as contas existentes foram adicionadas manualmente no arquivo *accounts.json*. O arquivo pode ser manipulado por qualquer editor de arquivo-texto, tornando simples a criação, exclusão ou edição de contas.

```
"account_list": [{
    "idn": "39620880080",
    "public_key": "-----BEGIN PUBLIC KEY-----\nMIGfMA0GCSqGSIb3DQEBAQUAA4GN
    "private_key": "-----BEGIN RSA PRIVATE KEY-----\nMIICXAIBAAKBgQDayPGjxv
}, {
    "idn": "17040189003",
    "public_key": "-----BEGIN PUBLIC KEY-----\nMIGfMA0GCSqGSIb3DQEBAQUAA4GN
    "private_key": "-----BEGIN RSA PRIVATE KEY-----\nMIICXQIBAAKBgQDvjaVFaV
}, {
    "idn": "21297039092",
    "public_key": "-----BEGIN PUBLIC KEY-----\nMIGfMA0GCSqGSIb3DQEBAQUAA4GN
    "private_key": "-----BEGIN RSA PRIVATE KEY-----\nMIICXAIBAAKBgQC9NIjBEe
}, {
```


Figura 7: Trecho do arquivo responsável por armazenar as contas da rede em formato json. Fonte: Autoria própria.

Como evidenciado na Figura 7, cada conta armazenada possui apenas três dados:

- Número de Identificação (*idn*): atributo responsável por armazenar o número de código identificador do usuário (podendo ser, por exemplo, o CPF de uma pessoa), em formato string;
- Chave pública (*public_key*): atributo responsável por armazenar a chave pública (RCA de 1024 bits, decodificada em formato UTF-8) da conta;
- Chave privada (*private_key*): atributo responsável por armazenar a chave privada (RCA de 1024 bits, decodificada em formato UTF-8) da conta;

Além dos atributos citados, novas propriedades podem ser adicionadas em caso de implementação de um sistema real, com a finalidade de que mais dados sobre o usuário sejam armazenados (tais como endereço, telefone e/ou e-mail).

3.3 TRANSAÇÕES

As transações são os principais registros armazenados nos blocos. Cada bloco armazena um vetor com suas respectivas transações, especificando as movimentações financeiras entre os usuários. As informações digitadas nos campos da interface gráfica (que em código estão presentes no arquivo *AddTransationsWindow.py*) da Figura 8 são utilizadas nas transações.

O campo *Init Destination* deve ser preenchido com o número de identificação do usuário que receberá o valor (*Init Value*) que está sendo criado naquele bloco. Cada bloco

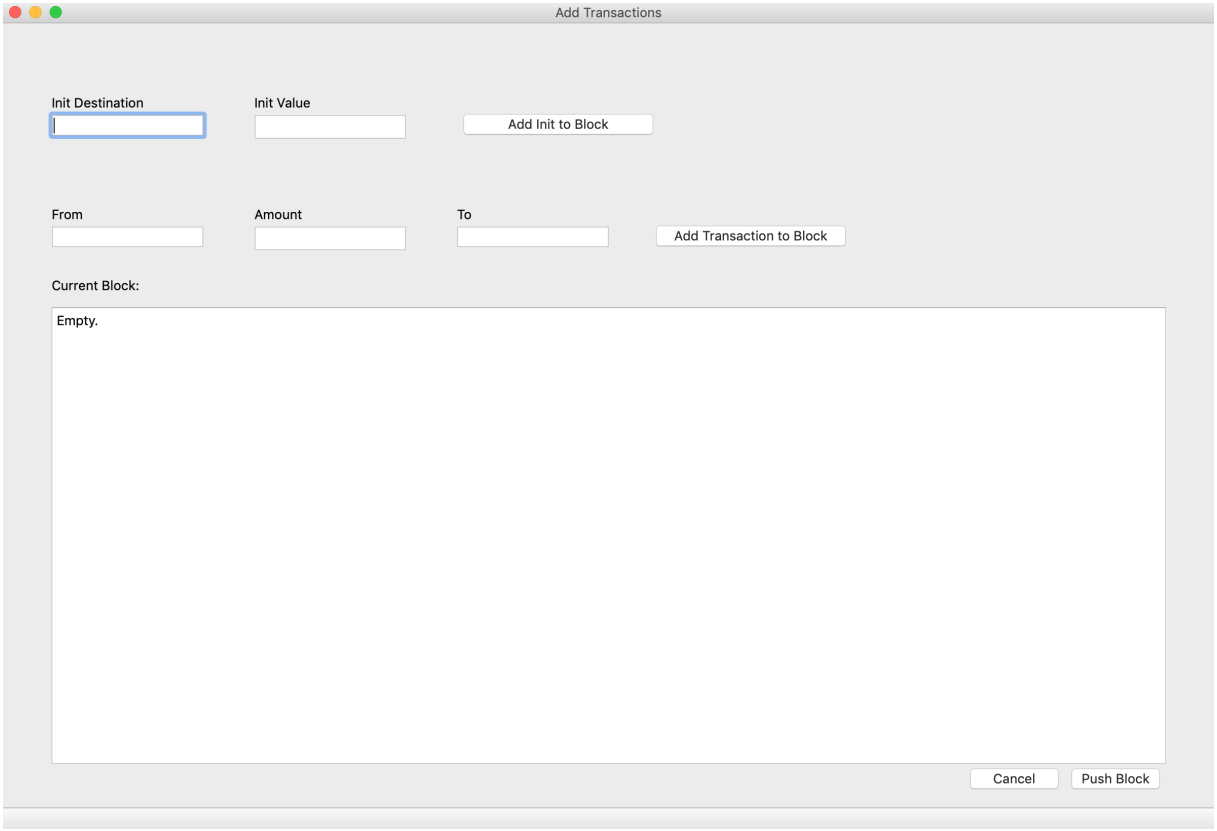

Figura 8: Interface para adição de transações em um novo bloco. Fonte: Autoria própria.

possui apenas um init, representando a quantidade de dinheiro que está sendo gerada com a criação daquele nó na Blockchain.

Os campos *From*, *Amount* e *To* são responsáveis por receber respectivamente as informações referentes ao idn do remetente, o valor que está sendo transferido e o idn do destinatário dos fundos.

Quando o manipulador da interface clicar em *Add Transaction to Block*, o método *clickedAddTransaction* será disparado. Após o gatilho, serão verificados se o número de identificação do remetente e do destinatário existem e se o remetente possui fundos para realizar a transferência. Caso os requisitos sejam atingidos, a transação é adicionada no bloco de texto *Current Block*, significando que a transação foi aceita e está apta a ser registrada no bloco corrente.

A verificação de saldo de uma conta é feita pela máquina central (no método *__check_if_has_balance* do arquivo *PushTransactionToNextBlock.py*), que varre todos os blocos anteriores (e até mesmo o corrente) para garantir que o remetente possui os fundos que deseja enviar ao destinatário.

Ao clicar no botão *Add Transaction to Block*, o *timestamp* do momento do clique é adicionado à transação, registrando assim o horário e data da transferência. Para que haja garantia de autenticidade, a assinatura do remetente é adicionada à transação e conferida logo em seguida pela central. No final do processo a transação é representada por uma string, como na Figura 9.

```
"1573188320.95665;17040189003;23.51;39620880080": "9746380f8fa0c1"+
"209bdf797bb2566e99a0f69463eb49710f1d3618837b9453446227e18a1c546e8"+
"db2f757d2f68005cbbf8ad1375eb5dffbbc528d4944492398c5277f9230ec725e"+
"5679375e7687a5e3b77f5855f927d84771fd823f8cfa1738db97c5586849a49cb"+
"f90488da13c0b7a1ef3af08ddafeb13f4447ee7b440b419"
```

Figura 9: Representação de uma transação, em dicionário de strings. Fonte: Autoria própria.

A representação de uma transação se trata de um dicionário (conjunto chave-valor) composto por duas strings, onde a chave é a união do timestamp do momento da inclusão da transação, com o idn do remetente, total transferido e idn do destinatário (separados por ponto e vírgula) e o valor se trata da assinatura SHA256, utilizando a chave privada do remetente. No código do sistema, a chave deste dicionário é formada no método *form_transaction*, presente no arquivo *CentralCore.py*, enquanto seu valor (assinatura) é atribuído um pouco mais adiante, no método *push_transaction* do arquivo *PushTransactionToNextBlock.py*.

Por não ser o foco do trabalho, o sistema não possui mecanismo de login. Portanto, atualmente todas as transações são inseridas no sistema a partir da mesma tela pelo usuário manipulador da interface gráfica. Por consequência, a própria central assina a transação com a chave privada do remetente (resgatando-a do arquivo *accounts.json*, e assinando no método *sign_transaction* da *CentralCore.py*) e a verifica em seguida (no método *check_signature* da mesma classe). Em um cenário real, o sistema de autenticação/login seria responsável por assinar e a central realizaria apenas a verificação desta assinatura.

3.4 FORMAÇÃO DO BLOCO

Quando o botão *Push Block* é pressionado pelo manipulador da interface, todos os dados referentes ao campo *Init* e às transações são passados ao construtor do bloco corrente. Os dados referentes ao bloco são então obtidos e um novo arquivo json é gerado com as informações do novo bloco.

Para que os blocos sejam considerados completos e possam ser inclusos na rede,

devem possuir todas as seguintes propriedades:

- *timestamp*: representação de data e hora do momento do clique no botão *Push Block*, registrando quando as informações do bloco foram finalizadas.
- *previous_block_hash*: *hash* SHA256 do bloco anterior, preenchido quando o bloco está prestes a ser inserido na Blockchain;
- *init_destination*: campo responsável por armazenar o idn do destinatário do Init, sinalizando qual será o destino das moedas criadas no bloco;
- *init_value*: campo responsável por armazenar o valor do Init que será gerado pelo bloco;
- *tx_dataset*: array de dicionários, responsável por armazenar todas as transferências registradas no bloco;
- *block_nonce*: nonce calculado na fase de mineração;
- *block_hash*: *hash* SHA256 calculado na fase de mineração.

As propriedades *Init Destination*, *Init Value* e o array de transferências são obtidos na tela de adição de transações, como explicitado na Seção 3.3 deste trabalho. Quando o manipulador da interface clica no botão *Push Block*, o *timestamp* atual e o *hash* do bloco anterior são obtidos. Com todos estes dados coletados, o bloco passará então pela rotina de validação com a finalidade de se calcular o *hash* do bloco e o nonce resultante. Após este processo, o bloco estará devidamente preenchido e pronto para ser armazenado na Blockchain. Todo este processo pode ser observado no código-fonte do sistema, no método *push_block* do arquivo *PushTransactionToNextBlock.py*.

3.5 VALIDAÇÃO

Quando o manipulador da interface gráfica clica no botão *Push Block*, a rotina de validação/mineração entra em ação como parte do processo de inclusão de um novo bloco, com a finalidade de gerar o *hash* do bloco corrente (no código-fonte, esta tarefa é realizada pelo método *mine_block*, no arquivo *MiningCore.py*).

Utilizando o mecanismo de prova de trabalho (proof-of-work), a máquina validadora procura o nonce que resultará em um *hash* com dificuldade 3 (DRESCHER, 2018),

isto é, iniciado com três algarismos zero. Com a descoberta do *hash* que atende a dificuldade exigida pelo sistema, estará provado que houve esforço computacional para que o bloco fosse inserido na rede.

O *hash* do bloco é calculado com base em todos os outros dados contidos na estrutura corrente. Como ilustrado pela Figura 10, o *hash* do bloco é composto pelo *hash* da concatenação de: *hash* do timestamp; *hash* do bloco anterior; *hash* dos campos init; *hash* das transações do bloco e *hash* do nonce.

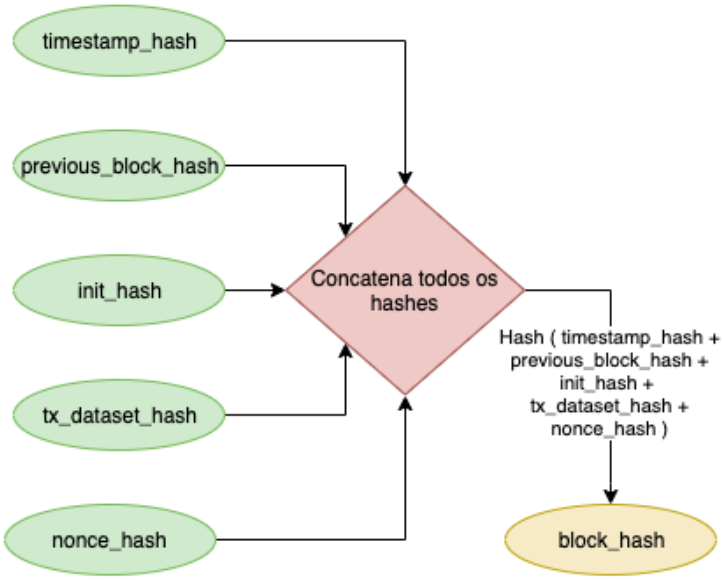

Figura 10: Formação do *hash* do bloco. Fonte: Autoria própria.

Como explicitado na Figura 11, os campos *Init Destination* e *Init Value* dão origem a um só *hash* que entra no cálculo do *hash* do bloco: *o init_hash*. Esse elemento é gerado a partir do *hash* da concatenação dos *hashes* dos dois campos referentes ao init, provenientes do bloco.

O mais importante integrante do *hash* do bloco é o *hash* do conjunto de transações. É esse elemento que garante que, caso haja qualquer alteração em qualquer uma das transações do sistema, o *hash* do bloco irá se alterar completamente (invalidando todos os blocos subsequentes da rede).

A estrutura criada para geração do *hash* do conjunto de transações do sistema é uma árvore de Merkle, apresentada na Seção 2.2.1.2 pela Figura 4. Para implementação da árvore, foi utilizado como base o algoritmo de Seo (2017), que atendeu de forma consistente as necessidades do trabalho.

A formação do *hash* dos conjuntos das transações se trata dos *hashes* das con-

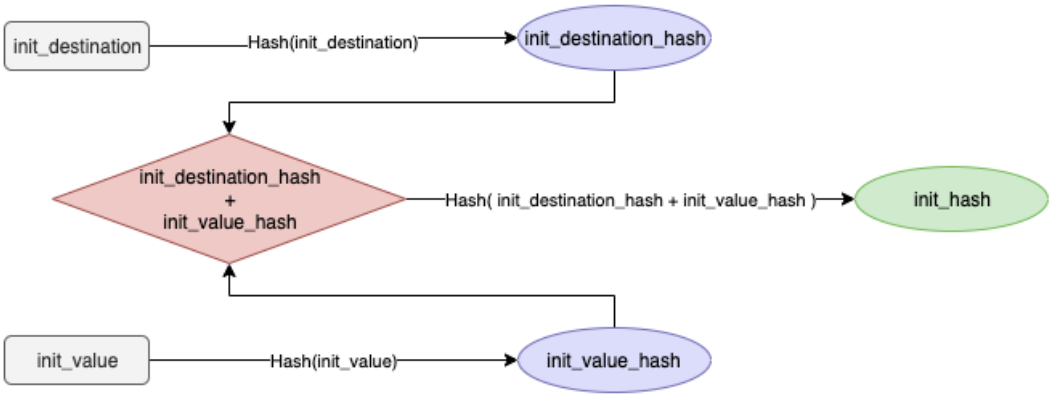

Figura 11: Formação do *init_hash*, presente na formação do *hash* do bloco. Fonte: Autoria própria.

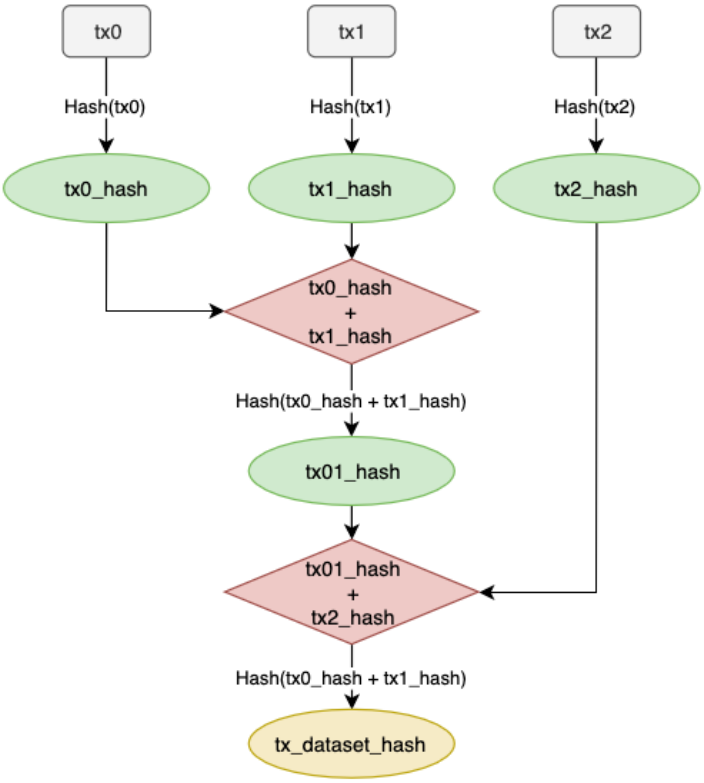

Figura 12: Formação do *tx_dataset_hash*, responsável por garantir a imutabilidade das transações. Fonte: Autoria própria.

catenações dos *hashes* de dois outros *hashes* de transações, realizados recursivamente. A Figura 12 exemplifica como seria a formação do *hash* do conjunto de transações no caso de um bloco com apenas três transações.

No sistema atual apenas uma máquina realiza todos os cálculos, podendo causar sobrecarga e/ou lentidão no processamento dos dados. Em uma implementação real, tais cálculos poderiam ser divididos entre múltiplas máquinas do sistema central, balanceando

assim a carga das operações.

Após a rotina de validação, o bloco corrente recebe seu *hash* SHA256 e finalmente segue para ser inserido na Blockchain.

3.6 INSERÇÃO NA BLOCKCHAIN

Todos os blocos completos e devidamente validados estão aptos a serem inseridos na rede. Seguindo o princípio apresentado por Vujicic et al. (2018), evidenciado na Figura 3, os blocos se tornam interdependentes, uma vez que dependem do *hash* do bloco anterior para ser considerado consistente.

No sistema desenvolvido para este trabalho de conclusão de curso, os blocos são salvos em arquivos json dentro de uma pasta dedicada a esta finalidade. Tal medida simplifica a disponibilização dos blocos.

Além do sistema verificar se a Blockchain continua consistente sempre antes de um novo bloco ser inserido, há ainda um botão no menu principal da aplicação para que seja feita a verificação retroativa dos blocos salvos no diretório do projeto. Caso alguma inconsistência seja detectada, o erro será destacado na tela principal.

4 ANÁLISE DOS RESULTADOS

Após obtenção dos resultados do desenvolvimento do trabalho de conclusão de curso, análises e discussões foram realizadas e apresentadas nas seções deste capítulo.

4.1 PRINCIPAIS AÇÕES PARA IMPLANTAÇÃO REAL DO SISTEMA

Para que o sistema seja implantado em uma aplicação real, algumas modificações e/ou implementações são imprescindíveis. As principais ações para atingimento do objetivo serão apresentadas nesta seção.

4.1.1 MÚLTIPLAS INSTITUIÇÕES FINANCEIRAS

Hoje o sistema conta com apenas uma central de cálculos que computa todas as transações, verificando-as e armazenando-as em blocos, além de controlar as informações das contas dos usuários da rede.

Em um cenário real, seria necessária a existência de várias instituições financeiras (bancos), cada uma controlando o armazenamento dos dados das contas de seus respectivos clientes. Concomitantemente, as contas deixariam então de serem escritas em arquivo-texto e passariam a persistir nos bancos de dados dessas empresas, sendo necessária uma nova arquitetura do sistema.

Uma discussão importante é acerca da comunicação entre as instituições financeiras e a Blockchain. O sistema atualmente registra os dados na rede apenas de maneira síncrona (somente a interface gráfica interage diretamente com a Blockchain). Porém, uma vez que diversos clientes de diferentes bancos podem inserir dados nos bloco, o sistema deve funcionar de maneira concorrente. Para que tal funcionalidade seja possível, deveria então ser implementado um mecanismo para controle de concorrência, permitindo que diversas instituições financeiras possam adicionar transações aos blocos simultaneamente (de forma assíncrona).

4.1.2 CENTRAL CALCULANDO SALDOS É COMPUTACIONALMENTE CUSTOSO

Atualmente o sistema confere se uma conta possui fundos para transferir através do ciclo de verificação de saldo, realizado pela central. Esse mecanismo confere se cada um dos remetentes das transações do bloco possuem recursos suficientes para realizar as transferências ali registradas. Essa tarefa é realizada de forma retroativa, observando o histórico de transações de todos os blocos passados quanto cada uma das contas recebeu e/ou transferiu para outras.

O ciclo de verificação de saldo ocorre para cada uma das transações de um bloco e é realizado pela Central, tarefa que pode ficar cada vez mais custosa a medida que o histórico de transações aumenta. Como tentativa de melhorar o desempenho da execução de tais verificações, poderia ser delegada aos mineradores a responsabilidade de se conferir o saldo de cada uma das contas remetentes.

4.2 VARIAÇÕES NO TEMPO DE CÁLCULO DE NONCE

Com a finalidade de encontrar uma relação entre o tempo de cálculo do nonce de cada bloco e a dificuldade da mineração da rede, uma bateria de execuções de cálculos de *hashes* foi realizada.

	Dificuldade (número de algarismos zero no início do hash)							
	1		2		3		4	
Amostras	Tempo (s)	Nonce	Tempo (s)	Nonce	Tempo (s)	Nonce	Tempo (s)	Nonce
1	0,0790	335	0,1382	581	1,6119	6561	246,4671	1066643
2	0,0190	64	1,5141	6426	22,3886	96803	83,7825	361937
3	0,0203	68	0,5375	2237	2,6310	10481	44,0111	189389
4	0,0221	77	2,0099	8118	22,0313	95061	617,4606	2632971
5	0,0187	61	0,1093	451	21,0703	89544	628,0278	2708269
6	0,1388	587	1,3032	5588	16,8460	70330	349,6005	1499339
7	0,0296	109	0,0733	295	36,5264	156716	148,1565	602528
8	0,0504	191	1,1101	4678	9,4834	40495	18,1831	77138
9	0,0166	53	4,8711	20936	63,5891	276863	672,9833	2855663
10	0,0975	186	0,2864	1168	16,0286	68788	173,0817	743908
Média	0,0492	173	1,1953	5048	21,2207	91164	298,1754	1273779

Tabela 2: Comparação de tempos de validação de blocos com diferentes dificuldades. Fonte: Autoria própria.

A Tabela 2 traz os resultados das execuções de 10 testes para cada uma das 4 dificuldades (para dificuldade 5 ou superior os tempos de cálculo dos *hashes* tornaram-se muito elevados, impossibilitando que mais dados fossem coletados).

Os cálculos foram realizados por um MacBook Pro 2015, com processador Intel Core i5 (2,9 GHz) e 8GB de memória RAM (DDR3, 1867MHz). Para cada uma das amostras, foram registrados o nonce encontrado e o tempo para cumprir a dificuldade da validação.

Analisando as amostras dos tempos dos cálculos de *hash* presentes na Tabela 2, observa-se que o tempo de cálculo é diretamente proporcional ao nonce encontrado. Isso significa que, quanto maior o nonce do bloco, mais tempo leva para se calcular o *hash* (o que faz sentido, uma vez que o algoritmo incrementa em 1 o nonce à cada ciclo de cálculo de *hash*).

Vale a pena analisar também as médias dos tempos de cálculo para cada uma das dificuldades. À medida em que a dificuldade do bloco é aumentada, a média do tempo de cálculo para atingir o *hash*-alvo (e a variância entre as amostras) também aumenta.

Figura 13: Gráfico das médias dos tempos de validação pela dificuldade de mineração, em escala logarítmica. Fonte: Autoria própria.

A Figura 13 traz um gráfico com as médias dos tempos das validações (dispostas em escala logarítmica) versus a dificuldade de mineração. A reta demarcada se trata de uma linha de tendência de caráter exponencial.

É sabido que uma reta crescente, quando traçada em um gráfico de escala logarítmica, significa que esta assume um crescimento exponencial. Logo, a partir da observação do gráfico, é correto afirmar que à medida que a dificuldade da mineração aumentar, o tempo de validação crescerá exponencialmente.

4.3 TRANSPARÊNCIA DAS TRANSAÇÕES

O sistema é, na forma que foi implementado, totalmente transparente. A priori, qualquer pessoa pode acessar o livro de registros e ler todos os *idn* (no caso exemplificado por este trabalho, o CPF) de todas as transações que compõe qualquer um dos blocos da Blockchain.

Tal recurso, ao mesmo tempo que se mostra um ponto positivo ao possibilitar grande rastreabilidade dos fundos trafegados na rede, pode ser um ponto negativo se tratando da privacidade financeira dos usuários do sistema (já que qualquer pessoa poderia ter acesso a todos os fundos recebidos e enviados por qualquer usuário).

5 CONCLUSÃO

Por permitir a execução de transações permanentes, transparentes e seguras, a implantação de um sistema de criptomoeda de caráter centralizado, emitida por um banco central, seria substancialmente transformadora para o contexto econômico de um país.

Com a execução deste trabalho, foi desenvolvido um sistema de criptomoeda centralizado, capaz de receber transações inseridas por um usuário, validá-las e inseri-las em uma Blockchain. Apesar de ter cumprido o papel de validar a ideia de implementar um sistema de criptomoeda, outras ações são necessárias para que o sistema seja implantado em um cenário real.

Como trabalhos futuros, melhorias no sistema poderiam ser realizadas a fim de torná-lo mais robusto. Entre os pontos mais importantes de melhoria estão a necessidade da adaptação do sistema para receber transações concorrentemente, reformulação do mecanismo de cálculo de saldos hoje existente (para que o mesmo seja mais performático), criação de um sistema de login e substituição do mecanismo de armazenamento de contas e blocos (que hoje é feito em arquivos-texto). Uma vez que essas medidas fossem tomadas em trabalhos futuros, o sistema se tornaria mais completo e, por sua vez, melhor aplicado em um cenário real.

A ideia do lançamento de uma criptomoeda governamental é tão viável que alguns países (tais como Japão, Inglaterra, Suécia e Canadá) já estão em pesquisas avançadas rumo a implantação de sistemas de moedas virtuais (DORRIER, 2020), reforçando a relevância deste trabalho de conclusão de curso e confirmando a opinião de Gustavo Cunha, que afirma que “o debate deixou de ser se os bancos centrais deveriam digitalizar suas moedas, mas quando e como fazer isso” (CUNHA, 2020).

Levando em consideração a implementação e os resultados obtidos por meio do desenvolvimento deste trabalho conclui-se que, apesar de demandar adaptações para que necessidades específicas sejam atendidas, sua realização e implantação são viáveis. O princípio explicitado neste trabalho pode ser reproduzido não só em sistemas centralizados,

mas também em qualquer outro cenário onde um livro de registros imutável e transparente é aplicável.

REFERÊNCIAS

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

BEIKVERDI, A.; SONG, J. Trend of centralization in Bitcoin's distributed network. **2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2015 - Proceedings**, 2015.

CHALAEMWONGWAN, N. State of the Art and Challenges Facing Consensus Protocols on Blockchain. p. 957–962, 2018. ISSN 19767684.

COINMARKETCAP. **Bitcoin Historical Data**. 2020. Disponível em: <https://coinmarketcap.com/currencies/bitcoin/historical-data/>.

CUNHA, G. **Como funcionaria o Bitcoin do governo**. 2020. Disponível em: <https://www.infomoney.com.br/colunistas/blog-do-cunha/como-funcionaria-o-bitcoin-do-governo/>.

CVE. **CVE Details: The ultimate security vulnerability datasource**. 2018. Disponível em: <https://www.cvedetails.com/vulnerability-list/vendor_id-11993/product_id-22441/Dlitz-Pycrypto.html>.

DIGICONOMIST. **Bitcoin energy comsumption**. 2018. Disponível em: <https://digiconomist.net/bitcoin-energy-consumption>.

DORRIER, J. **Five Major Central Banks Unite to Explore Launching Their Own Digital Currencies**. 2020. Disponível em: <https://singularityhub.com/2020/01/23/five-of-the-worlds-biggest-central-banks-unite-to-research-digital-currencies/>.

DRESCHER, D. **Blockchain básico: Uma introdução não técnica em 25 passos**. 1. ed. [S.l.]: Novatec, 2018. 309 p.

Ethereum Community. **A next-generation smart contract and decentralized application platform**. [S.l.], 2013. Disponível em: <https://github.com/ethereum/wiki/wiki/White-Paper>.

GATTESCHI, V. et al. To Blockchain or Not to Blockchain : That Is the Question. n. April, p. 62–74, 2018.

IANSITI, M.; LAKHANI, K. R. The Truth About Blockchain. **Harvard Business Review**, 2017. Disponível em: <https://hbr.org/2017/01/the-truth-about-blockchain>.

JUDMAYER, A. et al. **Blocks and Chains: Introduction to Bitcoin, Cryptocurrencies, and their Consensus Mechanisms**. [S.l.]: Morgan & Claypool Publishers, 2017. ISBN 9781627057165.

KOÇ, A. K.; YAVUZ, E.; DALKıLıÇ, G. Towards Secure E - Voting Using Ethereum Blockchain. 2018.

KONING, J. P. **Fedcoin : A Central Bank-issued Cryptocurrency**. [S.l.], 2016. 1–37 p. Disponível em: <https://www.r3cev.com/s/R3-Report-Fedcoin.pdf>.

LEWIS, A. **Ripple Explained: Medieval Banking with a Digital Twist**. [S.l.], 2014. Disponível em: <https://www.coindesk.com/ripple-medieval-banking-digital-twist/>.

LI, W. et al. Towards Scalable and Private Industrial Blockchains. **Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts - BCC '17**, p. 9–14, 2017. Disponível em: <http://dl.acm.org/citation.cfm?doid=3055518.3055531>.

LO, S. K. et al. Evaluating Suitability of Applying Blockchain. **2017 22nd International Conference on Engineering of Complex Computer Systems (ICECCS)**, p. 158–161, 2017. Disponível em: <http://ieeexplore.ieee.org/document/8292816/>.

MAGRAHI, H.; SENOT, O. NFB : A Protocol for Notarizing Files over the Blockchain. p. 1–4, 2018.

MICROSOFT. **Visual Studio Code**. 2020. Disponível em: <https://code.visualstudio.com/>.

MIRZAYI, S.; MEHRZAD, M. Bitcoin, an SWOT analysis. **2017 7th International Conference on Computer and Knowledge Engineering (ICCKE)**, n. Iccke, p. 205–210, 2017. Disponível em: <http://ieeexplore.ieee.org/document/8167876/>.

NAKAMOTO, S. Bitcoin: A Peer-to-Peer Electronic Cash System. **Www.Bitcoin.Org**, p. 9, 2008. ISSN 09254560. Disponível em: <https://bitcoin.org/bitcoin.pdf>.

NAUMOFF, A. **Why Blockchain Needs 'Proof of Authority' Instead of 'Proof of Stake'**. 2017. Disponível em: <https://cointelegraph.com/news/why-blockchain-needs-proof-of-authority-instead-of-proof-of-stake>.

POA Network. **Proof of Authority: consensus model with Identity at Stake.** 2017. Disponível em: <https://medium.com/poa-network/proof-of-authority-consensus-model-with-identity-at-stake-d5bd15463256>.

PYCRYPTODOME. PyCryptodome's documentation. 2020. Disponível em: <https://pycryptodome.readthedocs.io/en/latest/>.

PYQT5. **PyQt5 5.14.2**. 2020. Disponível em: <https://pypi.org/project/PyQt5/>.

PYTHON.ORG. **Python Release 3.7.3**. 2019. Disponível em: <https://www.python.org/downloads/release/python-373/>.

RAIKWAR, M. et al. A Blockchain Framework for Insurance Processes. **2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)**, p. 1–4, 2018.

REIS, L. **tcc2-centralized-cryptocurrency**. 2020. Disponível em: <https://github.com/reisaolucas/tcc2-centralized-cryptocurrency>.

SANTOS, O. A. Impactos Econômicos Da Criptomoeda Bitcoin. **II Encontro Anual de Iniciação Científica da UNESPAR**, 2016.

SEO, J. D. **Simple Merkle Tree in Python**. 2017. Disponível em: <https://github.com/JaeDukSeo/Simple-Merkle-Tree-in-Python>.

TAPSCOTT, D.; TAPSCOTT, A. **Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World**. [S.l.: s.n.], 2016.

TASATANATTAKOOL, P. Blockchain : Challenges and Applications. p. 473–475, 2018.

TOSH, D. K. et al. Consensus protocols for blockchain-based data provenance: Challenges and opportunities. **2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)**, p. 469–474, 2017. Disponível em: <http://ieeexplore.ieee.org/document/8249088/>.

TSCHORSCH, F.; SCHEUERMANN, B. Bitcoin and beyond: A technical survey on decentralized digital currencies. **IEEE Communications Surveys and Tutorials**, v. 18, n. 3, p. 2084–2123, 2016. ISSN 1553877X.

VUJICIC, D.; JAGODIC, D.; RANDIC, S. Blockchain Technology , Bitcoin , and Ethereum : A Brief Overview. n. March, p. 21–23, 2018.