MARINHA DO BRASIL
DIRETORIA DE ENSINO DA MARINHA
CENTRO DE INSTRUÇÃO ALMIRANTE WANDENKOLK

CURSO DE APERFEIÇOAMENTO AVANÇADO EM
SEGURANÇA DA INFORMAÇÃO E COMUNICAÇÕES

PRIMEIRO-TENENTE (QC-CA) MARLON ANTUNES DO AMPARO

BLOCKCHAIN: uma abordagem sobre a integridade das informações

Rio de Janeiro
2020

PRIMEIRO-TENENTE (QC-CA) MARLON ANTUNES DO AMPARO

BLOCKCHAIN: BLOCKCHAIN: uma abordagem sobre a integridade das informações

Monografia apresentada ao Centro de Instrução Almirante Wandenkolk como requisito parcial à conclusão do Curso de Aperfeiçoamento Avançado em Segurança da Informação e Comunicações.

Orientador:
CC Emanuel Ferreira Jesus

CIAW
Rio de Janeiro
2020

Amparo, Marlon Antunes do.
Blockchain: uma abordagem sobre a integridade das informações / Marlon Antunes do Amparo. — Rio de Janeiro, 2020.
61f.

Orientador técnico e acadêmico: CC Emanuel Ferreira Jesus.

Monografia (Curso de Aperfeiçoamento Avançado de Segurança da Informação e Comunicações) — Centro de Instrução Almirante Wandenkolk. Centro de Pós-Graduação Avançada, Rio de Janeiro, 2020.

1. Blockchain. 2. Valor de hash. 3. Integridade. 4. Gerenciamento de posse. I. Centro de Instrução Almirante Wandenkolk. Centro de Pós-Graduação Avançada. II. Título.

PRIMEIRO-TENENTE (QC-CA) MARLON ANTUNES DO AMPARO

BLOCKCHAIN: BLOCKCHAIN: uma abordagem sobre a integridade das informações

Monografia apresentada ao Centro de Instrução Almirante Wandenkolk como requisito parcial à conclusão do Curso de Aperfeiçoamento Avançado em Segurança da Informação e Comunicações.

Aprovada em _____.

Banca Examinadora:

CMG (RM1-EN) Gian Karlo Huback Macedo de Almeida, CIAW

CC Emanuel Ferreira Jesus, CTIM

Carlos Vinício Rodríguez Ron, D.Sc., PUC-Rio

CIAW
Rio de Janeiro
2020

Dedico este trabalho a todos que contribuíram, diretamente e indiretamente, na produção deste trabalho.

AGRADECIMENTOS

Agradeço a Deus por me guiar durante mais esta jornada e por todas as batalhas conquistadas na vida, sem Ele nada é possível.

Agradeço aos meus pais, pelo esforço de me proporcionar uma educação de qualidade e por todo incentivo e força que me deram no decorrer desta trajetória.

Agradeço a minha noiva Yasmin pela paciência, apoio nos momentos difíceis e todo suporte, tanto durante o curso, quanto no decorrer do desenvolvimento deste trabalho.

Agradeço ao meu orientador CC Emanuel Ferreira Jesus pela paciência na orientação e incentivo que tornaram possível a conclusão deste trabalho.

Agradeço ao Prof. Dr. Carlos Vinício Rodríguez Ron, por seus ensinamentos, paciência e motivação. É um prazer tê-lo na banca examinadora.

E por fim, ao coordenador do Curso de Aperfeiçoamento Avançado em Segurança da Informação e Comunicações, CMG (RM1-EN) Gian Karlo Huback Macedo de Almeida, pelo apoio e atenção concedidos ao longo de todo o curso, contribuindo de forma significativa para a formação profissional da turma de SIC.

“A menos que modifiquemos nossa maneira de pensar, não seremos capazes de resolver os problemas causados pela forma como nos acostumamos a ver o mundo”.

(Albert Einstein)

BLOCKCHAIN: uma abordagem sobre a integridade das informações

RESUMO

Blockchain é uma estrutura de dados na qual suas informações são agrupadas em blocos, que ligados ao seu anterior, formam uma cadeia de blocos. Essas cadeias não requerem nenhuma autoridade central ou terceira, pois são os próprios usuários que atuam como intermediários entre si para realizar todos os tipos de transações. O Blockchain utiliza diversos conceitos e princípios de ciência da computação e engenharia de software, como arquiteturas de rede, funções e referências de hash, criptografia, estruturas e repositório de dados, comunicação entre computadores em rede e quebra-cabeças computacionais. Embora seu uso mais comum seja o setor econômico, o Blockchain pode ser extrapolado para outros setores, como marketing e logística. O Blockchain resolve o problema dos gastos duplos através da verificação de cada transação, por meio do algoritmo de validação conhecido como prova de trabalho, realizado pelos chamados "mineradores", responsáveis por auditar e confirmar todas as transações realizadas. Em razão da prova de trabalho, adicionar um bloco novo é custoso do ponto de vista de processamento, e torna as tentativas de manipular o histórico de transações mais custosas ainda. Assim, o volume de esforço computacional agregado investido na criação de um histórico de transações parece um critério natural para selecionar um histórico no caso de haver mais de uma versão conflitante. Se todos os nós do sistema aplicarem o mesmo critério, em algum momento todos eles concordarão com uma versão idêntica do histórico.

Palavras- chave: Blockchain; valor de hash; integridade e gerenciamento de posse.

LISTA DE FIGURAS

LISTA DE QUADROS

SUMÁRIO

1. INTRODUÇÃO

1.1 Contextualização

O Blockchain tem recebido muita atenção na discussão pública e na mídia. Alguns admiradores argumentam que ele é a maior invenção desde o aparecimento da internet. Apesar de hoje a aplicação do blockchain estar se dissociando do Bitcoin, essa tecnologia começou junto com a criptomoeda. O conceito do primeiro blockchain público nasceu em 2008, no artigo acadêmico Bitcoin: um sistema financeiro eletrônico ponto a ponto, publicado por uma pessoa ou grupo sob o pseudônimo de Satoshi Nakamoto (suposto criador do Bitcoin). Criado em um cenário de crise econômica mundial e bolha imobiliária, a tecnologia por trás do Bitcoin nasceu para, entre outras coisas, prevenir o gasto duplo dos valores e aumentar a confiança das transações financeiras, levando-as para a internet. [25]

Um entendimento conceitual das bases técnicas faz-se necessário para compreender aplicações específicas do Blockchain, que foi construído tendo em mente três características: descentralização, segurança e imutabilidade de transações. As informações são armazenadas em estruturas de dados chamadas de blocos, onde cada bloco possui em seu cabeçalho o resumo criptográfico do bloco anterior. Estes resumos são usados para ligar os blocos e servem de garantia de que as informações armazenadas não foram violadas. O nome Blockchain - ou cadeia de blocos - deriva desta ligação entre os blocos. O uso de valores de hash criptográfico, além de tornar as transações seguras, dificultam ataques que visem alterar os blocos, pois para conseguir alterar um bloco é necessário um grande poder computacional., além de ser preciso alterar o bloco em questão e todos os subsequentes. [12]

1.2 O Problema

O problema em análise neste trabalho é analisar a efetividade do Blockchain para garantir a integridade em um sistema ponto a ponto distribuído, constituído de um número desconhecido de participantes, com nível de confiabilidade desconhecido.

A maioria dos usuários não pensa na integridade dos sistemas de software, considerando-os como existentes, pois na maioria das vezes, se interage com sistemas que mantêm a sua integridade. Contudo, caso um sistema apresente uma falha, ocasionando a perda de dados ou que pessoas estranhas foram capazes de acessar os seus dados, começa-se a perceber a importância da integridade do software.

A integridade é um importante aspecto de qualquer sistema de software, possuindo três componentes principais [6]:

a) Integridade dos dados: os dados usados e mantidos pelo sistema são completos, corretos e livres de contradições;

b) Integridade comportamental: o sistema se comporta conforme esperado e está livre de erros de lógica; e

c) Segurança: o sistema é capaz de restringir o acesso aos seus dados e às suas funcionalidades somente aos usuários autorizados.

1.3 Justificativa

Observa-se no cenário atual uma crescente utilização do blockchain nos sistemas de informação, mudando os paradigmas da segurança da informação. No entanto, note-se pouco conteúdo sobre esse tema na Marinha e o desconhecimento da maioria dos militares sobre o assunto, que é considerado por muitos a maior revolução tecnológica desde o surgimento da internet [24].

O blockchain está ampliando as possibilidades para a segurança de dados e tem atraído cada vez mais empresas interessadas em garantir maior segurança nas informações para seus clientes. Seu potencial econômico é de reduzir até US$ 12 bilhões de gastos, por ano, em custos de infraestrutura e acesso à transparência nos dados [11].

E, para acrescentar, a aplicação do blockchain promoverá a descentralização do armazenamento dos dados, que hoje ainda são centrados em servidores das grandes empresas. As informações rodarão, dessa forma, em um ambiente completamente distribuído. Assim, as grandes corporações não deverão controlar os dados dos usuários. Por outro lado, o Blockchain vai propiciar uma dificuldade maior para os hackers ou exploradores invadirem a rede [10].

Portanto, é possível perceber que a adoção do Blockchain não será só importante do ponto de vista da segurança, como também irá trazer de volta a privacidade para o usuário, afinal, este poderá se preservar ou compartilhar as informações apenas para quem quiser.

1.4 Objetivos

Nesta seção serão apresentados os objetivos que guiaram a produção deste trabalho. Inicialmente, será apresentado o objetivo geral, que representa o que se quer obter após o

estudo em questão. Em seguida, serão apresentados os objetivos específicos, que foram as etapas seguidas para alcançar o objetivo principal deste trabalho

A definição dos objetivos determina o que o pesquisador quer atingir com a realização do trabalho de pesquisa. Objetivo é sinônimo de meta, fim. Os objetivos podem ser separados em Objetivos Gerais e Objetivos Específicos [15].

1.4.1 Objetivo Geral

O propósito deste trabalho é analisar a tecnologia Blockchain nos aspectos de sua estrutura, funcionamento e principais conceitos, mostrando que essa tecnologia garante a integridade da informação em um sistema ponto a ponto distribuído.

1.4.2 Objetivos Específicos

Para alcançar o objetivo principal deste trabalho, serão adotas as seguintes medidas:

a) Revisão da literatura existente sobre o Blockchain;

b) Apresentação dos conceitos elementares por trás da tecnologia blockchain;

c) Analise das limitações e vulnerabilidades do Blockchain; e

d) Apresentação de uma visão geral a respeito das aplicações do Blockchain que já foram implementadas e que estão sendo usadas atualmente;

1.5 Metodologia

Será apresentada, nesta seção, a metodologia utilizada para a realização da pesquisa deste trabalho, dando foco em suas classificações e em suas limitações.

A Metodologia consiste em estudar, compreender e avaliar os vários métodos disponíveis para a realização de uma pesquisa acadêmica. Ela é a aplicação de procedimentos e técnicas que devem ser observados para construção do conhecimento, com o propósito de comprovar sua validade e utilidade nos diversos âmbitos da sociedade [20]

A metodologia tem como objetivo mostrar ao pesquisador como conduzir uma pesquisa. Ela o ajuda a refletir e o motiva a criar um novo olhar sobre o mundo: um olhar composto de curiosidade, indagação e criatividade [22]

1.5.1 Classificação da Pesquisa

A importância de se classificar uma pesquisa está na necessidade de definição dos instrumentos e procedimentos que o pesquisador deve adotar para o planejamento de sua investigação. A seguir, serão apresentados os enquadramentos da pesquisa realizada neste trabalho [15].

1.5.1.1 Quanto aos fins

A pesquisa aqui proposta, quanto aos seus objetivos ou fins, é explicativa, já que ela visa identificar os fatores que determinam ou contribuem para a ocorrência dos fenômenos. [22].

1.5.1.2 Quanto aos meios

Quanto aos meios este trabalho enquadra-se como uma pesquisa bibliográfica, já que serão utilizadas literaturas gerais, como livros, monografias, artigos e materiais publicados na internet, que tratam sobre os conceitos de Blockchain. [22].

1.5.2 Limitações do Método

Este trabalho se limitará a fazer uma pesquisa bibliográfica, restringindo-se a analisar a tecnologia blockchain nos aspectos de sua estrutura, funcionamento e principais conceitos. Não será proposta uma aplicação de âmbito militar, visto que não faz parte do escopo deste trabalho.

2. CONCEITOS ELEMENTARES

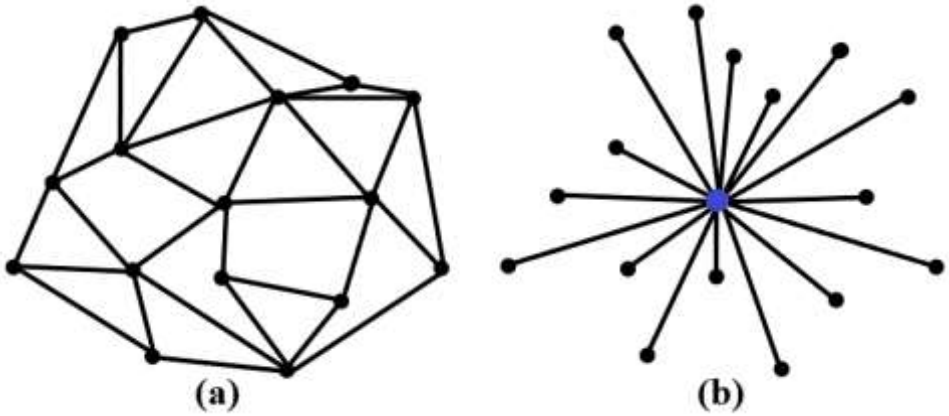
2.1 Tipos de arquitetura

Ao implementar um sistema, uma das decisões fundamentais diz respeito à sua arquitetura, ou seja, o modo como seus componentes se relacionam e são organizado. As duas principais abordagens para a arquitetura de sistemas de software são: centralizada e distribuída [23].

Em sistemas de software centralizados, os componentes estão localizados em torno de um componente central e estão conectados a ele. Em contraste, os componentes de sistemas distribuídos formam uma rede de componentes conectados, sem que haja nenhum elemento central para a coordenação ou controle.

A Figura 1 mostra essas duas arquiteturas. Os pontos representam os componentes do sistema, também chamados de nós, e as linhas representam as conexões entre eles. Observando a arquitetura distribuída na Figura 1 (a), é importante perceber os componentes estão conectados uns aos outros sem que haja um elemento central, todos eles são conectados uns aos outros, pelo menos indiretamente.

Figura 1 - Arquiteturas de rede distribuída e centralizada



Fonte: elaborado pelo autor

O lado direito, na Figura 1 (b), mostra uma arquitetura centralizada, no qual cada componente está conectado a um elemento central. Os componentes não estão conectados diretamente uns aos outros, eles têm apenas uma conexão direta com o componente central.

2.1.1 Vantagens dos sistemas distribuídos

As principais vantagens de um sistema distribuído em comparação aos computadores individuais são [23]:

a) Mais capacidade de processamento: a capacidade de processamento de um sistema distribuído é o resultado da combinação da capacidade de processamento de todos os computadores conectados. Desse modo, os sistemas distribuídos, em geral, têm mais capacidade de processamento que cada computador individual;

b) Redução de custos: como os sistemas distribuídos são constituídos de muitos computadores, os custos iniciais desses sistemas são mais elevados que o custo de um supercomputador. Entretanto, os custos para instalar, manter e operar um supercomputador ainda são mais altos;

c) Mais confiabilidade: um sistema distribuído não tem um ponto único de falha. Se um elemento falhar, os elementos restantes poderão assumir a sua função, assim o sistema pode continuar funcionando, mesmo quando alguma máquina individual falhar; e

d) Capacidade de se expandir naturalmente: a capacidade de processamento de um sistema distribuído é o resultado da capacidade de processamento agregada de seus nós. Pode-se aumentar a capacidade de processamento do sistema conectando computadores adicionais a ele ou então substituir os computadores por outros mais modernos e com maiores recursos (memória, processamento, armazenamento...).

2.1.2 Desvantagens dos sistemas distribuídos

As principais desvantagens de um sistema distribuído em comparação aos computadores individuais são [23]:

a) Overhead de coordenação: os sistemas distribuídos não têm entidades centrais para coordenar seus membros. Assim, a coordenação deve ser feita pelos próprios nós do sistema. O trabalho de coordenação entre as unidades é um desafio, e exige esforços e capacidade de processamento;

b) Overhead de comunicação: a coordenação exige comunicação. Portanto, os computadores que compõe um sistema distribuído precisam se comunicar uns com os outros. Isso exige um protocolo de comunicação, além de envio, recepção e processamento de mensagens, que por sua vez, demandam esforços e capacidade de processamento;

c) Dependência de rede: os computadores em sistemas distribuídos se comunicam por meio de mensagens passadas por uma rede, que têm seus próprios desafios e

adversidades. No entanto, sem rede não haveria sistema distribuído nem comunicação e, portanto, nenhuma coordenação entre os nós; e

d) Problemas de segurança: enviar informações por uma rede implica preocupações com segurança, pois entidades não confiáveis poderão utilizar indevidamente a rede a fim de acessar e explorar informações. Quanto menos restrito for o acesso à rede pela qual os nós distribuídos se comunicam, maiores serão as preocupações com a segurança em um sistema distribuído.

2.2 Sistemas ponto a ponto

As redes ponto a ponto (ou P2P, do termo em inglês *peer-to-peer*), um tipo de sistema distribuído, são constituídas de computadores individuais (nós), que disponibilizam diretamente seus recursos computacionais aos demais membros da rede, sem que haja um ponto central de coordenação. Os nós da rede são iguais no que diz respeito aos seus direitos e funções no sistema. Além disso, todos eles são tanto consumidores quanto fornecedores de recursos [23].

Os sistemas ponto a ponto têm aplicações interessantes, o compartilhamento de arquivos, a distribuição de conteúdo e a proteção de privacidade são alguns exemplos. A maior parte dessas aplicações utiliza uma ideia simples, porém eficaz: transformar os computadores dos usuários nos nós que compõem o sistema distribuído como um todo [8]. Como resultado, quanto mais usuários ou clientes usarem o software, maior e mais potente se tornará o sistema.

Todo mercado que atue, principalmente, como intermediário entre produtores e consumidores de bens e serviços imateriais ou digitais pode ser substituído por um sistema ponto a ponto, exemplo disso é o mercado financeiro. O ato de fazer ou receber empréstimos ou transferir dinheiro de uma conta para outra são apenas uma transferência de um bem imaterial, administrado por intermediários. Fazer uma transferência monetária de uma conta bancária para outra em um país diferente, por exemplo, envolve até cinco intermediários, que precisam de tempo de processamento e impõem as próprias taxas. Em um sistema ponto a ponto, a mesma transferência seria muito mais simples e exigiria menos tempo, além de ter um custo menor.

À medida que a digitalização continuar, mais e mais itens da vida cotidiana e um volume cada vez maior de bens e serviços se tornará imaterial e se beneficiará da eficiência

dos sistemas ponto a ponto: pagamentos, empréstimos, seguros, emissão e validação de certidões de nascimento, contratos, entre outros.

O sistema ponto a ponto deve ter integridade para atender às expectativas dos usuários e reforçar a sua confiança. Se a confiança dos usuários não for reforçada pelo sistema em decorrência da falta de integridade, eles o abandonarão, resultando no fim do sistema. Por questões de simplicidade, duas ameaças podem ser consideradas como principais à integridade dos sistemas P2P [8]:

a) Falhas técnicas: sistemas ponto a ponto são constituídos pelos computadores individuais de seus usuários, que se comunicam por meio de uma rede. Todos os componentes de hardware e software de um computador, assim como qualquer componente de uma rede de computadores, têm o risco de falhar ou gerar erros.

b) Participantes maliciosos: participantes desonestos e maliciosos constituem a ameaça mais severa a um sistema *peer-to-peer*, pois atacam a base sobre o qual qualquer sistema desse tipo se sustenta: a confiança.

2.3 Hashing de dados

O *hashing* refere-se ao processo de geração de uma saída de tamanho fixo a partir de uma entrada de tamanho variável. Isto é feito através do uso de fórmulas matemáticas conhecidas como funções *hash* [14]. Há muitas funções de *hash* diferentes no que concerne ao tamanho do valor de *hash* que geram, embora nem todas envolvam o uso de criptografia, as chamadas funções *hash* criptográficas são componentes fundamentais no estudo do Blockchain.

Uma função de *hash* criptográfico, também conhecida apenas como *hash*, é um algoritmo matemático que transforma qualquer bloco de dados em uma série de caracteres de comprimento fixo. Independentemente do comprimento dos dados de entrada, a saída será sempre um valor de *hash* do mesmo comprimento [5]. Vale ressaltar que os valores de *hash* podem ter zeros na frente de modo a compor o tamanho necessário.

As funções de *hash* criptográfico criam “impressões digitais” para qualquer tipo de dado e têm as seguintes propriedades [21]:

a) São determinísticas: ser determinística significa que a função de *hash* produz valores de *hash* idênticos para dados de entrada idênticos.

b) São pseudoaleatórias: ser pseudoaleatória significa que o valor de *hash* devolvido por uma função de *hash* muda de forma imprevisível quando os dados de entrada são

alterados. Mesmo que os dados de entrada mudem apenas um pouco, o valor de *hash* será diferente de forma imprevisível. Assim, prever o valor de *hash* com base nos dados de entrada não deve ser possível.

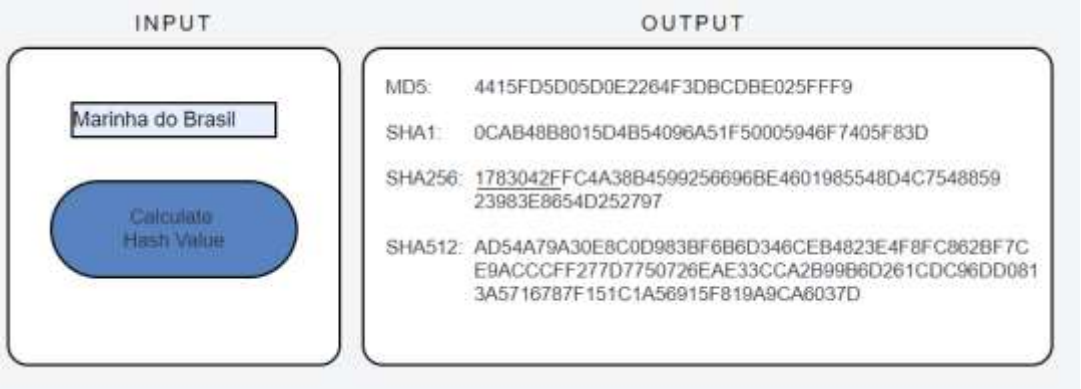
c) São funções unidirecionais: uma função unidirecional não possibilita que seus valores de entrada sejam rastreados com base nas saídas. Deste modo, ser uma função unidirecional significa que ela não pode ser usada de modo inverso. Isso quer dizer que os valores de *hash* não informam nada sobre o conteúdo dos dados de entrada. Funções unidirecionais também são chamadas de irreversíveis.

d) São resistentes à colisão: uma função de *hash* é resistente à colisão se for muito difícil encontrar duas ou mais porções distintas de dados para as quais ela gere valores idênticos de *hash*. Uma colisão de *hash* é o equivalente de ter duas pessoas com impressões digitais idênticas.

2.3.1 Gerando valores de hash

Este trabalho usou uma ferramenta[1] para criar valores de *hash* para um dado textual simples, dessa forma, a compreensão deste conceito poderá ser atingida de modo mais fácil. A figura a seguir mostra os valores de *hash*, para cada função de *hash*, referente à entrada de dados “Marinha do Brasil”.

Figura 2 - Valores de hash referente à Marinha do Brasil



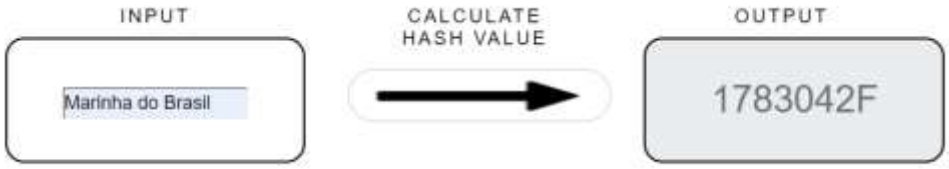
Fonte: elaborado pelo autor

Pode-se observar que os valores de hash diferem por causa dos diferentes detalhes de implementação das funções de *hash* que os geram. Normalmente, valores de *hash*

[1] Ferramenta disponível em: http://www.blockchain-basics.com/HashFunctions.html

criptográficos são bem longos, e desse modo, para o olho humano, são difíceis de ler ou comparar. Visto isso, este trabalho utilizou uma versão abreviada do valor de *hash* criptográfico SHA256[2] para seja possível comparar diferentes maneiras de gerar dados de hashing.

Figura 3 - Hash abreviado referente à Marinha do Brasil



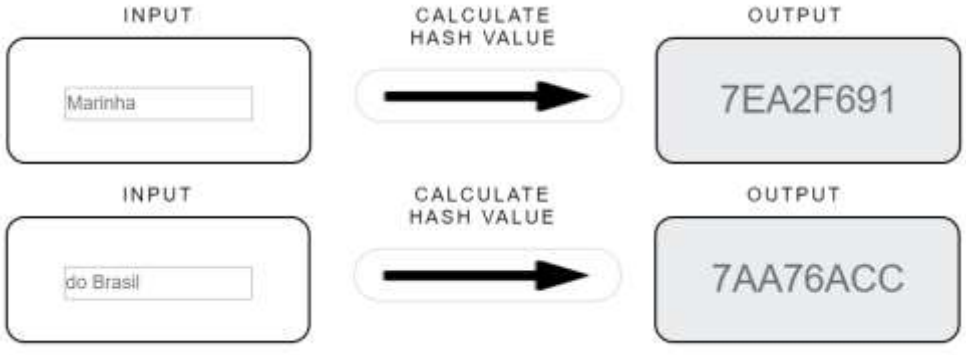
Fonte: elaborado pelo autor

2.3.2 Padrões para hashing de dados

As funções de *hash* só aceitam uma porção de dados em um determinado instante, ou seja, não há nenhuma função de *hash* que aceite um conjunto de dados independentes de uma só vez. Contudo, com frequência é necessário um único valor de *hash* para uma coleção grande de dados, como na estrutura de dados blockchain, por exemplo. A solução é utilizar um dos padrões a seguir na aplicação de funções de *hash* aos dados [8]:

a) *Hashing* independente: *hashing* independente significa aplicar a função de *hash* em cada porção de dados independentemente. A Figura 4, vista a seguir, mostra esse conceito calculando o valor de *hash* abreviado para duas palavras distintas separadamente.

Figura 4 - Hashing de dados independente



Fonte: elaborado pelo autor

Como é possível ver na Figura 4, palavras diferentes produzem valores de *hash* distintos.

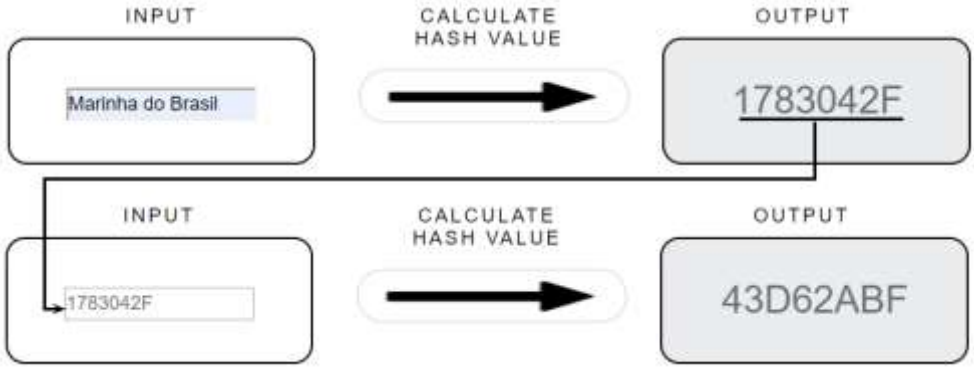
[2] Disponível em: http://www.blockchain-basics.com/Hashing.html

b) *Hashing* combinado: o objetivo do *hash* combinado é obter um único valor de *hash* para mais de uma porção de dados em uma só tentativa. Combinar todas as porções de dados independentes em uma única porção e calcular o seu valor de *hash* depois é a estratégia para isso. Essa abordagem é útil caso se tenha interesse de criar um único valor de *hash* para um conjunto de dados em um determinado instante.

A Figura 3 é um exemplo do conceito de *hashing* combinado. As palavras individuais são inicialmente combinadas em uma só frase, com um espaço em branco entre elas, e o *hash* da frase resultante é gerado em seguida.

c) *Hashing* repedido: *hashing* repedido é a aplicação de uma função de *hash* à sua própria saída. A Figura 5, mostra o conceito calculando o valor de *hash* abreviado repetidamente.

Figura 5 - Hashing de dados repetido

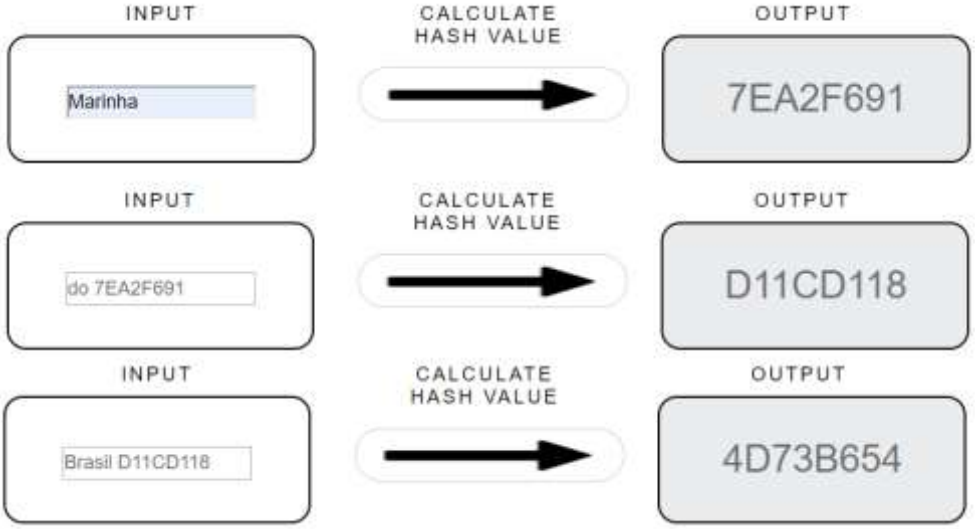


Fonte: elaborado pelo autor

O texto “Marinha do Brasil” produz o valor de *hash* abreviado 1783042F, que produz, por sua vez o valor de *hash* abreviado 43D62ABF.

d) *Hashing* sequencial: o objetivo do *hashing* sequencial é a atualização incremental de um valor de *hash* à medida que novos dados chegarem. Realiza-se isso utilizando um *hashing* combinado e repetido ao mesmo tempo. O valor de *hash* de um dado é combinado com os novos dados e, em seguida, é passado pela função de *hash* a fim de que se obtenha um valor de *hash* atualizado.

Figura 6 - Hashing de dados sequencial

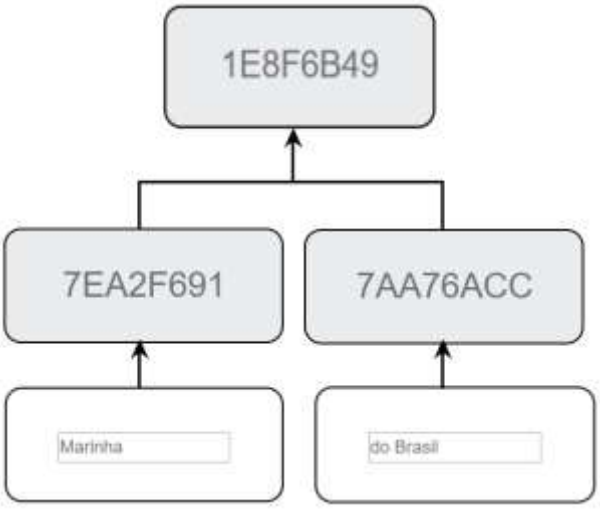


Fonte: elaborado pelo autor

A Figura 6 mostra o conceito de *hashing* sequencial, que começa com o *hashing* da palavra “Marinha”, que produz o valor de *hash* abreviado 7EA2F691. Ao chegar um novo dado (“do”), ele é combinado com o valor de *hash* existente e fornecido como entrada para uma função de *hash*. Por fim, a união desse *hash* abreviado é combinado com “Brasil”, formando então o *hash* abreviado 4D73B654.

e) *Hashing* hierárquico: *hashing* hierárquico é a aplicação de *hashing* combinado em um par de valores de *hash*, formando uma pequena hierarquia de valores de *hash* com um único valor no topo, como pode ser visto na Figura 7, a seguir.

Figura 7 - Hashing de dados hierárquico



Fonte: elaborado pelo autor

O *hashing* hierárquico é mais eficaz que o *hashing* combinado pois combina valores de *hash*, que possuem tamanho fixo, em vez de combinar os dados originais, que poderiam ter qualquer tamanho.

2.3.3 Casos de aplicação do valor de hash

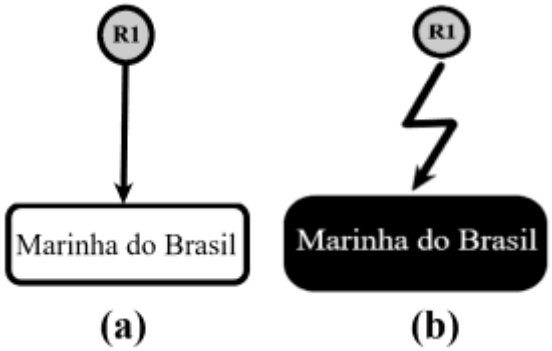
Um caso simples de aplicação de valores de *hash* é utilizá-lo para realizar comparações. Em vez de comparar explicitamente o conteúdo dos dados parte a parte, compara-se seus valores de *hash* criptográficos. Se os valores forem diferentes, os dados em consideração também serão distintos. Caso os valores de *hash* criptográficos forem idênticos, seus dados de entrada correspondentes também serão idênticos [26].

A mesma ideia pode ser aplicada caso se deseje detectar mudanças nos dados (alteração dos dados). Compara-se o valor de *hash* criptográfico dos dados criado anteriormente, com um valor de *hash* criptográfico recém criado para os mesmos dados. Se os valores de *hash* forem idênticos, os dados não foram alterados.

Um caso de aplicação um pouco mais sofisticado é referenciar dados armazenados em um local (um disco rígido ou um banco de dados) e garantir que eles permaneçam inalterados. A ideia consiste em combinar o valor de *hash* criptográfico dos dados armazenados com informações sobre o local em que estão. Se esses dados forem alterados, as duas informações deixarão de ser consistentes e, portanto, a referência de hash se tornará inválida [8]. Essas referências podem ser entendidas como um tíquete de um guarda-volumes que mostra qual o local de um item dentro do guarda volume, ou seja, os programas de computador utilizam referencias para “lembrar” do lugar em que os dados foram armazenados e recuperá-los posteriormente.

A Figura 8 (a), vista a seguir, mostra o funcionamento das referências de *hash* de modo esquemático, apresentando uma referência de *hash* válida. O círculo cinza com rótulo R1 representa uma referência desse tipo. A caixa branca representa alguns dados que devem permanecer inalterados. A seta do círculo para a caixa representa o funcionamento da referência de *hash*. Ela aponta da referência para o dado ao qual se refere.

Figura 8 - Referência de hash válida a esquerda e inválida a direita



Fonte: adaptado de [8]

Já a Figura 8 (b), mostra a representação de uma referência de *hash* quebrada ou inválida. A caixa preta representa dados que foram alterados após a referência ter sido criada. O círculo cinza continua representando a referência de *hash* originalmente criada. A seta serrilhada que aponta do círculo para a caixa alterada enfatiza o fato de a referência de *hash* R1 ser inválida e não permitir mais acessos para recuperar os dados, pois estes foram modificados.

Valores de *hash* também podem ser usados para criar "quebra-cabeças" que exijam recursos computacionais para serem resolvidos. Não é possível resolvê-los com base em conhecimento, em dados armazenados ou por meio de raciocínio, a única forma de solucioná-los é por capacidade de processamento [8].

Os elementos que compõem um quebra-cabeça de *hash* são [8]:

a) Dados especificados que devem ser mantidos inalterados, por conveniência, chamados apenas de dados;

b) Dados que podem ser livremente alterados, chamados de *nonce*[3];

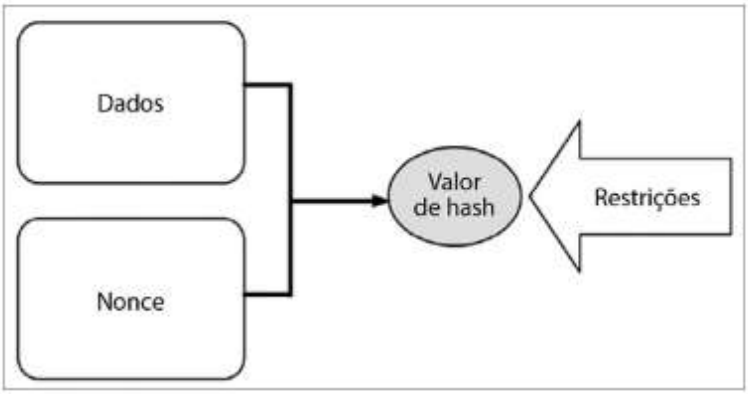
c) Função de *hash* a ser aplicada; e

d) Restrição no valor de *hash* do *hashing* combinado, também chamado de nível de dificuldade.

A Figura 9, vista a seguir, mostra a configuração de um quebra-cabeça de *hash*. Um *hashing* combinado é aplicado aos dados e ao *nonce*. O valor de *hash* resultante deve atender às restrições especificadas.

[3] É designado de *nonce* ou *number that can only be used once* (número que só se pode usar uma vez), um número arbitrário usado na criptografia. Numa rede blockchain baseada em prova de trabalho, o *nonce* funciona numa combinação com o *hash*, como elemento de controle para evitar a manipulação das informações dos blocos. Fonte [3]

Figura 9 - Elementos de um quebra-cabeça de hash



Fonte: [8]

Vale ressaltar que os quebra-cabeças só podem ser resolvidos por tentativa erro. Isso exige adivinhar um *nonce*, calcular o valor de *hash* dos dados combinados com a função de *hash* necessária e avaliar o valor de *hash* resultante com base nas restrições. Se o valor de *hash* satisfizer as restrições, o quebra-cabeça é resolvido, caso contrário, utiliza-se outro *nonce*, até resolver o quebra-cabeça. O *nonce* que, combinado com os dados especificados, produzir um valor de *hash* que satisfaça as restrições é chamado de solução [8]. No Quadro 1, visto a seguir, pode-se observar um exemplo de um quebra-cabeça de hash[4].

Cabe frisar que no contexto do Blockchain, os quebra-cabeças de *hash* são chamados de prova de trabalho (ou PoW, do termo em inglês *Proof of Work*), pois sua solução prova que alguém fez o trabalho necessário para resolvê-lo [8].

Na explicação de *hashing* repedido, pôde-se notar que o texto “Marinha do Brasil” produz o valor de *hash* abreviado 1783042F. Mas qual dado combinado com Marinha do Brasil produziria um valor de *hash* abreviado iniciado com dois zeros? Este é o quebra-cabeça de *hash*, ou seja, encontrar o nonce que combinado com Marinha do Brasil produza um valor de hash abreviado iniciado com dois.

[4] Para a montagem do Quadro 1 foi utilizada a ferramenta disposta no site: http://www.blockchain-basics.com/HashPuzzle.html.

Quadro 1 - Nonces para resolver um quebra-cabeça de hash

Nonce	Texto	Saída
0	Marinha do Brasil 0	9C9E6798
1	Marinha do Brasil 1	9373E936
2	Marinha do Brasil 2	9B752779
3	Marinha do Brasil 3	A2B31846
4	Marinha do Brasil 4	ACE25D00
...	...	...
345	Marinha do Brasil 345	00AD4BD1
346	Marinha do Brasil 346	15A02E60
347	Marinha do Brasil 347	6C15E6AB

Fonte: elaborado pelo autor

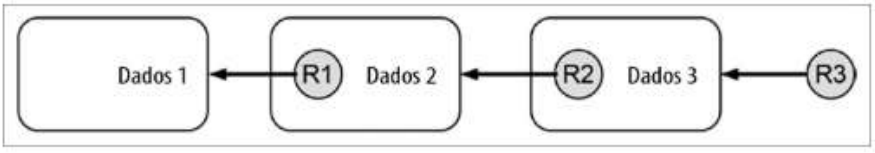
Exigir que o valor de *hash* atenda a uma determinada restrição é a parte essencial do quebra-cabeça de *hash*. Essas restrições são chamadas de dificuldade ou nível de dificuldade, são expressas como um número natural e referem-se ao número de zeros na frente que o valor de *hash* deve ter. Quanto maior o nível de dificuldade, mais zeros na frente são necessários, mais complicado o quebra-cabeça e mais capacidade de processamento e tempo serão necessários para resolvê-lo.

2.3.4 Padrões de armazenamento de referências de hash

Há dois padrões clássicos de uso de referências de *hash* para armazenar dados de modo sensível a mudanças: cadeia e árvore [8].

Uma cadeia de dados ligados, também chamada de lista ligada [7], é formada quando cada porção de dados também contém uma referência de *hash* para outra porção de dados. A criação de uma cadeia, como mostra a Figura 10, é feita inicialmente com uma porção de dados cujo rótulo é “Dados 1” e a criação da referência de *hash* “R1”. Quando novos dados chegam, eles são unidos à referência de hash que aponta para “Dados 1”. R2 refere-se aos dados que acabaram de chegar e à referência de hash R1. A referência “R3” é contém toda informação necessária para acessar todos os dados da cadeia na ordem inversa, ela também é chamada de cabeça de cadeia.

Figura 10 - Dados em cadeia

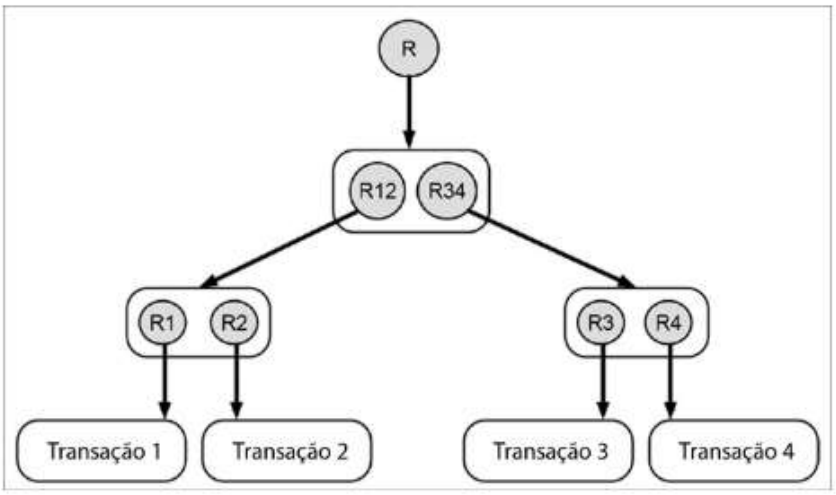


Fonte: [8]

Já a estrutura em árvore, que também é conhecida por árvore de Merkle[5], relaciona todas as transações e as agrupa em pares para obter um *hash* raiz ou "endereço mestre" que é a base árvore [2], ou seja, as porções de dados são agrupadas na medida que estejam disponíveis, deixando-as acessíveis por meio de uma única referência de *hash*.

A criação da árvore de Merkle, mostrada na Figura 11, é realizada da seguinte forma: inicialmente, as referências de *hash* para os dados de transação são criadas (R1 a R4), que são agrupadas em pares, criando-se as referências R12 e R34. Esse procedimento é repetido até obter-se uma única referência de *hash*, também chamada de raiz da árvore (R).

Figura 11 - Árvore de Merkle



Fonte: [8]

2.4 Segurança

No Blockchain, três conceitos relacionados à segurança, são extremamente importantes [28], são eles:

a) Identificação: a identificação significa afirmar ser alguém apresentando um nome ou outra informação que seja usada como um identificador. A identificação não prova que o indivíduo que apresentou a informação é realmente quem afirma ser;

b) Autenticação: o propósito da autenticação é evitar que alguém afirme ser outra pessoa. Significa verificar ou provar que o indivíduo é realmente quem afirma ser. É importante que a prova da identidade esteja associada unicamente ao indivíduo (uma foto do rosto, impressão digital ou outra informação que o identifique unicamente); e

[5] Ralph C. Merkle é um dos pesquisadores que desenvolveu a criptografia de chave pública e a estrutura das árvores Merkle (patenteada em 1979) com base no projeto "Merkle Puzzles" [2].

c) Autorização: a autorização significa conceder acesso a recursos ou serviços específicos em virtude das características ou propriedades da identidade de alguém. É consequência tanto de uma autenticação bem sucedida quanto de uma avaliação das características ou dos direitos de uma pessoa.

Além desses conceitos, o Blockchain utiliza a criptografia assimétrica para alcançar dois objetivos: identificar contas e autorizar transações [8]. No que se refere a identificar contas, o Blockchain precisa identificar usuários ou contas de usuários para manter o mapeamento entre proprietários e propriedades. Os números das contas no Blockchain são chaves públicas, que são usadas para identificar as contas envolvidas na transferência de posse. Se tratando de autorizar transações, os dados de transação sempre devem incluir informações que sirvam como prova de que o proprietário que está cedendo a posse realmente concorda com a transferência descrita. O fluxo de informações que esse acordo implica começa no proprietário da conta que está cedendo a posse, e deve alcançar todos que inspecionam os dados da transação. O proprietário da conta que transfere a posse cria um texto cifrado com sua chave privada. Todos os demais podem conferir essa prova de concordância usando a chave de criptografia pública, que é o número da conta que está cedendo a posse [8].

O Blockchain deve garantir que somente o proprietário de um bem possa transferi-lo para outras contas e nesse ponto que o conceito de autorização deve ser abordado. A principal ideia de garantir que somente o proprietário transfira a posse consiste em utilizar uma medida de segurança digital equivalente às assinaturas à mão, que sirvam para identificar uma conta, declarar a concordância do proprietário com o conteúdo dos dados da transação e aprovar a sua execução, permitindo que os dados sejam adicionados ao histórico de dados de transação [8].

As assinaturas digitais utilizam *hashing* criptográfico e um fluxo de informações de chave privada. Os dois principais elementos das assinaturas digitais são: criação de uma assinatura e verificação dos dados usando a assinatura.

3. BLOCKCHAIN ATRAVÉS DE ANALOGIAS

3.1 Livro-Razão, gerenciamento de posse e o Blockchain

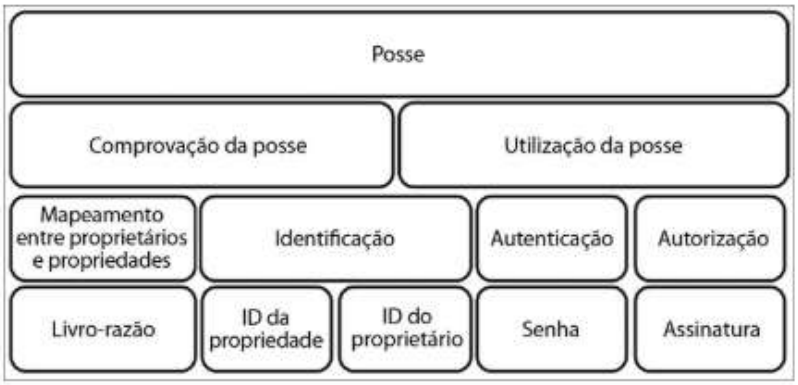
Atualmente, pode-se identificar as pessoas através de diversas formas: carteiras de habilitação ou identidade, certidão de nascimento, passaporte entre outros. Para identificar objetos utiliza-se números de série, datas, e certificados de produção ou descrições detalhadas no produto. Esses documentos, após criados, não podem sofrer alterações. Caso uma pessoa mude de nome, por exemplo, outro documento terá de ser feito.

O mapeamento entre os proprietários e os objetos em geral é feito com um livro-razão (Livro-Razão - também denominado "Razão Auxiliar" - é obrigatório pela legislação comercial e tem a finalidade de demonstrar a movimentação analítica das contas escrituradas no diário[6] e constantes do balanço) [19]. Esse documento não permanece constante depois de criado, toda transferência de posse deve ser documentada em um registro desse tipo porque um registro desatualizado não poderia ser uma testemunha confiável para comparar uma posse.

A Figura 12, vista a seguir, mostra a relação entre os conceitos envolvidos no projeto de um software para gerenciamento de posse. Os conceitos em cada camada podem ser vistos como realizações das camadas acima delas. Por exemplo, a comprovação de posse exige a identificação tanto dos proprietários quanto das propriedades, assim como um mapeamento entre eles. A utilização da posse exige a identificação, assim como a autenticação e a autorização, para que somente uma pessoa legítima utilize a propriedade. As caixas na linha inferior representam a camada de implementação. Elas mostram, por exemplo, que a senha e a assinatura são conceitos usados para implementar a autenticação e a autorização.

[6] O Livro Diário Registro apresenta, em ordem cronológica de data e sequencial de lançamento, todos os registros efetuados na contabilidade durante o período. É um livro obrigatório para as empresas [4].

Figura 12 - Conceitos de posse



Fonte: [8]

Conceito presente na base do gerenciamento de posse, o livro-razão pode ser visto como uma implementação concreta de um mapeamento entre os proprietários e suas propriedades. A Figura 13 mostra como a comprovação de posse e a sua transferência se relacionam com o propósito e as propriedades de um livro-razão.

Figura 13 - Livro-razão

Fonte: [8]

O livro-razão deve desempenhar dois papéis opostos. Por um lado, serve como meio para comprovar a posse, o que depende da leitura de dados históricos mantidos nele. Por outro, ele deve documentar qualquer transferência de posse, que, por sua vez, implica novos dados gerados e escritos no livro-razão. Uma das diferenças mais importantes entre esses dois propósitos pode ser resumida pela natureza oposta entre transparência e privacidade [8].

Comprovar a posse será mais fácil se o livro-razão estiver aberto a todos e, portanto, a transparência é a base para comprovar os direitos de posse. Entretanto, a transferência da posse deve estar restrita exclusivamente a quem a tiver por direito. Assim, a privacidade constitui a base da transferência de posse [8]. Como escrever no livro-razão significa alterar a posse, somente entidades muito confiáveis devem ter acesso de escrita. Contudo, o que aconteceria se um livro de registros como esse fosse danificado ou destruído? Ou o que aconteceria se alguém responsável por atualizá-lo cometesse um erro ou forjasse de propósito? Neste caso, o livro de registros não iria refletir a realidade. Esses problemas seriam solucionados da seguinte forma: em vez de manter um único livro-razão, deve-se utilizar um sistema ponto a ponto distribuído de livros-razão [8].

A relação entre gerenciamento de posse com um livro-razão e o Blockchain pode ser resumido da seguinte forma [8]:

a) Um livro razão individual é usado para manter informações sobre posse, o que equivale a uma estrutura de dados blockchain armazenando dados relacionados à posse;

b) Os livros-razão individuais são armazenados nos nós de um sistema ponto a ponto;

c) O algoritmo de blockchain é responsável por deixar que nós individuais cheguem coletivamente a uma versão consistente do estado de posse na qual o veredicto final será baseado;

d) A integridade nesse sistema é a sua capacidade de fazer afirmações verdadeiras sobre posses; e

e) A criptografia é necessária para criar um meio confiável para a identificação, a autenticação e a autorização, além de garantir a segurança dos dados.

3.1.1 Conceitos de gerenciamento de posse com o Blockchain

O objetivo dessa secção é apresentar os conceitos que compões o Blockchain e, para isso, há sete tarefas principais a serem tratadas ao fazer o *design* e o desenvolvimento de um *software* que gerencie posses usando um sistema ponto a ponto distribuído de livros-razão em um ambiente aberto e não confiável [8]. São eles:

a) Descrever a posse: as transações são uma boa maneira de descrever qualquer transferência de posse, e o histórico completo das transações é a chave para identificar os proprietários atuais;

b) Proteger a posse: é necessário ter um modo de evitar que as pessoas acessem as propriedades dos outros. Assim, pode-se utilizar criptografia, que oferece uma maneira de proteger as transações em um nível individual. A proteção da posse inclui três elementos principais: identificar e autenticar os proprietários e restringir o acesso à propriedade aos proprietários;

c) Armazenar dados de transação: é necessário um modo de armazenar todo o histórico de transações, pois ele será usado para deixar claro de quem é a posse. Como o histórico de transações é o elemento essencial para elucidar a posse, deverá ser armazenado de forma segura. A estrutura de dados blockchain é o equivalente digital de um livro-razão;

d) Preparar livros-razão para serem distribuídos em um ambiente não confiável: em um sistema ponto a ponto distribuído existirão cópias do livro-razão executando em nós não confiáveis, em uma rede não confiável. A melhor maneira de evitar que o histórico de transações seja alterado é deixando-o inalterável, ou seja, não poderão ser modificados depois de escritos. Contudo, é necessário aceitar o acréscimo de novas transações. Assim uma estrutura de dados blockchain deve permitir somente a concatenação: é possível adicionar novas transações, mas é praticamente impossível alterar dados adicionados anteriormente;

e) Distribuir os livros-razão: apenas fornecer cópias de livros-razão que só permitam concatenação não servira para atender os objetivos desejados: um sistema distribuído que gerencie posses envolvendo interação entre pares ou nós;

f) Adicionar uma nova transação nos livro-razão: como a estrutura de dados permite adicionar novas transações, deve-se garantir que somente transações válidas e autorizadas sejam acrescentadas. Isso é possível permitindo que os membros do sistema adicionem novos dados e, além disso, transformando cada participante desse sistema em supervisores de seus pares. Como resultado, todos os membros supervisionarão uns aos outros e apontarão qualquer erro cometido pelos demais; e

g) Decidir quais os livros-razão representam a verdade: como o histórico de transações constitui a base para identificar os proprietários legítimos, ter diferentes históricos de transação conflitantes é uma séria ameaça à integridade do sistema. É necessário, então, de um critério para encontrar e selecionar um histórico de transações que represente a verdade. Entretanto, há outro problema: não há uma autoridade central em um sistema ponto a ponto capaz de declarar qual o histórico de transações deve ser escolhido. Pode-se solucionar este problema fazendo com que cada nó do sistema ponto a ponto decida por conta própria qual histórico de transações representa a verdade, de modo que a maioria dos participantes concorde independentemente com essa decisão. A maneira como o blockchain permite

acrescentar novas transações à estrutura de dados blockchain já contém a solução para esse problema.

3.1.2 Documentando a posse

Essa seção explica como o Blockchain documenta a posse e trata a sua transferência, além de destacar a importância da ordenação para a documentação da transferência de posse. Por fim, ela enfatiza a importância da integridade dos dados das transações para a integridade do sistema como um todo.

A documentação da posse com o Blockchain envolve descrever a transferência da posse e manter o histórico de transferências. Para descrever a transferência da posse, são necessárias algumas informações, tais como [8]:

a) Um identificador da conta que transferirá a posse para outra conta;
b) Um identificador da conta que passará a ter a posse;
c) A quantidade de bens sendo transferida;
d) O horário em que a transação deve ser feita;
e) Uma taxa a ser paga para o sistema por executar a transação; e
f) Uma prova de que o proprietário da conta que está transferindo a posse realmente está de acordo com essa transferência.

A maior parte desses dados é conhecida de todos que já tenham feito uma transferência monetária em um banco, porém, a analogia termina quando as taxas são consideradas. Pelo fato de serem instituições centralizadas, os bancos mantêm uma tarifa central aplicada a todos os clientes. Em comparação, o Blockchain é um sistema distribuído, que não possui uma tarifa centralizada. Ao usar o Blockchain, cada usuário deve informar previamente ao sistema quanto está disposto a pagar para executar a transação.

O Blockchain mantém o histórico completo de todas as transações que já ocorreram armazenando os dados de transação na estrutura de dados blockchain na ordem em que ocorreram. Qualquer transação que não faça parte desse histórico é considerada uma transação que não aconteceu. Assim, adicionar dados de transação na estrutura de dados blockchain significa fazer essa transação acontecer, permitindo que ela influencie no resultado do histórico para identificar o proprietário atual. Portanto, esse histórico completo é suficiente para documentar a posse [8].

3.2 Transformando um livro em um Blockchain

Esta seção explica como transformar um livro em uma pequena biblioteca com um catálogo de ordenação, que na verdade, será uma versão simplificada da estrutura de dados blockchain.

Mesmo diante da tecnologia atual, os livros se mantêm no cotidiano das pessoas, e pelo fato de serem tão comuns, algumas inovações que eles trouxeram ao serem inventados passam despercebidas. Algumas de suas propriedades incluem:

a) Armazenamento de conteúdo: os livros armazenam conteúdo em suas páginas;

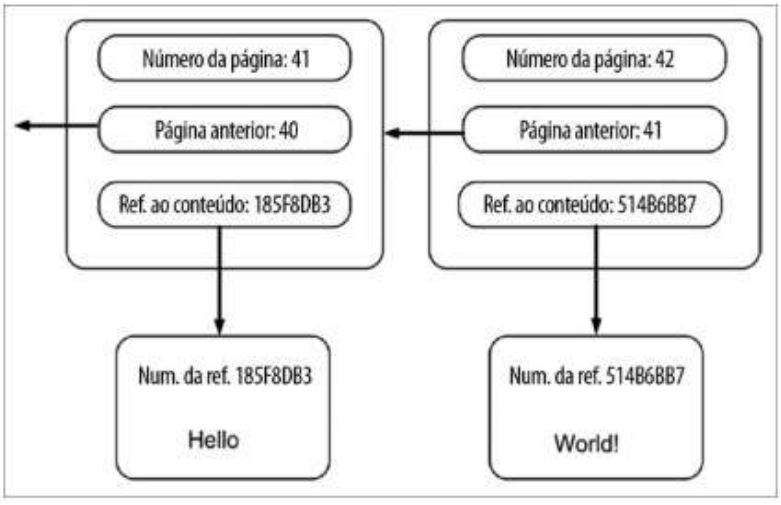
b) Ordenação: as frases nas páginas, assim como as páginas do livro, são ordenadas;

c) Conexão entre as páginas: as páginas estão fisicamente conectadas por meio da lombada (parte da costura do livro), e logicamente conectadas pelo conteúdo e pelos números das páginas.

Como consequência dessas propriedades, pode-se navegar pelos livros para frente e pra trás, virando as páginas, ou pular diretamente para páginas específicas utilizando seus números.

Os números das páginas servem para realizar a ordenação do livro, e se por exemplo, uma página fosse arrancada, seria possível identificá-la facilmente. Isso é possível pois a ordenação é realizada através de números naturais consecutivos. Caso a ordenação fosse realizada através de números primo, a tarefa de identifica-la seria mais difícil. Uma solução seria colocar em cada página não só o seu próprio número, mas também o número de sua página anterior, como pode ser observado na figura a seguir:

Figura 14 - Páginas de um livro com o número da página a anterior

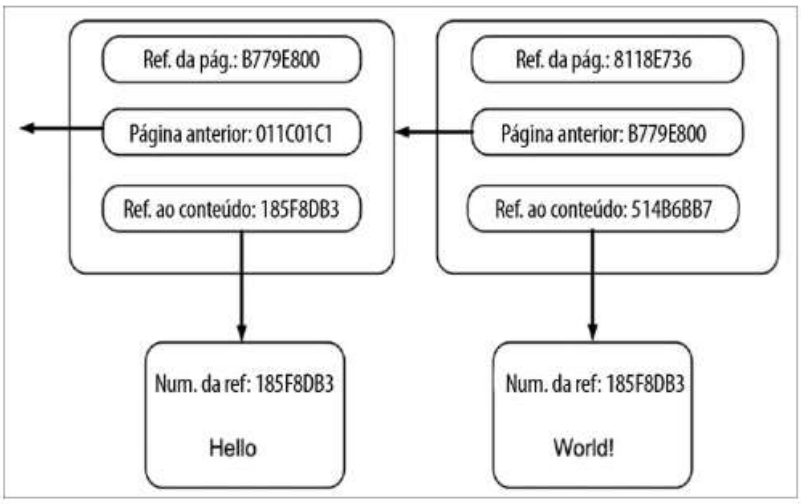


Fonte: [8]

Para deixar esse livro hipotético mais prático, pode-se exteriorizar o seu conteúdo, permitindo que ele se concentre somente na tarefa de manter a ordenação. Assim, as páginas não terão mais conteúdo, terão apenas números de referência que apontam para o conteúdo, o qual poderá ser armazenado em qualquer lugar. Ou seja, o livro foi transformado em uma pequena biblioteca. Onde antes era armazenado o conteúdo e os números das páginas em conjunto, agora é um catálogo cujo único objetivo é mantes a ordem do conteúdo [8].

Pode-se experimentar, também, um esquema de numeração diferente, substituindo os números naturais por números de referência, que são criados com base no número de referência da página anterior e o número de referência do conteúdo A Figura 15 mostra o resultado dessa transformação.

Figura 15 - Numeração das páginas como referências



Fonte: [8]

Este catálogo de ordenação (livro hipotético), apesar de todas mudanças, ainda continua sendo um livro tradicional, cujas páginas estão fixadas à lombada do livro. Contudo, se a lombada fosse retirada, o livro passaria a ser uma pilha de páginas soltas, sem a conexão física entre as páginas. Mesmo diante disso, a ordem das páginas não seria perdida, pois toda página contém o seu número de referência e o da página anterior.

Resumindo todos os passos acima: um livro tradicional foi transformado em duas pilhas de páginas soltas, ligadas por números de referência únicos. Uma pilha de páginas tem o conteúdo, enquanto a outra mantém a ordenação. Assim, este livro transformado é composto de [8]:

a) Uma unidade imaginária constituída de uma página do catálogo de ordenação e sua página de conteúdo correspondente;

b) Uma pilha de páginas de soltas chamada de catálogo de ordenação;

c) Uma pilha de páginas soltas com o conteúdo;

d) Números de referência de páginas para identificar e ligar as páginas do catálogo de ordenação; e

e) Números de referência de conteúdo para identificar e ligar as páginas de conteúdo.

Esses elementos podem ser comparados aos elementos de uma estrutura de dados blockchain, vistos no quadro abaixo:

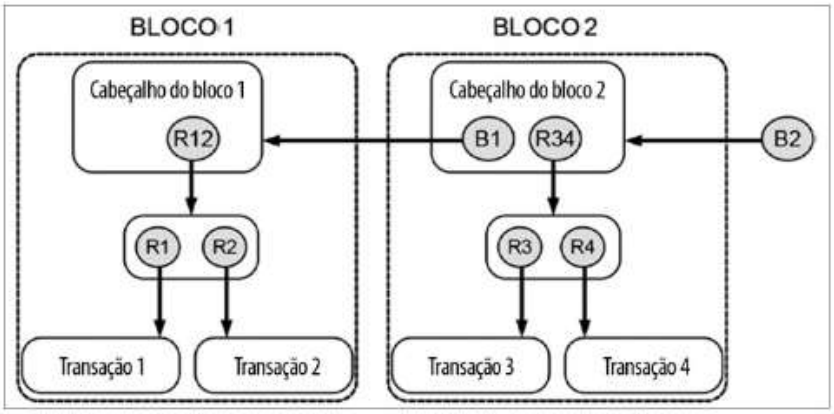
Quadro 2 – Comparação do livro aos elementos de uma estrutura de dados blockchain

Livro transformado	Estrutura de dados blockchain [17]
Página do catálogo de ordenação	Cabeçalho (header) de bloco
Catálogo de ordenação completo	Cadeia de cabeçalhos de bloco
Número de referência para a página anterior	Valor de hash de um cabeçalho de bloco
Número de referência para a página anterior	Valor de hash de um cabeçalho de bloco anterior
Conteúdo	Dados de transação
Página de conteúdo	Árvore de Merkle contendo dados de transação
Referência à página de conteúdo	Raiz da árvore de Merkle contendo os dados da transação
Unidade imaginária de uma página do catálogo de ordenação e sua página de conteúdo correspondente	Bloco da estrutura de dados blockchain
Catálogo de ordenação completo e todas as páginas de conteúdo juntas	Estrutura de dados blockchain

Fonte: [8]

A figura abaixo resume o que foi visto, apresentando esquematicamente uma estrutura de dados blockchain simplificada, que armazena quatro transações.

Figura 16 - Figura 16 - Estrutura de dados blockchain simplificada



Fonte: [8]

A figura acima mostra uma estrutura de dados blockchain simplificada, constituída de dois blocos (BLOCO 1 e BLOCO 2) que foram desenhados com linha tracejadas para enfatizar à natureza imaginária dos blocos. O BLOCO 1 é o primeiro nessa estrutura de dados, portanto, não tem um bloco anterior e, consequentemente, o cabeçalho do Bloco 1 não contém referência a um bloco anterior. A estrutura de dados blockchain representada mantém referências de hash para duas árvores de Merkle distintas, cujas raízes têm como rótulos R12 e R34, que indicam os dados de transação contidos (R12 contém as duas primeiras transações – transação 1 e transação 2 – e suas referências de hash correspondentes, R1 e R2, que apontam para elas). O cabeçalho do bloco mais recente adicionado e a referência que aponta para ele são chamados de cabeça da estrutura de dados blockchain. Na Figura 16, a cabeça da estrutura é a referência B2. Ressalta-se a importância de não confundir os termos “cabeça” (head) e “cabeçalho” (header): a estrutura de dados blockchain é constituída de vários blocos, cada um com o próprio cabeçalho, porém possui apenas uma cabeça.

Caso se deseje adicionar novas transações (transação 5 e transação 6), os seguintes passos devem ser seguidos:

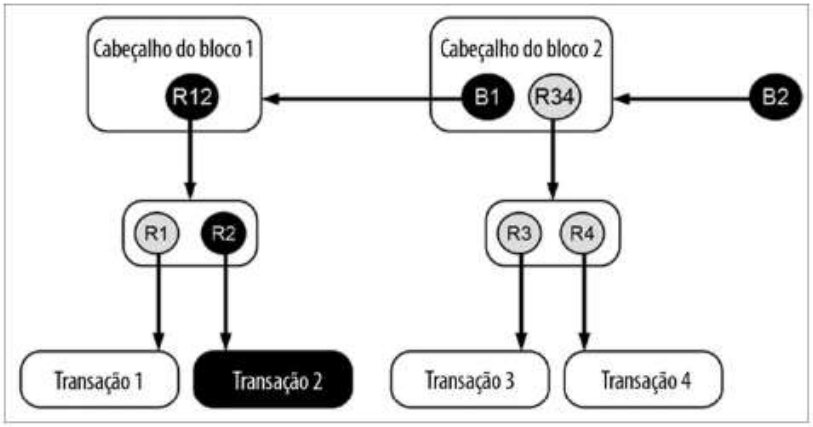
1) Criar uma nova árvore de Merkle contendo todos os dados das novas transações;

2) Criar um novo cabeçalho de bloco (cabeçalho do Bloco 3) contendo a referência de hash (B2), que aponta para o cabeçalho de seu bloco anterior (cabeçalho do bloco 2), e a raiz da árvore de Merkle, que contém os dados das novas transações (R56).

3) Criar uma nova referência de hash (B3) e a declarar como a nova cabeça da estrutura de dados blockchain.

Já para modificar ou atualizar uma estrutura de dados blockchain, é necessário modificar todas as referências de hash subsequentes, como ser visto na figura a seguir:

Figura 17 - Alterando transações



Fonte: [8]

Caso se deseje alterar algum detalhe da transação 2, deve-se atualizar toda sequência de hash: R2, R12, B1 e B2. Vale ressaltar que a estrutura de dados blockchain não diferencia mudanças intencionais e não intencionais, ela só se importa com a consistência das referências de hash, que se for inválida, resultará em uma estrutura de blockchain inválida.

Tendo em vista a possibilidade de alteração dos dados, um dos maiores desafios do Blockchain é manter o sistema aberto a todos, e ainda assim, proteger o histórico de dados de transação de modo que não seja manipulado por nós desonestos [8]. Um modo eficaz de resolver esse problema é deixar o histórico de dados de transação imutável, ou seja, que não possa ser alterado.

A principal ideia usada pelo Blockchain para deixar o histórico de transações imutável é fazer com que o alterar seja tão custoso a ponto de tornar-se uma tarefa impeditiva. Assim, o conjunto de tecnologias blockchain faz com que o conteúdo da estrutura de dados seja imutável, impondo custos de processamento significativos para todo bloco escrito, reescrito ou adicionado à estrutura. Os custos computacionais são exigidos por quebra-cabeças de *hash*, que são únicos para cada cabeçalho de bloco, como já visto anteriormente.

O procedimento para adicionar um novo bloco à estrutura de dados blockchain, visto até agora, não precisa de um alto processamento, pois exige apenas a adição da referência de *hash* que aponta para a cabeça atual da cadeia no novo cabeçalho de bloco e declará-la como a nova cabeça da cadeia. Contudo, o desafio de deixar a estrutura de dados blockchain imutável é fazer com que a adição de um novo bloco seja uma tarefa custosa do ponto de vista de processamento. Os aspectos a seguir devem ser considerados para que isso seja feito [8]:

a) Todo cabeçalho de bloco da estrutura de dados blockchain deve ter, no mínimo, os seguintes dados:

i. A raiz de uma árvore de Merkle contendo os dados da transação;
ii. Uma referência de *hash* para o cabeçalho do bloco anterior;
iii. O nível de dificuldade do quebra-cabeça de *hash*;
iv. O horário em que teve início a solução do quebra-cabeça de *hash*; e
v. O *nonce* que resolve o quebra-cabeça de *hash*.

b) O processo de criação de um novo cabeçalho de bloco envolve os seguintes passos:

i. Obter a raiz da árvore de Merkle que contém os dados de transação a serem adicionados;

ii. Criar uma referência de *hash* para o cabeçalho do bloco que será o antecessor do ponto de vista do novo cabeçalho de bloco;

iii. Obter o nível de dificuldade necessário;

iv. Obter o horário atual;

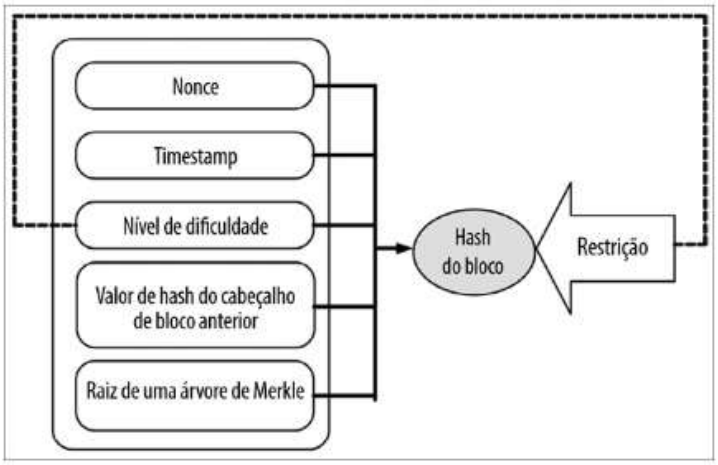
v. Criar um cabeçalho de bloco preliminar contendo os dados mencionados nos passos i a iv;

vi. Resolver o quebra-cabeça de *hash* para o cabeçalho de bloco preliminar; e

vii. Finalizar o novo bloco adicionando o *nonce* que resolve o quebra-cabeça de *hash* no cabeçalho preliminar.

A Figura 18 mostra o quebra-cabeça de *hash* que deve ser resolvido quando um novo bloco é adicionado à estrutura de dados blockchain. Ela mostra os dados do cabeçalho de bloco cujo valor de *hash* deve atender à restrição especificada ou ao nível de dificuldade. Pode-se notar que o nível de dificuldade faz parte do cabeçalho de bloco e, portanto, também faz parte de seu valor de *hash*. Isso garante que ninguém será capaz de evitar os custos de processamento do quebra-cabeça de *hash* reduzindo arbitrariamente o nível de dificuldade.

Figura 18 - Quebra-cabeça que deve ser resolvido



Fonte: [8]

c) Todo cabeçalho de bloco deve obedecer às regras de validação a seguir:

i. Deve conter uma referência de *hash* válida para um bloco anterior;

ii. Deve conter uma raiz válida para uma árvore de Merkle contendo os dados da transação;

iii. Deve conter um nível de dificuldade correto;

iv. Seu *timestamp* deve ser posterior ao *timestamp* de seu cabeçalho de bloco anterior;

v. Deve conter um *nonce*; e

vi. O valor de *hash* de todas as cinco porções de dados combinadas atende ao nível de dificuldade.

As regras de validação garantem que somente os blocos para os quais o quebra-cabeça de *hash* foi resolvido e os custos de processamento foram pagos serão adicionados à estrutura de dados blockchain. A regra iv garante que todos os dados de transação estão realmente ordenados de acordo com o horário em que foram adicionados.

A atividade de adição de um novo bloco à estrutura de dados blockchain por meio da resolução de um quebra-cabeça de hash é chamada de mineração (*mining*) ou mineração de bloco (*block mining*) [8].

4. FUNCIONAMENTO DO BLOCKCHAIN

A estrutura de dados blockchain faz com que qualquer alteração em seus dados seja perceptível em virtude da mudança das referências de *hash* com relação aos dados que elas referenciam. Isso provoca a necessidade de reescrever todos os blocos afetados por uma manipulação, o que é custoso do ponto de vista de processamento, fazendo com que a manipulação do histórico de transações não seja atraente. Como resultado, a estrutura de dado blockchain passa a ser um repositório de dados imutável, viável apenas para concatenações. Contudo, ter um histórico de dados de transação isoladamente pode ter um valor limitado quando se deseja esclarecer a posse, pois é necessário um grupo de computadores que atue como testemunhas [8]. Assim, é desejável o compartilhamento dessas informações.

4.1 Disseminando o repositório de dados

A comunicação entre os nós que compõem o sistema ponto a ponto distribuído possui os seguintes propósitos [8]:

a) Manter as conexões existentes ativas: de modo independente, cada computador na rede mantém uma lista dos pares com os quais se comunica. Regularmente, cada computador verifica se esses pares continuam disponíveis enviando-lhes uma pequena mensagem (*ping*) e solicitando uma mensagem de resposta (*pong*). Os pares que repetidamente não responderem são removidos;

b) Estabelecer novas conexões: um computador pode se associar a um sistema ponto a ponto enviando uma mensagem de requisição para um nó que compõe o sistema. O nó que receber a requisição adicionará o endereço de quem fez o pedido à sua lista de pares e enviará uma confirmação como resposta. Ao receber essa resposta, o nó adicionará o endereço de quem respondeu seu pedido na sua lista de pares. Como resultado, uma nova conexão é estabelecida e o sistema terá se expandido em mais um nó; e

c) Distribuir novas informações: este tipo de comunicação serve ao objetivo da aplicação do sistema, isto é, o gerenciamento de posses. Isto é feito encaminhando-se os novos dados de transação e os novos blocos a serem adicionados à estrutura de dados blockchain através de troca de informações, também chamadas de comunicações por fofoca (*gossip*). O compartilhamento de informações relacionadas à posse ocorre nas três ocasiões a seguir:

i. De forma contínua: novas informações são distribuídas à medida que ocorrem. Todo nó conectado ao sistema, em algum momento, receberá todas as notícias;

ii. Como uma atualização: os nós que se reconectarem ao sistema depois de terem sido desconectados por um momento receberão todos os dados de transação e blocos perdidos nesse período.

iii. Como parte do procedimento de associação: novos nós que se associarem ao sistema ainda não tiveram chance de construir o próprio histórico de transações. Assim, a transferência de uma cópia da versão atualizada da estrutura de dados blockchain se faz necessária.

Portanto, os diferentes tipos de comunicação garantem que novos nós se associem ao sistema, contribuindo com seu crescimento. Além disso, o sistema é mantido de maneira unida com base na comunicação que visa estabelecer novas conexões e manter as existentes. Somado a isso, o sistema utiliza uma comunicação que garante que, em algum momento, todos os membros do sistema receberão todos os dados de transação e blocos a serem adicionados no blockchain.

4.2 Verificando e adicionando transações

Por ser um sistema aberto, nós desonestos podem se conectar ao blockchain, criar transações e enviá-las aos outros nós que o compõe. Consequentemente, é difícil garantir que as transações enviadas pela rede estejam corretas. Para garantir que somente transações válidas sejam adicionadas ao sistema, todos os nós também têm permissão para atuar como supervisores de seus pares e recompensá-los por adicionar transações válidas e autorizadas e por encontrar erros no trabalho de outros. Como resultado, todos os nós do sistema têm um incentivo para processar corretamente as transações, supervisionar e apontar erros cometidos por qualquer um de seus pares [17].

O algoritmo de blockchain é uma sequência de instruções que governa como os nós processam novos dados de transação e blocos. As regras e os procedimentos individuais a seguir, podem ser associados aos blocos [8]:

a) Regra de validação: em última instância, o objetivo do algoritmo de blockchain é garantir que a estrutura de dados blockchain contenha somente blocos válidos, constituídos de dados de transação e cabeçalhos de bloco válidos. A validade desses dados é analisada com base em dois grupos distintos de regras de validação:

i. Regras de validação para dados de transação: as regras de validação para dados de transação definem quais dados são necessários para descrever uma transação. Essas regras incluem correção formal, correção semântica e autorização.

ii. Regras de validação para cabeçalhos de bloco: as regras de validação para cabeçalhos de bloco têm como foco a correção formal e semântica desses cabeçalhos. Essas regras são independentes do conteúdo dos dados de transação e dizem respeito ao modo como a informação é adicionada à estrutura de dados blockchain. Um elemento central na validação dos cabeçalhos é a verificação da prova de trabalho ou do quebra-cabeça de *hash*.

b) Recompensa: criar blocos válidos custa energia, tempo e dinheiro, pois exige resolver o quebra-cabeça de *hash* único para cada bloco, o que requer processamento. O quebra-cabeça de *hash* é o elemento fundamental para deixar a estrutura de dados blockchain imutável. Desse modo, torna-se indispensável resolvê-lo, sendo necessário oferecer uma recompensa pelo trabalho dispendido. Assim, o algoritmo de blockchain define como os nós que que submetem blocos válidos são recompensados;

c) Punição: o Blockchain também precisa de medidas para punir os participantes por agirem contra a integridade do sistema. Medidas típicas de punição são retomar a recompensa pelos blocos aceitos no passado, mas que tenham sido identificados como inválidos mais tarde. Outra forma de punição é a ausência de recompensa. Deixar que os nós façam a prova de trabalho, mas não os recompensar alegando que o bloco foi identificado como duplicado, antigo demais ou inútil. Ou sejam, não receber uma recompensa também é uma forma de punição; e

d) Competição: é importante evitar o desperdício de recursos na distribuição de recompensas, sendo essencial distribuir somente para os nós que contribuam significativamente com a manutenção do sistema. A melhor forma de fazer isso é estabelecer uma competição por recompensas com base em um critério bem definido. O algoritmo de blockchain mantém uma concorrência contínua por recompensas com base em dois critérios: competição por velocidade e competição por qualidade. Somente o nó que vencer as duas competições receberá a recompensa pela submissão de um novo bloco.

i. A competição por velocidade entre os nós é baseada no quebra-cabeça de *hash*. O elemento central na criação de um bloco válido é a prova de trabalho, o que significa resolver o quebra-cabeça de *hash* do novo bloco. Não há uma maneira de solucionar um quebra-cabeça de hash com antecedência, pois ele depende do próprio conteúdo do bloco. Como resultado, todos os nós participam da competição para resolver o quebra-cabeça de hash. Assim que um

nó submeter um novo bloco, a competição de velocidade termina e ele passa a ser o único candidato na competição de qualidade.

ii. A competição por qualidade tem como foco a correção do bloco submetido. Assim que um nó submeter um novo bloco, ele será enviado a todos os nós do sistema. Ao receber um novo bloco, cada nó deve agir como um árbitro na competição por qualidade, o que significa validar o novo bloco com base nas regras de validação. Se for constatado que o bloco é válido, o nó que o submeteu receberá a recompensa e uma nova competição por velocidade será aberta. Se for constatado que o bloco é inválido, ele será descartado e a competição por velocidade será reaberta com todas as transações que já estavam em jogo. Isso tem um aspecto interessante, pois os nós tem a possibilidade de retornar ao jogo em busca da recompensa se provarem que o bloco é inválido, assim farão um controle mais rigoroso possível.

4.3 Escolhendo um histórico de transações

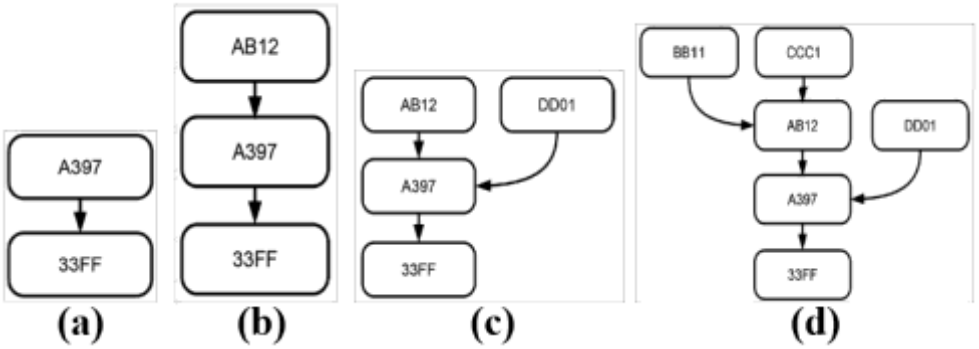
Em razão da prova de trabalho, adicionar um bloco novo é custoso do ponto de vista de processamento, e torna as tentativas de manipular o histórico de transações mais custosas ainda. Assim, o volume de esforço computacional agregado investido na criação de um histórico de transações parece um critério natural para selecionar um histórico no caso de haver mais de uma versão conflitante. Se todos os nós do sistema aplicarem o mesmo critério, em algum momento todos eles concordarão com uma versão idêntica do histórico. A versão do histórico de transações selecionada de maneira coletiva é usualmente chamada de cadeia autoritativa (*authoritative chain*) ou histórico autoritativo [8].

A ideia deselecionar um histórico de transações com base no esforço de processamento investido para criá-lo levou aos seguintes critérios:

a) Critério da cadeia mais longa [17]: o critério da cadeia mais longa se baseia na ideia de que a estrutura de dados blockchain contendo o maior número de blocos representa o máximo de esforço computacional agregado. O exemplo a seguir elucida esse critério:

Em uma situação inicial, todos os nós de um sistema mantêm uma versão idêntica da estrutura de dados blockchain e concordam com ela, conforme a Figura 19 (a), onde cada uma das caixas representa um bloco identificado por um valor de hash abreviado. A seta que aponta de uma caixa para outra representa a referência de hash que liga um cabeçalho de bloco ao seu antecessor.

Figura 19 - Critério da cadeia mais longa



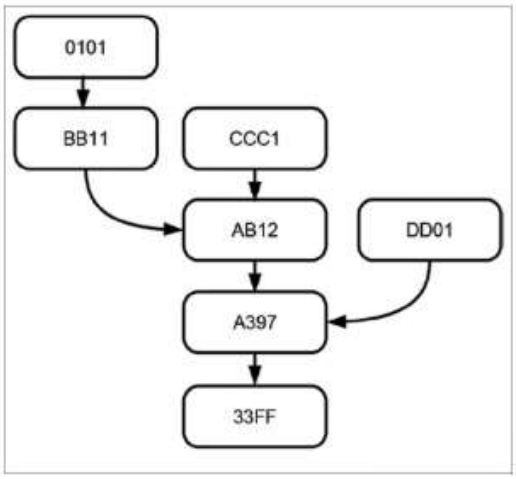
Fonte: adaptado de [8]

A Figura 19 (b) mostra a estrutura de dados blockchain que a maioria dos nós mantém depois que um deles resolveu o quebra-cabeça de *hash* de um novo bloco e o enviou aos seus pares. Do ponto de vista da maioria, há somente uma versão do Blockchain, constituída de três blocos. No entanto, enviar um novo bloco por uma rede exige tempo, e a operação pode se deparar com algum tipo de adversidade. Devido a um atraso na transmissão da mensagem, uma minoria dos nós não recebeu o bloco AB12. Desse modo, a estrutura para eles é a cadeia da Figura 19 (a). Em algum momento, um deles resolverá com sucesso o quebra-cabeça de hash para um novo bloco com valor de *hash* DD01 e passará para seus pares. A maioria dos nós receberá tanto o bloco AB12 quanto o bloco DD01 e manterão ambos, conforme a figura 19 (c), composta de dois ramos sobre um tronco comum. Nessa situação, o critério da cadeia mais longa irá gerar um resultado ambíguo, pois as duas cadeias têm o mesmo tamanho. Nessas circunstâncias, os nós têm a liberdade de decidir qual ramo vão estender.

Repentinamente, a maioria dos nós recebe dois novos blocos, BB11 e CCC1, ambos referenciando o bloco AB12 como seu antecessor. A incorporação desses dois novos blocos resulta em uma estrutura contendo três cadeias, conforme a figura 19 (d). Pelo critério da cadeia mais longa, descarta-se a cadeia DD01 → A397 → 33FF, porém novamente há ambiguidade em virtude de duas cadeias possuírem o mesmo tamanho.

Em algum instante, chegará um novo bloco, referenciando o bloco BB11 como seu antecessor, gerando a estrutura vista na figura 20.

Figura 20 - Estrutura com versões conflitantes



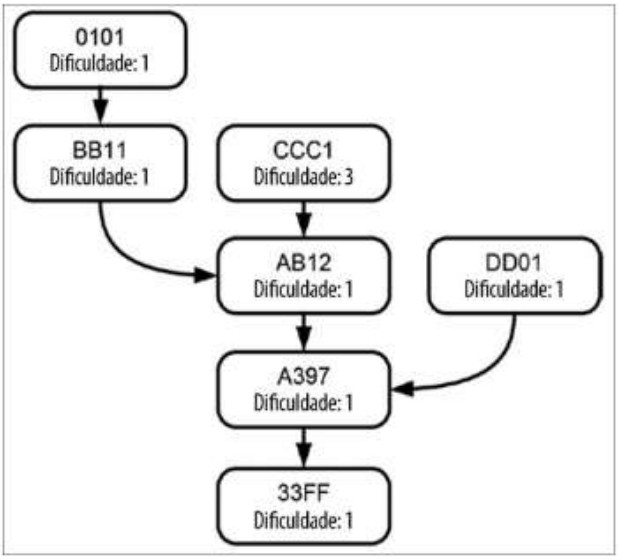
Fonte: [8]

Essa estrutura de dados contém muitas versões conflitantes do histórico de transações, mas o critério da cadeia mais longa produz um resultado não ambíguo, formada pelos blocos 0101 → BB11 → AB12 → A397 → 33FF.

Uma observação interessante é o fato do blockchain não se parecer com uma cadeia em linha reta, mas se assemelhar mais a uma árvore.

b) Critério da cadeia mais pesada [29]: as aplicações de blockchain raramente utilizam um nível de dificuldade constante para todos os blocos, normalmente a dificuldade é dinâmica e baseada na velocidade com que os novos blocos são adicionados, o que resulta em esforços de processamento diferentes para adicionar um novo bloco à estrutura blockchain. Assim, no critério da cadeia mais pesada, para cada caminho, esse esforço investido pode ser calculado pela soma dos níveis de dificuldade de todos os blocos que pertencem a esse caminho. A Figura 21 representa uma estrutura de dados igual à Figura 20, mas dessa vez, mostra também o nível de dificuldade para cada um de seus blocos.

Figura 21 -Critério da cadeia mais pesada



Fonte: [8]

Pode-se notar que, pelo critério da cadeia mais pesada, a cadeia escolhida será diferente (CCC1 → AB12 → A397 → 33FF).

Selecionar uma cadeia específica entre as diversas versões e defini-la como a cadeia autoritativa tem as seguintes consequências:

a) Blocos órfãos: todos os blocos da estrutura de dados em forma de árvore que não façam parte do caminho autoritativo serão abandonados pelos nós, e recebem o nome de blocos órfãos [18].

b) Recompensa restituída: blocos órfãos são inúteis para o propósito de esclarecer a posse, pois não contribuem com a cadeia autoritativa. Como resultado, a recompensa dada ao nó eu os criou e os submeteu deve ser restituída.

c) Robustez contra manipulações: o caminho da estrutura de dados blockchain em forma de árvore que representa o maior esforço computacional é a versão autoritativa do histórico de transações. Definir e manter o caminho autoritativo é somente uma questão de controlar a maior parte da capacidade de processamento de todo o sistema. Definir um novo caminho autoritativo que comece em um dos blocos internos da estrutura do blockchain exige se alinhar com o caminho mantido pela maioria e dominá-lo. Esse fato é a base para a robustez do blockchain. Desde que os nós honestos sejam donos da maior parte dos recursos computacionais do sistema, o caminho mantido por eles crescerá mais rapidamente e superará qualquer caminho concorrente. Para manipular um bloco interno, um invasor teria de refazer a prova de trabalho desse bloco, refazer o quebra-cabeça de hash de todos os blocos depois e, em seguida, teria de se alinhar com o caminho mantido pelos nós honestos de dominá-lo [17].

5. LIMITAÇÕES, CONFLITOS E VULNERABILIDADES

5.1 Limitações do Blockchain

A natureza aberta do blockchain a ausência de controle central são as bases de seu funcionamento, mas também podem causar limitações para sua adoção. As principais limitações técnicas são:

a) Falta de privacidade: o blockchain é um livro-razão ponto a ponto distribuído que mantém o histórico de dados de transação completo. Todos os detalhes das transações (bens, quantidades transferidas, contas envolvidas e horário da transferência) estão acessíveis a todos [17]. Isso é necessário para permitir que qualquer participante esclareça de quem é a posse e verifique novas transações. Assim, a falta de privacidade é um elemento constituinte do blockchain. Contudo, essa característica é um fator limitante para aplicações que exijam privacidade;

b) Custos elevados e alto processamento [8]: o blockchain é um sistema ponto a ponto que visa alcançar dois objetivos: permitir que todos adicionem novos dados de transação no histórico, mantido coletivamente, e garantir que o histórico esteja protegido, evitando que seja manipulado. O Blockchain atinge esses objetivos utilizando uma estrutura de dados imutável, mas que aceite concatenações, exigindo a solução de um quebra-cabeça de *hash* para adicionar um novo bloco. Resolver o quebra-cabeça de *hash* ou fornecer a prova de trabalho é propositalmente custoso do ponto de vista de tempo e processamento, resultando em um alto consumo de energia elétrica e necessidade de hardwares potentes. Exemplo disso é o Bitcoin, que é responsável por 0,25% do consumo total de energia em âmbito mundial, segundo análise realizada pelo Índice de Consumo de Eletricidade Bitcoin de Cambridge (CBECI), ferramenta online desenvolvida pela Universidade de Cambridge para estimar o gasto de energia, em tempo real, da rede bitcoin [27];

c) Centralidade oculta: em virtude da alta necessidade de processamento para a solução do quebra-cabeça de *hash*, o sistema acaba se tornando rentável apenas para aqueles que investirem em hardwares especializados. Assim, validar e adicionar novos dados de transação ao sistema deixará de ser rentável para aqueles que não possuírem *hardware* especializado, fazendo com que estes deixem de contribuir com recursos computacionais para o Blockchain. Como resultado, o grupo de participantes supostamente grande e diversificado, que mantém coletivamente a integridade do sistema, em algum momento se tornará um grupo bem pequeno de entidades, cada um com uma enorme capacidade de processamento, detendo

o controle da maior parcela das recompensas. Desse modo, esse pequeno grupo de entidades poderia abusar de seu poder, determinando uma espécie de centralidade oculta e debilitando a natureza distribuída do sistema, que ainda continuará sendo distribuída, porém com integridade duvidosa [16]; e

d) Tamanho crítico: a robustez contra manipulações depende da suposição de que a maior parte da capacidade de processamento do sistema é controlada por nós honestos. No entanto, em sistemas ponto a ponto pequenos, com capacidade de processamento limitada, essa maioria ainda pode ser bem pequena, o que, por sua vez, poderia possibilitar um ataque de 51%. Esse problema é particularmente relevante para criptomoedas com baixo mercado de capitalização e adoção limitada de usuários. Desse modo, alcançar um tamanho crítico que impossibilite ataques de 51% é um desafio que todo blockchain novo deve enfrentar.

5.2 Conflitos do Blockchain

Há dois conflitos importantes no Blockchain, representados por duas de suas principais limitações técnicas: transparência x privacidade e segurança x velocidade [8].

a) Transparência x privacidade: o blockchain define a posse através do histórico de dados de transação, que se assemelha a um registro público de transações ou a um livro-razão público. Ser aberto e transparente é um conceito intrínseco do blockchain para a verificação de posse. Essa natureza aberta constitui a base para resolver o problema de gasto duplo, pois qualquer um pode auditar as transações de todos os demais e, desse modo, descobrir facilmente ataques de tipo. Essa abordagem, porém, se opõe ao conceito de privacidade, que significa manter os dados de transação ou os seus detalhes, como contas envolvidas ou valores transferidos, ocultos do público.

b) Segurança x velocidade: o histórico de dados de transação é protegido para que não seja manipulado e armazenado em uma estrutura de dados imutável, mas que aceite concatenações, exigindo a solução de um quebra-cabeça de hash para adicionar um novo bloco ou reescrever um bloco já existente. Isso deixa custoso manipular ou forjar o histórico de dados de transação, a ponto de ser proibitivo, e reduz a velocidade com que novos dados podem ser adicionados ao blockchain. Assim, o conflito resultante está em proteger o histórico de dados de transação com base em uma prova de trabalho que consome tempo, por um lado, e os requisitos de velocidade por outro.

As origens dos conflitos vistos acima são duas operações fundamentais do Blockchain: leitura e escrita de dados de transação [8]. O conflito entre a transparência e privacidade pode

ser associado à leitura do blockchain, enquanto o dilema entre segurança e velocidade pode ser ligado à escrita de dados nessa estrutura.

A incompatibilidade entre esses objetivos pode ser solucionada estabelecendo-se um compromisso ou decidindo em favor de uma opção em detrimento a outra. Quanto ao conflito entre transparência e privacidade, o cerne da questão fica em conceder o acesso de leitura a todos ou a um grupo limitado de nós, ou seja, blockchains públicos (concedem acesso de leitura e o direito de criar transações a todos os nós) e blockchains privados (limitam o acesso de leitura e o direito de criar transações a um grupo pré-selecionado de nós). Já quanto ao conflito entre segurança e velocidade, o cerne da questão fica em conceder o acesso de escrita a todos, mas deixá-la custosa do ponto de vista de processamento (blockchains sem permissão), ou restringir o acesso de escrita a um grupo pré-selecionado de nós identificados como confiáveis, com uma versão menos custosa da prova de trabalho (blockchains com permissão) [8].

Vale ressaltar que restringir o acesso de leitura ou escrita no histórico de dados de transação causa impactos em alguns aspectos do blockchain, como a arquitetura e seu propósito. O blockchain estudado neste trabalho, conforme visto, decidiu em favor da transparência e segurança.

5.3 O problema do gasto duplo e sua relação com a integridade

Em um sistema ponto a ponto distribuído, os livros-razão que controlam as informações de posse são mantidos pelos computadores individuais de seus participantes e, desse modo, cada participante mantém a própria cópia dele. Assim que a posse de um bem é transferida de uma pessoa a outra, todos os livros-razão do sistema devem ser atualizados para que contenham a versão mais recente da realidade. Entretanto, passar informações para os membros e atualizar os livros-razão individuais exige tempo. Até que o último participante do sistema receba a nova informação e atualize a sua cópia, o sistema não estará consistente [8]. O fato de nem todos os livros-razão apresentarem informações atualizadas os deixa suscetíveis à exploração por alguém que já tenha a informação mais recente.

Em um nível mais abstrato, o problema do gasto duplo pode ser visto como um problema de manutenção da consistência dos dados em sistemas ponto a ponto distribuídos. Como a consistência dos dados constitui um dos aspectos da integridade do sistema, pode-se dizer que o problema do gasto duplo é um exemplo específico de violação da integridade do sistema.

Para resolver esse problema o Blockchain combina duas tecnologias e dessa forma garante que não haverá mais preocupação com esta questão: a metodologia de comunicação ponto a ponto e a criptografia (utilizando chaves públicas e privadas), que fornecem confiabilidade e segurança para a rede Blockchain [8].

A resolução do problema do gasto duplo através de confirmações da rede que utilizam algoritmos de criptografia que não podem ser quebrados é a garantia tecnológica, sem influência humana, que o sistema é íntegro. Qualquer transação que receber o número máximo de confirmações da rede será incluída na Blockchain e a outra será encarada como "gasto duplo" e será descartada automaticamente pela rede, tornando-se um "bloco-órfão" [13]

Sendo assim, a rede Blockchain se protege contra "gastos duplos" através da verificação de cada transação, por meio do algoritmo de validação conhecido como prova de trabalho, realizado pelos chamados "mineradores", responsáveis por auditar e confirmar, ou seja, "minerar" todas as transações realizadas, prescindindo de uma autoridade central para tanto [13].

5.4 Ameaças ao esquema de votação

No processo de alcançar um acordo coletivo no que diz respeito ao histórico de transações, os blocos individuais que compõe a estrutura de dados blockchain podem ser vistos como uma célula de votação, enquanto o quebra-cabeça de *hash* seria um preço que deixa a submissão de uma célula custosa, detendo nós desonestos que não fazem parte dessa votação [8].

Qualquer procedimento de tomada de decisão coletiva será alvo de manipulações caso a influência no resultado compense. O Blockchain e seu algoritmos de consenso distribuído não é uma exceção. Independentemente de como essas manipulações pareçam diversificadas, elas têm um único objetivo: transformar os blocos que fazem parte da cadeia autoritativa em blocos órfãos e definir uma nova cadeia autoritativa que represente um histórico de dados de transação e uma distribuição alternativa dos direitos de posse mais favorável do ponto de vista dos invasores.

No entanto, é possível discutir essas manipulações de vários pontos de vista. Quanto à tomada de decisão coletiva, essas manipulações tentam reunir a maior parte do poder de votação a fim de impor um resultado desejado. Economicamente, elas tentam alterar a alocação dos direitos de posse. Do ponto de vista da arquitetura, essas manipulações tentam

definir, pelo menos temporariamente, um elemento oculto de centralidade que altere o estado do sistema. Em geral, esses ataques são chamados de ataques 51%[7].

Qualquer tentativa de manipular o processo de tomada de decisão coletiva do blockchain pretende reunir a maior parte do poder de votação, e pelo fato do blockchain vincular o poder de votação à capacidade de processamento por meio do quebra-cabeça de hash, qualquer tentativa de reunir a maior parte desse poder de votação, na verdade, significa reunir a maior parte da capacidade de processamento de todo o sistema ponto a ponto.

[7] Um ataque 51% é uma tentativa de reunir ou controlar a maior parte de todo o poder de votação em um processo de tomada de decisão coletiva [8].

6. APLICAÇÕES DO BLOCKCHAIN

6.1 Aplicações teóricas do Blockchain

Com base nas propriedades do Blockchain e em sua característica de repositório para qualquer tipo de dado, os seguintes casos de uso podem ser adotados [8]:

a) Prova de existência: esse uso do Blockchain tem como foco a armazenagem de dados com o único propósito de comprovar a sua existência. Portanto, os recursos de ordenação e os de *timestamp* não são usados. Exemplos desse tipo, englobam: registros de itens que se supõe serem únicos, como marcas, patentes, códigos de licença e endereços de internet e e-mail;

b) Prova de tempo: neste caso, o instante em que um dado foi adicionado é importante. O Blockchain pode atender essa necessidade pois os blocos armazenam o horário do processo de sua adição. Exemplos desse tipo englobam eventos monitorados no tempo, como: notificações de entrega e pagamento, abertura e encerramento de procedimentos de licitações públicas e gerenciamento de previsões;

c) Prova de identidade: o blockchain atende esse caso de uso pois armazena dados que podem ser usados par identificar alguém ou algo e provê conceitos básicos de segurança para identificação e autenticação. Aplicações desse tipo incluem: documentos de identidade digitais para pessoas, animais ou bens e administração, por parte do governo, de documentos pessoas, como carteiras de habilitação e passaportes; e

d) Prova de posse: esse padrão de uso tem como foco gerenciar e deixar claro de quem é a posse. Exemplos desse tipo englobam: sistemas para gerenciar posses de imóveis, carros, ações de empresas, títulos e criptomoedas.

6.2 Casos reais de aplicações do Blockchain

Já existem diversas aplicações do blockchain sendo utilizadas no dia a dia, algumas delas são [1]:

a) Categoria: Segurança nas redes

A Remme é um projeto do blockchain de segurança cibernética que visa melhorar os padrões atuais de segurança para usuários e empresas que estão tentando proteger seus dados de uso não autorizado. Ela também foi construída para lidar com os ataques cibernéticos e

deve ajuda as organizações com infraestrutura crítica, tecnologia médica, trocas de criptografia e muito mais.

Além disso, essa empresa deve eliminar completamente o aspecto humano automatizando tudo. Os ataques comuns contra os quais a Remme protege incluem Brute Force, Phishing, Keylogging, ataques de reutilização de senhas e muito mais.

b) Categoria: Saúde

A SimplyVital Health é uma solução blockchain que conecta pacientes e fornecedores sob o mesmo teto. É uma solução de gerenciamento de dados de saúde e também usa previsões de aprendizado de máquina e algoritmos para a melhor solução possível. Os insights analíticos são fornecidos para a experiência do paciente. Ele também funciona igualmente para provedores privados, hospitais e sistemas de saúde. Um dos seus principais componentes é o Health Nexus. É um protocolo compatível com HIPAA que é aceito em todo o mundo. Um protocolo que pode ser usado para conectar dados de assistência médica e controle do paciente.

Outra aplicação na saúde é a MedRec, que é uma plataforma blockchain que resolve o problema dos Registros Eletrônicos de Saúde (EHRs) ao fornecer uma solução blockchain, na qual os dados dos pacientes podem ser armazenados e acessados por diferentes organizações, indivíduos e especialistas em saúde. Isso cria um ecossistema que funciona para os pacientes e lhes fornece a capacidade de ter seus relatórios médicos na ponta do dedo. O uso de blockchain também oferece transparência e cria uma impressão individual única. Isto leva a custos mais baixos e suporte de cuidados de saúde mais fácil para o paciente.

c) Categoria: Armazenamento

A STORJ.io é uma plataforma de armazenamento em nuvem distribuída que se concentra em dar controle aos usuários, fornecendo armazenamento criptografado de ponta a ponta. Somente os usuários podem ter acesso aos seus dados, e os pagamentos blockchain o habilitam, permitindo que os usuários aluguem espaço para a rede. Ao fazer isso, um usuário pode ganhar dinheiro compartilhando o espaço HD. É também de código aberto e gratuito, e qualquer um pode ajudar a desenvolver a solução.

d) Categoria: Governo

A Samsung SDS fez uma parceria com o governo sul-coreano para criar serviços governamentais mais transparentes que podem ajudar a melhorar os três principais componentes da sociedade que é segurança pública, bem-estar e transporte. Está previsto para ser concluído em 2020.

e) Categoria: Processo eleitoral

O sistema das eleições sempre foi alvo de questionamentos, seja ele físico ou eletrônico. Em março de 2018, um distrito do país africano Serra Leoa atualizou o seu processo de votação para o modelo digital. O distrito de Freetown armazenou os votos através da tecnologia blockchain, onde cada voto era registrado em um bloco particular, acessado apenas pelos funcionários que fariam a conta.

Outro exemplo é o aplicativo desenvolvido pela fundação Democracy Earth, sediada na Califórnia, chamado Sovereign. A partir dele, votos são registrados em blockchain para que sejam contabilizados com segurança. Com isso, eleitores podem votar por meio de chaves eletrônicas, com um número de votos e candidatos específico. Isso impede a manipulação de resultados. Porém, a tecnologia ainda não foi adotada em votações de larga escala.

f) Categoria: Sistema financeiro [9]

Com o uso do blockchain, fazer pagamentos internacionais pode se tornar mais rápido e eficiente. O Programa Alimentar Mundial das Nações Unidas (PMA) criou um sistema de pagamentos baseado na criptomoeda Ethereum, plataforma que realiza contratos e aplicações descentralizadas por meio da tecnologia blockchain.

Outro exemplo é que a IBM, gigante do setor de TI, desenvolveu uma maneira de auxiliar instituições financeiras a processar pagamentos internacionais por meio do blockchain, com o intuito de alavancar a tecnologia de contabilidade e oferecer uma transparência maior para todos os envolvidos nos processos bancários.

7. CONCLUSÃO

O principal problema solucionado pelo Blockchain é prover e manter a integridade em um sistema ponto a ponto distribuído, constituído de um número desconhecido de nós, com nível de confiabilidade desconhecido. Para isso, é necessário que ele proteja o histórico de dados de transação de modo que não seja manipulado por nós desonestos. Um modo eficaz de resolver esse problema é deixar o histórico de dados de transação imutável, ou seja, que não possa ser alterado.

A principal ideia usada pelo Blockchain para deixar o histórico de transações imutável é fazer com que o alterar seja tão custoso a ponto de tornar-se uma tarefa impeditiva. Assim, o conjunto de tecnologias blockchain faz com que o conteúdo da estrutura de dados seja imutável, impondo custos de processamento significativos para todo bloco escrito, reescrito ou adicionado à estrutura. Os custos computacionais são exigidos por quebra-cabeças de *hash*, que são únicos para cada cabeçalho de bloco.

A estrutura de dados blockchain faz com que qualquer alteração em seus dados seja perceptível em virtude da mudança das referências de *hash* com relação aos dados que elas referenciam. Isso provoca a necessidade de reescrever todos os blocos afetados por uma manipulação, o que é custoso do ponto de vista de processamento, fazendo com que a manipulação do histórico de transações não seja atraente. Como resultado, a estrutura de dado blockchain passa a ser um repositório de dados imutável, viável apenas para concatenações, que propicia segurança e privacidade aos usuários do sistema.

7.1 Considerações Finais

Blockchain é uma solução tecnológica que permite o registro de transações com o uso de uma espécie de banco de dados virtual. Esses dados são alocados de forma compartilhada em uma rede de computadores interconectados, por meio dos quais as informações sobre transações são registradas. Toda transação dentro da estrutura blockchain recebe uma identificação e a identificação da transação anterior a ela. Com isso, é possível fazer todo o caminho reverso das transações. Toda ocorrência relacionada a um bloco de dado (como quem acessou, quem fez a mudança e quando) é registrada automaticamente no sistema, sem a possibilidade de alteração.

7.2 Sugestões para Futuros Trabalhos

Em virtude da necessidade de proteger diversas informações de caráter estratégico militar, sugere-se, para trabalhos futuros, um estudo para a aplicação do Blockchain para o armazenamento de dados na Marinha do Brasil.

REFERÊNCIAS

[1] 101 BLOCKCHAIN. **Blockchain applications**. Disponível em: <https://101blockchains.com/blockchain-applications/>. Acesso em: 30/12/2019.

[2] ACADEMY.**O que é árvore de Merkle**. Disponível em: <https://academy.bit2me.com/pt/o-que-e-arvore-de-merkle/>. Acesso em: 20/01/2020.

[3] ACADEMY. **O que é nonce**. Disponível em: <https://academy.bit2me.com/pt/o-que-e-nonce/>. Acesso em: 20/01/2020.

[4] BÄSCHTOLD, Ciro. **Contabilidade Básica**. Curitiba: Instituto Federal Paraná, 2011. 138 p.

[5] BINANCE. **What is hashing**. Disponível em: <https://www.binance.vision/pt/security/what-is-hashing>. Acesso em: 05/01/2020.

[6] Boritz, J. Efrim.IS **practitioners'views on core concepts of information integrity**. International Journal of Accounting Systems. Elsevier. Volume 6, Issue 4. December 2005: 260-279.

[7] Cormen, Thomas H. **Introduction to algorithms**. 3 ed. Cambridge:2009.

[8] DRESCHER, Daniel. **Blockchain Básico**: Uma introdução não técnica em 25 passos. São Paulo: Novatec, 2018. 364 p.

[9] EXAME. **Blockchain entenda o que é e quais são as principais aplicações**. Disponível em: <https://exame.abril.com.br/tecnologia/blockchain-entenda-o-que-e-e-quais-sao-as-principais-aplicacoes>. Acesso em: 31/01/2020.

[10] ITFORUM. **A importância do Blockchain para mudar a relação do usuário na web 3.0**. Disponível em: <https://www.itforum365.com.br/a-importancia-do-blockchain-para-mudar-relacao-do-usuario-na-web-3-0/>. Acesso em 18/12/2019.

[11] IVAREJO. **Blockchain**. Disponível em: <http://ivarejo.com.br/blog/blockchain/>. Acesso em: 20/12/2019.

[12] JESUS, Emanuel Ferreira. **Stalker**: Uma Nova Estratégia para o Atacante Egoísta em Blockchains. Orientador: ANTÔNIO AUGUSTO DE ARAGÃO ROCHA. 2018. 73 p. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal Fluminense, Niterói, 2018.

[13] JUSBRASIL. **O que é gasto duplo e como o Bitcoin é capaz de evitá-lo**. Disponível em: <https://brunonc.jusbrasil.com.br/artigos/584812107/o-que-e-gasto-duplo-e-como-o-bitcoin-e-capaz-de-evita-lo>. Acesso em: 15/01/2020.

[14] KASPERSKY. **Hash o que são e como funcionam**. Disponível em: <https://www.kaspersky.com.br/blog/hash-o-que-sao-e-como-funcionam/2773/>. Acesso em:31/01/2020.

[15] KAUARK, F. S.; MANHÃES, F. C.; MEDEIROS, C. H. **Metodologia da Pesquisa**: um guia prático. Itabuna – Bahia. Via Litterarum, 2010. 89p.

[16] Kroll, Joshua A., Ian C. Davey e Edward W. Felten. **The economics of Bitcoin mining, or Bitcoin in the presence of adversaries**. Anais do WEIS. 2013.

[17] Nakamoto, Satoshi. **Bitcoin: a peer-to-peer electronic cash system**. 2008. Bitcoin. Disponível em: <https://bitcoin.org/bitcoin.pdf>. Acesso em: 09/10/2019.

[18] Okupski, Krzysztof. **Bitcoin Developer Reference**. Última alteração: 30 de julho de 2016.

[19] Portal de Contabilidade. **Livro razão**. Disponível em: <http://www.portaldecontabilidade.com.br/obrigacoes/livrorazao.htm>. Acesso em: 15/01/2020.

[20] PRODANOV, C. C. e FREITAS, E. C. **Metodologia do Trabalho Científico**: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico. 2. ed. Novo Hamburgo – Rio Grande do Sul. Universidade FeeVale, 2013. 277p.

[21] Rogaway, Philip e Thomas Shrimptom. **Cryptographic hash-function basic**: definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. B. Roy and W.Meier (eds), Fast software encryption. FSE 2004. Berlin Heidelberg: Springer, 2004.

[22] SILVA, E. L.; MENEZES E. M. **Metodologia da pesquisa e elaboração de dissertação**. 4. ed. Florianópolis: UFSC, 2005. 138p.

[23] Tanenbaum, Andrew S. e Maarten Van Steen. **Sistemas Distribuídos**: princípios e paradigmas. Pearson Education do Brasil, 2007.

[24] Taylor, Simon. **Blockchain**: understanding the potential. July 2015.

[25] TECNOBLOG. **Como funciona o Blockchain-bitcoin**. Disponível em: <https://tecnoblog.net/227293/como-funciona-blockchain-bitcoin/>. Acesso em: 18/12/2019

[26] Tsudik, Gene. **Message authentication whih one-way hash functions**. ACM SIGCOMM Computer Comunication Review 225 (1992): 29-38.

[27] University of Cambridge. Cambridge Centre for Alternative Finance. https://www.cbeci.org/methodology/ . Cambridge Bitcoin Electricity Consumption Index

[28] Van Tilborg, Henk e Sushil Jajodia, eds. **Encyclopedia of cryptography and security**. Nova York: Springer Science & Bussiness Media. 2014.

[29] Wood, Gavin. Ethereum: **A secure decentralized generalized transaction ledger**, EIP-150 revision. Disponível em: http://gavwood.com/paper.pdf. Acessado em: