UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Guilherme Henrique de Araújo Santos

Uma biblioteca peso-leve para simulação de consenso em blockchain

Uberlândia, Brasil

2023

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Guilherme Henrique de Araújo Santos

Uma biblioteca peso-leve para simulação de consenso em blockchain

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Ivan da Silva Sendin

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2023

Guilherme Henrique de Araújo Santos

Uma biblioteca peso-leve para simulação de consenso em blockchain

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Trabalho aprovado. Uberlândia, Brasil, 03 de fevereiro de 2023:

Ivan da Silva Sendin
Orientador

Paulo Rodolfo da Silva Leite Coelho
Professor convidado

Renato Aparecido Pimentel da Silva
Professor convidado

Uberlândia, Brasil
2023

Dedico este trabalho à minha mãe Martha e meu pai Antônio que sempre estiveram ao meu lado nos bons e maus momentos, incentivando e me dando forças diariamente em todos os desafios da minha vida até hoje. Também o dedico à toda comunidade científica, a fim de que ela cresça cada dia mais com novos trabalhos e descobertas.

Agradecimentos

Agradeço primeiramente a Deus por me manter firme e de pé com saúde, sabedoria e forças para poder completar os desafios que tive durante minha trajetória até este momento.

Aos meus pais pelo apoio, por acreditarem em mim e sempre estarem ao meu lado transmitindo amor, forças, compaixão e empatia. Aos meus avós que não estão mais presentes aqui, em especial ao meu avô João Moura, que me ensinou sobre honestidade, respeito, humildade e perseverança.

A minha namorada Marília que me deu bastante suporte nessa jornada me trazendo muita felicidade, resiliência e calma em todas as horas.

A minha gatinha de estimação Shakira que me proporcionou várias risadas e momentos de fofura.

Ao meu orientador Ivan Sendin pela disposição, entusiasmo e apoio que foram fundamentais para a conclusão deste trabalho.

Aos meus amigos que cresceram comigo e a todos que me proporcionaram inúmeros momentos felizes.

Ao grupo de Programa de Educação Tutorial do curso Sistemas de Informação (PET-SI) pelos projetos desenvolvidos, por me ensinarem coisas incríveis, aumentarem meu amor pela ciência e por todas as amizades conquistadas durante os mais de 2 anos que estive por lá. Quero ressaltar um grande agradecimento aos tutores Wendel Melo e Paulo Henrique pela paciência, carinho e ajuda.

Aos professores do ensino fundamental, médio e da Faculdade de Computação UFU por todo o conhecimento que me proporcionaram ao decorrer da minha vida e aos professores Renato Pimentel e Paulo Coelho presentes nessa banca.

Todos vocês fizeram diferença e foram importantes para mim, muitíssimo obrigado!

"A educação é a arma mais poderosa que você pode usar para mudar o mundo."

– Nelson Mandela

Resumo

A notável ascensão das criptomoedas e do token não fungível (NFT) no mundo globalizado levaram pesquisadores do mundo todo a buscar o porquê da sua popularidade tão rápida e da fama de confiabilidade e segurança. Com transações que utilizam conceitos de imutabilidade e descentralização, foi possível moldar uma nova economia baseada em dados confiáveis e controlados pelos próprios usuários desse "novo sistema", em termos computacionais, uma rede composta por uma cadeia de blocos com informações imutáveis interligadas entre si, a chamada Blockchain. A partir da ideia de estudar e permitir que a comunidade científica possa explorar esse conteúdo de diversas formas, com diferentes hipóteses e trabalhos científicos sobre assuntos relacionados ao conteúdo apresentado, foi desenvolvida nesse trabalho uma biblioteca peso-leve em Python para simulação das operações em uma *blockchain* baseada no algoritmo de prova de trabalho (PoW). A biblioteca permite ao usuário criar mineradores, realizar tentativas de mineração, minerar novos blocos com PoW, obter consenso na propagação de novos blocos, ocorrência de *forks* (bifurcações, trifurcações), explorar a segurança da aplicação, verificar o histórico de mineradores e por fim gerar arquivos com informações relacionadas aos mineradores, *blockchain* e operações da rede.

Palavras-chave: Blockchain, Criptomoedas, Consenso, *Forks*, Prova de trabalho.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

NFT	Non-fungible token
PoW	Proof of work
IoT	Internet of things
P2P	Peer-to-peer
PoS	Proof of stake
PoET	Proof of elapsed time
SBFT	Simplified Byzantine Fault Tolerance
PoA	Proof of authority
CPU	Central processing unit
GPU	Graphics processing unit
ASIC	Application-specific integrated circuit
BTC	Bitcoin
SHA	Secure hash algorithms
NSA	National Security Agency
NIST	National Institute of Standards and Technology
CSV	Comma-separated values
PNG	Portable Network Graphics
PC	Poder computacional
BM	Blocos minerados
PCR	Poder computacional da rede
QF	Quantidade de fraudadores
PPCCF	Porcentagem do poder computacional de cada fraudador
PCMF	Poder computacional do minerador fraudador

Sumário

1 Introdução

Nos primórdios da computação o armazenamento de dados foi tratado de forma simplificada em relação aos métodos utilizados atualmente (HAFF; HENRY, 2017), a falta de hardwares especializados e o custo para armazenar informações em grande escala era gigantesco, o que fazia programadores e pesquisadores procurarem estratégias que visavam otimizar os dados da forma mais concisa possível, já que cada byte de memória gasto era uma maior despesa financeira. Essa maneira de pensar, que visava a economia de armazenamento, levou a computação a sofrer com alguns problemas. O mais famoso deles foi o "bug do milênio" (SOCIETY, 2022), em que, até a década de 90, os anos eram representados apenas com dois dígitos nos sistemas computacionais, por exemplo, 1999 era apenas 99 e na virada de milênio, as datas voltariam para 00, o que representaria 1900 e não 2000, assim alguns sistemas poderiam entrar em colapso. Desafios como este fazem a tecnologia evoluir e a partir daí desenvolvedores e pesquisadores do mundo todo começaram uma luta com o objetivo de melhorar hardwares e softwares, por intermédio de maneiras mais evoluídas e otimizadas de codificação, novas estratégias para diversificar o armazenamento, memórias computacionais com maior proporção e o início do conceito de computação em nuvem (cloud computing) (RANGER, 2022).

A chegada da computação em nuvem nos anos 2000 foi um dos marcos mais importantes para a tecnologia da informação que até então se preocupava em adquirir servidores e data centers de forma proprietária, a fim de garantir a disponibilidade de seus serviços para os clientes. A mudança desse paradigma foi um desafio, pois muitas empresas tinham receio de confiar o armazenamento dos dados de seus clientes em terceiros, porém ao longo dos anos essa tecnologia se popularizou e se tornou mais forte no mercado, o que possibilitou o fácil manejo de armazenamento (RANGER, 2022).

Com essa melhora em hardware e a disponibilidade de recursos, os problemas anteriores não são mais um impasse. Surgiram diferentes maneiras para o armazenamento de informações, destaca-se uma em relação as outras pela sua capacidade de revolucionar a roda da economia com criptomoedas (ANTONOPOULOS, 2017; SERHACK, 2018) e soluções robustas que entregam maior segurança, imutabilidade e, como principal fator, a descentralização dos dados de um único responsável para todos os envolvidos, a atualmente conhecida *Blockchain*.

1.1 Visão Geral

O primeiro registro oficial de uma *blockchain* surge por meio do artigo (NAKAMOTO, 2008). Desde seu lançamento, essa *blockchain* é utilizada para organização e distribuição da criptomoeda Bitcoin, que se tornou a criptomoeda mais popular do mundo atualmente e desde sua criação revoluciona o mercado financeiro constantemente, em especial pelo seu conceito de utilizar uma cadeia de blocos para armazenar suas informações de transações, além de trazer uma visão alternativa para banco de dados distribuídos (ANTONOPOULOS, 2017). A tecnologia de *blockchain* também permite a duplicidade de uso em redes públicas e privadas (chamadas de redes permissionadas), a qual é uma inovação que mudou o mundo ao possibilitar casos de uso em diferentes ambientes, como criptomoedas, varejo, internet das coisas (IoT), logística, agronegócio, tokens não fungíveis (NFT's), finanças e mobilidade urbana.

Por meio da utilização de *blockchains*, inúmeras criptomoedas foram criadas, por exemplo, tem-se o Bitcoin, Litecoin, Ethereum e o Monero. De forma sucinta, a *blockchain* é uma estrutura de blocos que faz o papel de um banco de dados par a par (P2P) distribuído. Na prática, cada nó envolvido na rede possui o mesmo conjunto de informações armazenadas e disputam entre si a mineração de um novo bloco. O primeiro nó a minerar de forma válida tem o privilégio de inserir seu bloco na cadeia (geralmente, o bloco é adicionado na *blockchain* de maior altura) e a atualização é notificada aos demais mineradores, que ao aceitarem que um novo bloco foi inserido, entram em consenso e iniciam o processo de mineração para o próximo bloco. Dessa forma, a *blockchain* torna-se um livro razão que guarda informações imutáveis, acessíveis e descentralizadas (ANTONOPOULOS, 2017).

Para que a rede entre em consenso e a nova informação inserida no livro razão seja incontestável, é necessário que ocorra uma validação desses novos dados por todos os nós existentes na rede, por meio da utilização do conceito de consenso, uma forma racional de uma rede formada por pessoas com interesses próprios concordarem sobre uma determinada informação. A maneira mais popular para a obtenção de consenso entre cada envolvido na rede é a *Proof of Work* (PoW), em português, prova de trabalho. Em paralelo, existem diferentes maneiras para se obter consenso que podem ser abordadas em futuros projetos de pesquisa baseados neste trabalho, como *Proof of Stake* (PoS), *Proof of Elapsed Time* (PoET), *State of the art Byzantine Fault Tolerant* (SBFT), *Proof of Authority* (PoA), entre outras que são bem analisadas e explicadas no artigo (REBELLO et al., 2019) e no site criado por (LAMOUNIER, 2018).

O mecanismo sobre qual este trabalho foi desenvolvido é o mecanismo de PoW, que consiste em utilizar recursos computacionais como a unidade central de processamento (CPU), a unidade de processamento visual (GPU) ou um circuito integrado de aplicação específica (ASIC) para realizar uma determinada tarefa. O primeiro sistema computacional a realizar a tarefa adequadamente estará:

a) Validando as transações feitas pela criptomoeda;

b) Obtendo uma recompensa financeira para si, definida pelo próprio protocolo da criptomoeda.

O processo de realização da prova de trabalho é chamado de mineração devido a sua semelhança com uma mineração real, em que se efetua um trabalho com a possível expectativa de ganho proporcional ao esforço realizado.

1.2 Objetivos

O principal objetivo deste trabalho é estudar e detalhar os principais conceitos sobre *blockchain*, segurança, imutabilidade e apresentar o desenvolvimento de uma biblioteca peso-leve de código aberto em Python, que permita realizar a simulação das operações recorrentes em uma *blockchain* de forma similar à realidade. O objetivo geral pode ser subdividido da seguinte forma:

a) Visão geral sobre criptomoedas ao explicar seu histórico e a influência que o Bitcoin exerce como moeda virtual;

b) Uma perspectiva geral dos conceitos de *blockchain*;

c) Aspectos de segurança no mecanismo de consenso: *cryptojacking*, mineração egoísta e ataque 51%;

d) Estudo e simulação: prova de trabalho, mecanismo de consenso, *forks* (bifurcações, trifurcações, etc.) e possíveis vulnerabilidades.

1.3 Justificativas

Incentivar pesquisadores e a comunidade científica a explorar as atividades em *blockchain* para que essa tecnologia venha a ser utilizada em suas diferentes variantes, o que possibilita casos de uso em todas as camadas de organização de dados de uma sociedade, dessa maneira, a inovação tecnológica do mundo pode ser inserida dentro das universidades brasileiras. Possibilitar e criar uma ponte desse conhecimento sobre *blockchain* no Brasil é a principal justificativa de atuação e criação deste trabalho. O intuito é que esta pesquisa leve a comunidade científica e tecnológica a compartilhar e incentivar a busca pelo conhecimento ao utilizar a biblioteca criada para realizar diferentes estudos e implementações de novos métodos que possam colaborar com demais trabalhos.

1.4 Metodologia

A divisão deste trabalho se define, inicialmente, em realizar uma pesquisa aprofundada das bibliografias existentes, com o intuito de coletar o máximo de dados confiáveis para estudar e entender a *blockchain* e os principais desafios do conceito de prova de trabalho. Após este estudo, o foco é direcionado a elaborar um material que explique de forma clara e coesa os ideais de uma *blockchain*, como o processo de mineração envolvendo *hash*, obtenção de consenso entre cada nó presente na rede, ocorrência de *forks* (momento em que duas ou mais *blockchains* estão transitando na rede com informações verdadeiras) e vulnerabilidades encontradas no mecanismo de consenso.

Após a realização destes procedimentos, começa-se a colocá-los em prática através de uma prova de conceito, que se dá no desenvolvimento de uma biblioteca de código aberto em Python, para possibilitar que cada conceito estudado possa ser explorado, trazendo à comunidade open source brasileira e mundial uma melhor visão sobre *blockchain* na prática.

2 Fundamentação teórica

Neste capítulo serão explicados os conceitos fundamentais sobre o estado da arte do projeto, com a abordagem dos assuntos ao utilizar imagens e analogias para exemplificar de forma simples e intuitiva. Todas as imagens foram criadas pelo autor para facilitar a introdução do conteúdo de maneira correta e próxima das funcionalidades existentes na biblioteca criada.

2.1 Criptomoedas

As criptomoedas são ativos digitais que atuam como uma forma de dinheiro, não são palpáveis e não existem fisicamente como cédulas, porém, assim como as moedas tradicionais (dólar, euro e real) elas são criadas e distribuídas para seus usuários, podem circular de forma centralizada ou descentralizada, que é o destaque deste tipo de sistema. O interesse na descentralização dos dados faz crescer o desejo por criptomoedas neste estilo, pois oferecem um maior controle dos usuários sobre seus próprios bens a partir do momento que evita-se a intermediação de uma figura central (NAKAMOTO, 2008).

Uma moeda centralizada é aquela emitida por um órgão central e controlada por um pequeno grupo de pessoas. Nas moedas tradicionais tem-se como exemplos os bancos que fazem a gestão do dinheiro. Dessa maneira, o usuário precisa confiar em um terceiro, o banco, para realizar sua transação e não sabe-se a quantidade de dinheiro que o banco diz ter é a real situação (SERHACK, 2018). Um cliente não pode realizar a auditoria do seu banco para verificar essa questão. Uma moeda centralizada só é valorizada por meio da confiança nos administradores desse dinheiro.

Em uma moeda descentralizada utiliza-se da tecnologia de *blockchain* para emitir e gerenciar as transações que ocorrem na sua rede descentralizada, valendo-se de diferentes sistemas para a obtenção de consenso entre os participantes da rede. Geralmente, as redes descentralizadas são dispostas no formato P2P que não exige um servidor centralizado, mas sim várias máquinas conectadas entre si. Assim, ao ocorrer uma perda de conexão a rede não cai e continua estável para os demais nós.

A rede *blockchain* utiliza-se da criptografia para manter a segurança contra fraudes, o que traz um sistema confiável, seguro e imutável. Se um bloco de informações é inserido nela, este não poderá ser alterado, pois se ele for modificado, seu *hash* também seria alterado e isso invalidaria todos os blocos que foram adicionados após o bloco adulterado. Cada bloco é interconectado com o *hash* do bloco anterior e para a inserção de um novo bloco é feito todo um processo de mineração que busca resolver de maneira matemática o

encapsulamento dos dados de forma criptografada no intuito de encontrar um *hash* válido (NAKAMOTO, 2008).

Essa segurança que muitas criptomoedas trazem consigo em seus sistemas fez a popularidade delas aumentarem gradativamente, ao fazer com que as pessoas procurem compreender mais essa nova forma de economia. Porém, as criptomoedas descentralizadas possuem a fama de serem instáveis quando se fala do seu valor de mercado (GREENBERG, 2011), ao poder fazer com que alguém se torne repentinamente milionário do dia para a noite ou perder todas as suas economias. É um sistema seguro de transações, mas não quando se trata de investimentos e esse é o principal risco ao adquirir criptomoedas.

2.1.1 Bitcoin

O Bitcoin é uma criptomoeda que foi apresentada ao mundo em 2008, conhecida como o primeiro caso de sucesso a vir a público, se trata de um sistema baseado em *block-chain*. Sua ascensão é devido a ruptura do ideal de um sistema tradicional de economia, que envolve o auxílio de uma terceira parte confiável para realizar todos trâmites de uma transação financeira, os bancos.

As transações bancárias, realizadas por meio de confiança, não garantem aos seus usuários que toda transferência feita ocorre de maneira legítima e honesta, de forma que nenhum usuário pode realizar a auditoria em seus bancos. Segundo o autor (SERHACK, 2018, Mastering Monero: The future of private transactions, p. 15) *"uma vez que um ator nefasto ou os bancos podem 'criar' dinheiro editando de forma fraudulenta os saldos contábeis ou o banco de dados de transações."*

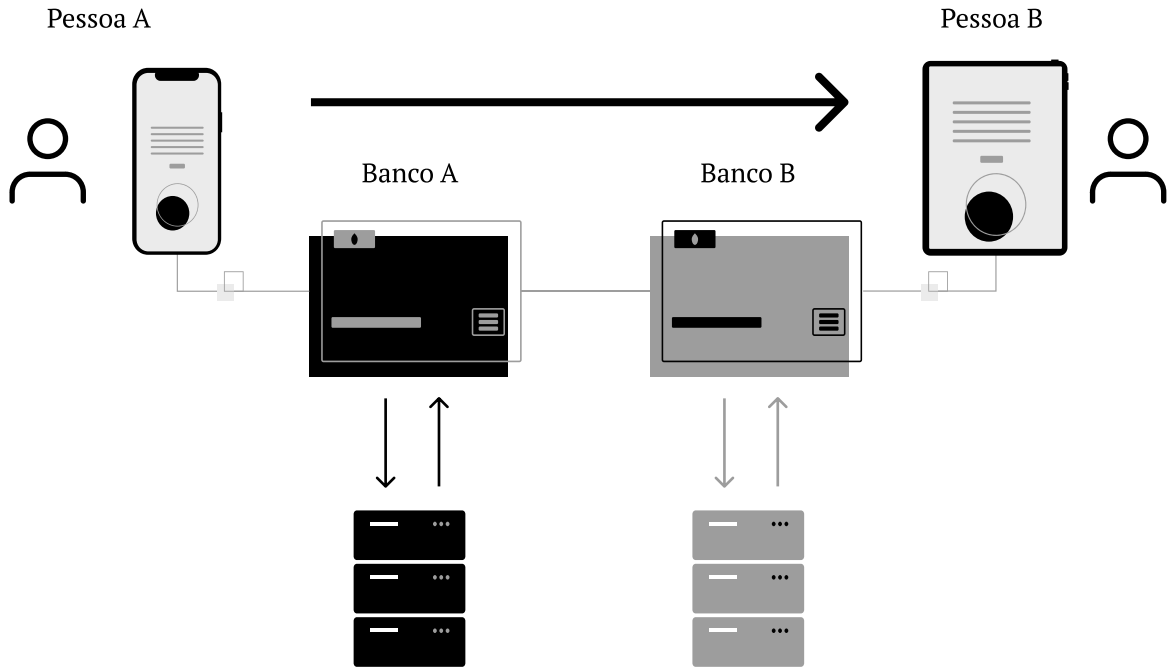

Figura 1 – Transação bancária tradicional.

Fonte: próprio autor, 2023.

Na Figura 1 pode-se analisar que se a **pessoa A** quiser realizar uma transação bancária com a **pessoa B**, ela precisa, necessariamente, confiar nos intermediários **Banco A** e **Banco B** que farão apenas uma alteração em seus bancos de dados para confirmar a transação.

Na rede compartilhada do Bitcoin essa mesma transação ocorreria de forma diferente, a identidade das pessoas envolvidas na transação seria anônima e a partir daí o indivíduo não faz parte da transação com seus dados, ele possui uma carteira Bitcoin e por meio dessa carteira ele realiza suas transações. Por exemplo, com o endereço da carteira para qual se quer realizar uma transferência, basta inserir a quantidade de Bitcoins que quer transferir, a rede em seu processo de mineração vai armazenar a transação em novos blocos inseridos na *blockchain* e a transferência está feita, a própria rede é responsável por validar as transações (ANTONOPOULOS, 2017).

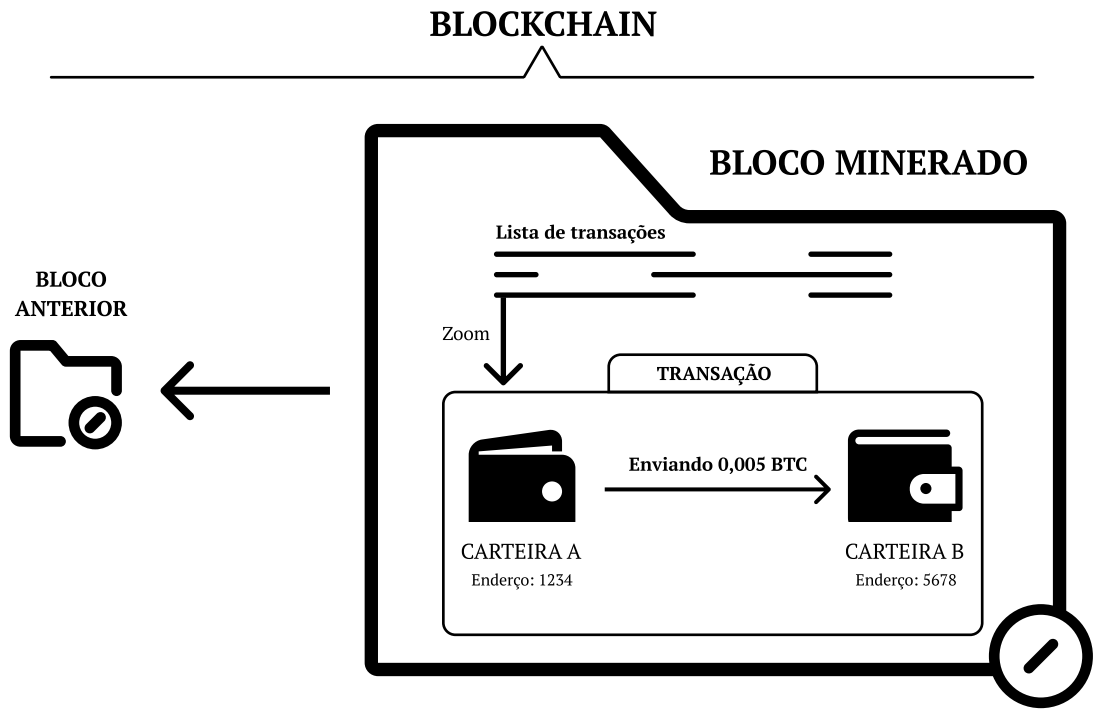

Figura 2 – Transação realizada por meio do Bitcoin.

Fonte: próprio autor, 2023.

Na Figura 2, a **carteira A** disposta do endereço da **carteira B** solicita o envio de 0,005BTC. É iniciado o processo de mineração em que as informações são inseridas em um bloco contendo várias transações, ele é validado e inserido na *blockchain*. Ao observar as transações do bloco com um zoom, é possível analisar a transação da carteira A de endereço 1234 transferindo 0,005BTC para a carteira B de endereço 5678.

2.1.2 Blockchain

A *Blockchain* é uma forma de armazenamento de dados que utiliza uma rede P2P para validar suas informações, com o intuito de trazer segurança e confiabilidade. Ela é uma cadeia de blocos com informações referentes a cada transação dos nós envolvidos na rede.

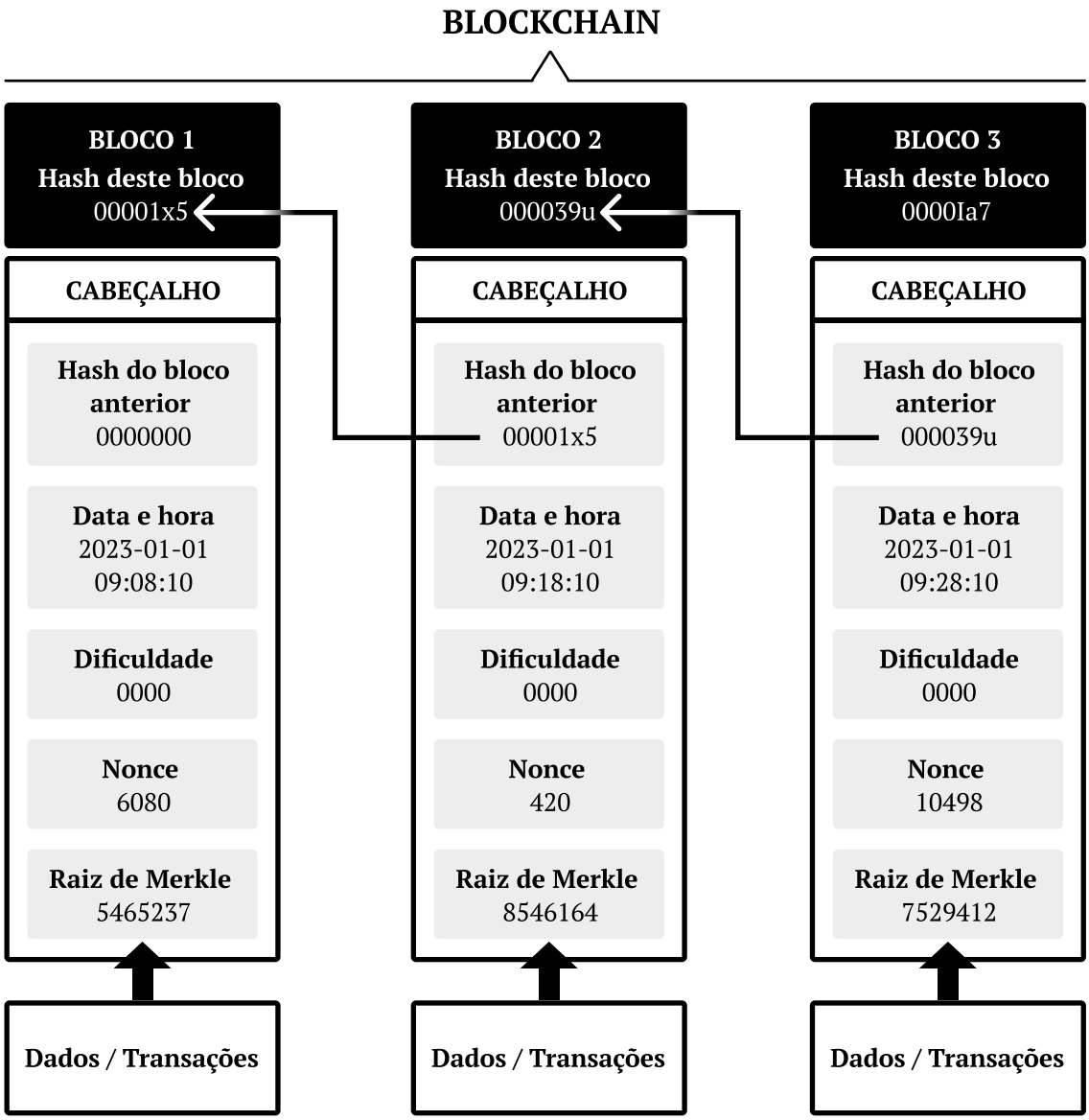

Figura 3 – Representação da estrutura *Blockchain*.

Fonte: próprio autor, 2023.

A Figura 3 representa uma *blockchain* com seu conjunto de blocos interligados entre si por meio da referência ao *hash* do bloco anterior. O bloco inicial possui o *hash* do bloco anterior com todos os algarismos em zero pois ele não possui um bloco anterior. Todo *hash* de um bloco é iniciado em 0000, uma maneira de obter consenso. A Tabela 1 detalha cada campo de informação descrito na imagem.

Tabela 1 – Propriedades de cada bloco que é inserido na *blockchain*. Detalhamento dos campos exibidos em cada bloco da imagem 3.

Nome	**Significado**
Número do bloco	Representação numérica da altura em que o bloco foi inserido na *blockchain*.
Hash deste bloco	Este *hash* não está explicitamente a mostra na *blockchain* como a figura induz, ele é gerado a partir das informações contidas no cabeçalho de um bloco e é utilizado somente para que o próximo bloco possa referenciá-lo.
Hash do bloco anterior	Identificação do bloco anterior, é essa ligação que permite um bloco identificar seu predecessor na cadeia de blocos, é por meio dessa referência que a *blockchain* é construída e dá origem ao livro razão interconectado.
Data e hora	Momento registrado (data e horário) em que este bloco foi criado.
Dificuldade	A dificuldade imposta pelo algoritmo de PoW, pode ser relacionado ao prefixo do *hash* obtido durante a prova de trabalho. É uma métrica consensual entre os participantes da rede.
Nonce	Número aleatório utilizado para validar um bloco no processo de mineração, geralmente relacionado a quantidade de tentativas realizadas para que o bloco consiga entrar em consenso com os demais.
Raiz de Merkle	Um *hash* referente à raiz da árvore de Merkle dos dados/transações contidas no bloco.
Dados/Transações	Informações das transações ocorridas na rede desde o último bloco inserido na *blockchain*.

Fonte: próprio autor, 2023.

2.1.2.1 Principais características

Cada *blockchain* possui sua particularidade definida de acordo com seus protocolos e a forma como vai ser dimensionada (distribuída ou centralizada). Embasado em uma rede *blockchain* descentralizada, pode-se observar as seguintes características (GUPTA, 2018):

a) Uma *blockchain* é desenvolvida de acordo com o protocolo que melhor se encaixa com a solução que deseja atender. Um protocolo popularmente utilizado é o PoW;

b) Uma *blockchain* é uma cadeia de blocos interconectados;

c) O primeiro bloco inserido em uma *blockchain* é chamado de bloco gênesis, pois ele é o primogênito de todos os blocos posteriores, onde dá-se o início à adoção do protocolo escolhido;

d) Para a inserção de um novo bloco, ele, necessariamente, precisa referenciar o bloco anterior para dar sequência na cadeia de blocos e tornando-se fiel a rede *blockchain*;

e) Para a inserção de blocos atualizados é necessária uma análise individual sobre cada bloco novo recebido, tornando-o válido ou inválido, por meio de um consenso coletivo entre os usuários da rede *blockchain*.

2.1.3 Mineração com PoW

Um processo de mineração que utiliza o mecanismo de obtenção de consenso PoW busca encontrar uma evidência matemática válida para a inserção de um novo bloco na *blockchain* vigente. Para que isso aconteça, uma série de operações são feitas para garantirem a veracidade das informações, a segurança e imutabilidade dos dados. O resultado é obtido em formato de um *hash* com 256 bits de acordo com a dificuldade imposta na rede. A dificuldade se dá pelo prefixo desse *hash*, quanto maior a quantidade de bits que o prefixo tiver, maior será a dificuldade do trabalho para conseguir uma prova válida (CHICARINO, 2019). Quanto mais difícil um problema, maior será o esforço computacional gasto para resolvê-lo e em alguns sistemas de criptomoedas este esforço é recompensado. No Bitcoin, por exemplo, um minerador recebe uma remuneração da própria criptomoeda por ter despendido de força computacional para poder minerar novos blocos.

A mineração com PoW será dividida em 7 subseções que irão detalhar o processo por completo, essas subdivisões tem o intuito de direcionar de forma específica cada conceito abordado.

2.1.3.1 Camada de dados

Ao iniciar o processo de mineração de um novo bloco, uma quantidade de transações válidas predeterminada pelo consenso da rede são alocadas neste bloco. Podem ser armazenadas em formato de array, dicionários, mas a maneira mais utilizada são as árvores de Merkle (MERKLE, 1979), pois facilitam a busca das transações devido a sua maneira de organização das informações, em que os dados estão nas folhas e cada nó representa um valor de *hash* para suas folhas.

2.1.3.2 Hash válido

Após os dados serem coletados é gerado um *hash* unidirecional a partir das informações contidas no processo de mineração do bloco. Para fazer isso„ é escolhida uma maneira de criptografia, na maioria dos casos, o SHA-256, um algoritmo de código aberto criado pela NSA e NIST, com o intuito de gerar uma sequência de bits exclusiva para um conjunto de dados (PENARD; WERKHOVEN, 2008). O algoritmo não permite que o dado criptografado seja descoberto a partir do *hash* gerado.

O *hash* resultante do algoritmo é sempre diferente para cada sequência de bits que recebe como entrada dos dados e possui um tamanho fixo de 256 bits após o processo de *hashing*, seja para codificar uma letra ou um livro inteiro, dessa maneira a privacidade do conteúdo é garantida (HAWKES; PADDON; ROSE, 2004).

O *hash* é utilizado como um identificador único de um bloco, ele é a resposta encontrada após todo o processo de mineração, ou seja, a prova do trabalho realizado. O próximo tópico conta com o detalhamento deste processo de trabalho.

2.1.3.3 Prova do trabalho

O processo da prova de trabalho é uma das partes mais importantes na mineração, pois como visto no tópico anterior, o *hash* é um identificador único, cada bloco inserido possui um *hash* diferente. Porém, a construção de um *hash* válido exige um processamento a ser desempenhado, inicialmente os dados são criptografados com uma função criptográfica de *hash* gerando um valor de *hash* inválido, pois não possui o prefixo pré estabelecido pela rede. A Figura 4 exemplifica essa situação.

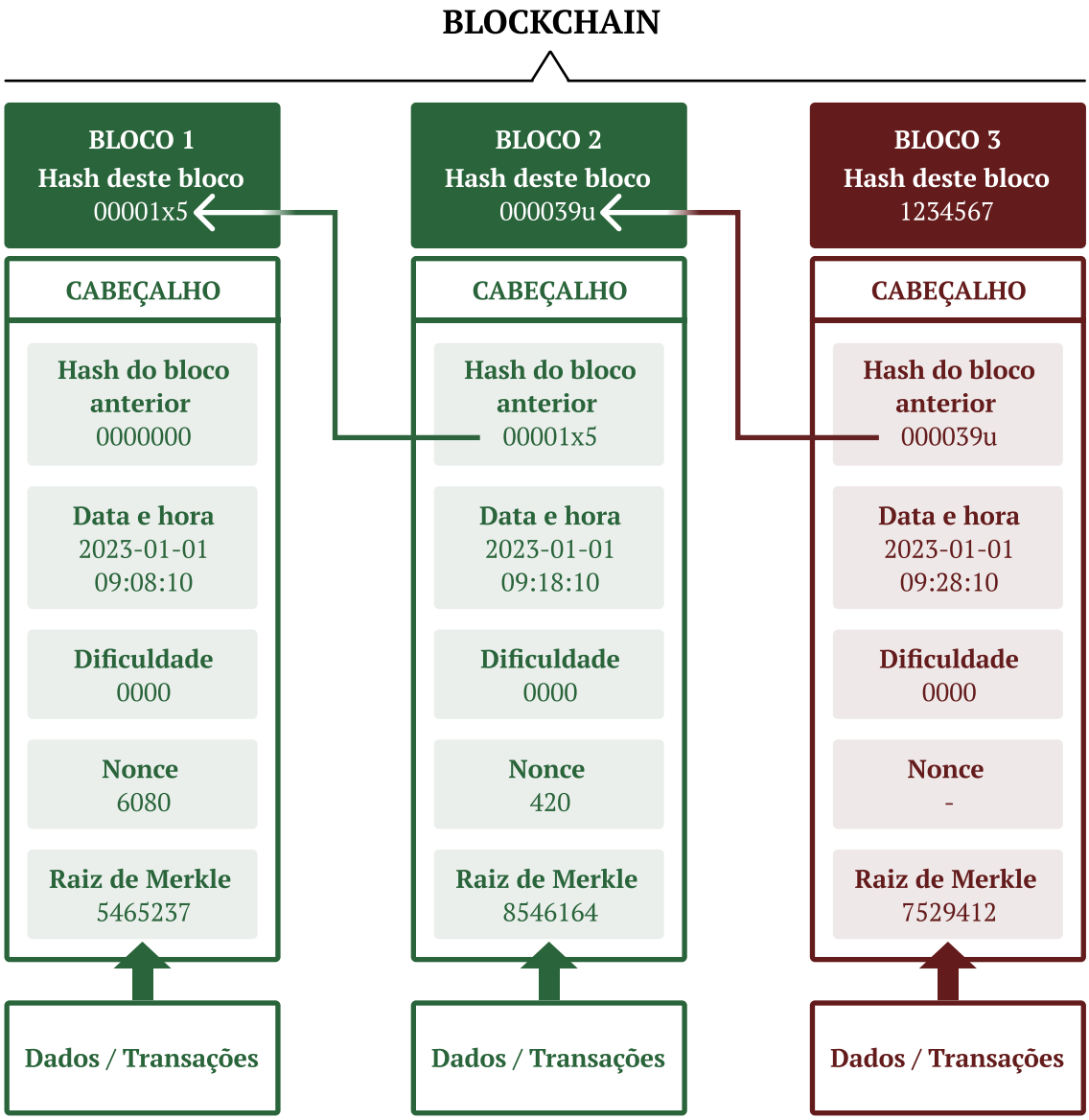

Figura 4 – Demonstração de um bloco que seria considerado inválido pois seu *hash* ainda não obteve a prova de trabalho.

Fonte: próprio autor, 2023.

Na Figura 4, pode-se observar que todos os blocos inseridos anteriormente na *blockchain* se iniciam com um prefixo 0000 e o *hash* gerado para o novo bloco não se inicia em 0000 e sim em 1234, quando isso acontece, é dito que o bloco ainda não obteve a prova de trabalho. Para que esse bloco possa obtê-la, é realizado o processo em que se busca um *nonce*[1] que valide um *hash* com prefixo 0000, sem perda e sem alteração do conteúdo de um bloco (CHICARINO, 2019).

[1] Número aleatório escolhido uma vez, geralmente relacionado à quantidade de tentativas realizadas para que possa ser encontrado um *hash* válido para o bloco.

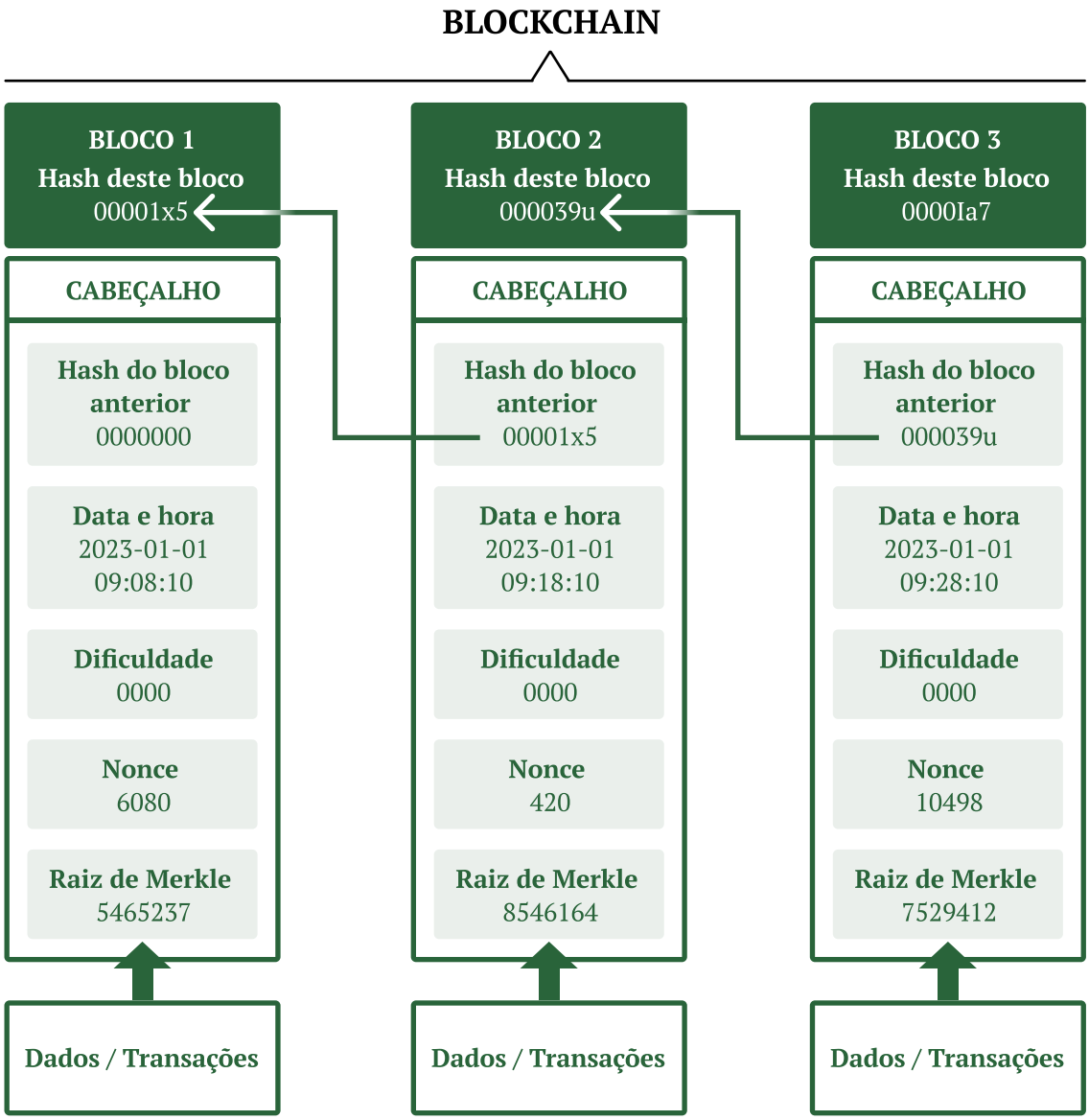

Figura 5 – Transição de um bloco inválido para um bloco válido após a obtenção da prova de trabalho.

Fonte: próprio autor, 2023.

Na Figura 5, pode-se observar a mudança nos valores referentes ao *nonce* e ao *hash* do bloco 3, pois após encontrado o *nonce*, é possível chegar ao *hash* que representa o bloco. Desta forma tem-se um *hash* válido para o bloco e pode-se dizer que este bloco está pronto para obter o consenso da rede pois é valido.

2.1.3.4 Controle de dificuldade

Para *blockchains* baseadas em PoW, com ênfase em recompensar os gastos ocorridos no processo de mineração, é aconselhável que seja definido em consenso pela rede, um intervalo de tempo ideal para que encontre-se a prova de trabalho de cada bloco, esse

período de tempo deve levar em consideração o poder computacional atual dos membros da rede e ser constantemente ajustado a medida que a quantidade de mineradores cresce e a facilidade de minerar um novo bloco aumenta. Cada vez que o intervalo de construção dos blocos fica menor, é necessário que se aumente a dificuldade para encontrar a prova de trabalho e o contrário também deve ocorrer, se o tempo decorrido entre a propagação de um bloco e outro aumentar, essa dificuldade deve ser reduzida. Este controle de dificuldade é fundamental para existência de muitos sistemas, como o Bitcoin, pois este método sustenta a frequência da emissão da moeda e a velocidade da liquidação das transações (ANTONOPOULOS, 2017).

2.1.3.5 Obtenção de consenso

Após o processo de obtenção da prova de trabalho do novo bloco, o nó da rede que conseguiu realizar a mineração, atualizará sua *blockchain* e notificará aos vizinhos de que foi inserido um novo bloco, dessa forma passa a existir uma nova *blockchain* de maior altura na rede. Os vizinhos que receberam o novo bloco do minerador, validarão e aprovarão esse bloco, após este processo, eles se tornam aptos a divulgar essa atualização para seus vizinhos, e assim por diante, no intuito de todos os nós da rede estarem informados sobre a *blockchain* mais atualizada. A Figura 6 evidencia este processo, mineradores propagam sua *blockchain* atualizada para seus vizinhos, esses que irão processá-la e se válida (como é o caso do exemplo), propagarão para os seus vizinhos e consecutivamente, esse procedimento é realizado até essa *blockchain* ser espalhada pela rede inteira.

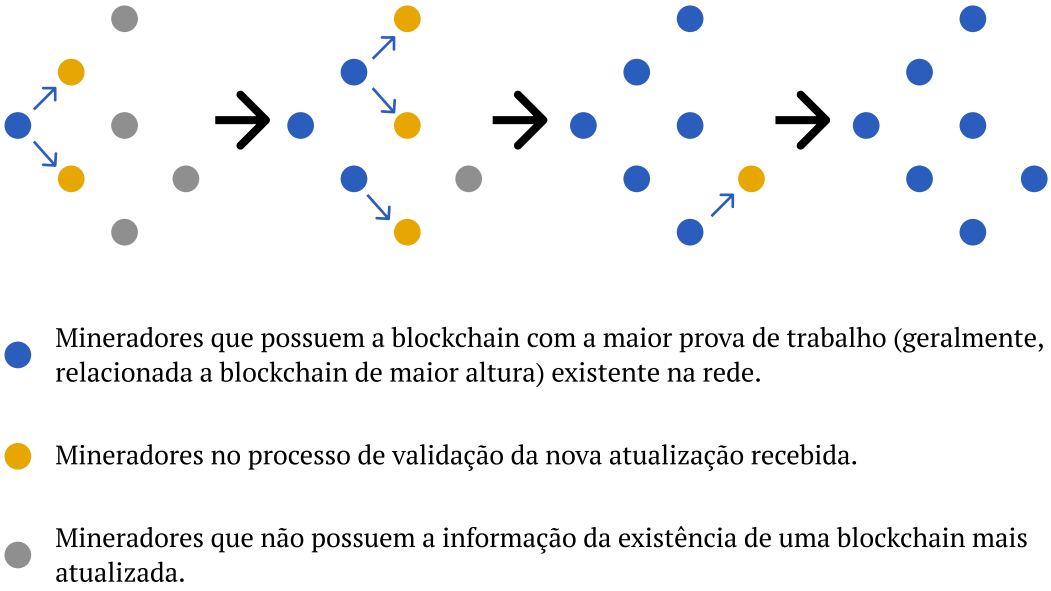

Figura 6 – Representação da propagação de uma atualização na *blockchain*

Fonte: próprio autor, 2023.

2.1.3.6 Forks

Este tópico visa explicar o conceito de bifurcações, trifurcações e outras maneiras de divisões da *blockchain* por meio da nomenclatura original do termo em inglês (*fork*). Um *fork* possui equivalência aos termos bifurcações e trifurcações, porém engloba todos os tipo de divisões ou subdivisões para mais de uma *blockchain* válida na rede. Portanto, para evitar se referir às trifurcações com a utilização do termo bifurcações, este trabalho usará a terminologia *forks*.

Em uma rede *blockchain* que possui diversos nós, pode acontecer de dois ou mais mineradores realizarem a mineração de um novo bloco, válido, ao mesmo tempo, a partir deste momento duas ou mais *blockchains* são consideradas válidas na rede. Esta ocorrência é chamada de *fork* (bifurcações, trifurcações, etc.) e a Figura 7 retrata este processo.

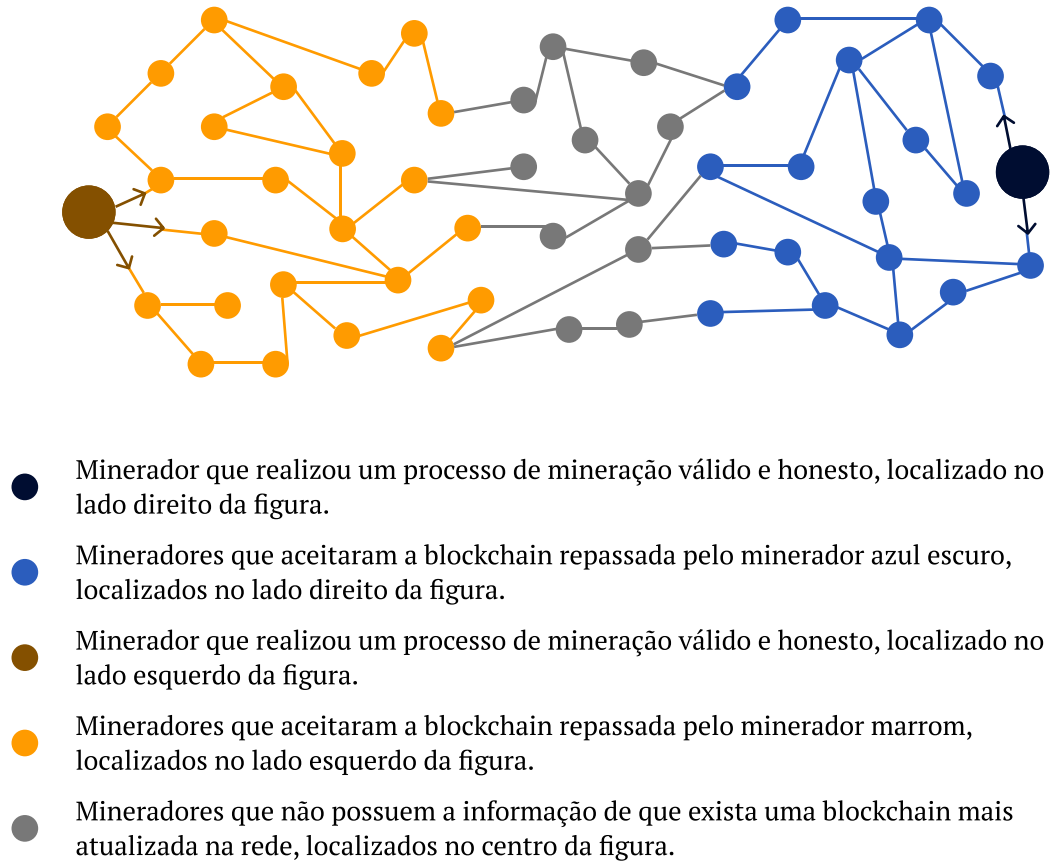

Figura 7 – Representação de *forks* em uma *blockchain*.

Fonte: próprio autor, 2023.

Observa-se na Figura 7, dois mineradores mineraram um novo bloco válido ao mesmo tempo, neste momento, esses mineradores possuem uma atualização válida na *blockchain* e estão aptos para enviarem suas atualizações aos vizinhos mais próximos. Portanto, ao iniciarem o repasse da atualização para seus vizinhos, a rede se dividirá em

duas, pois uma parte dos nós existentes irá aceitar a atualização do minerador azul e outra parte irá aceitar a atualização do minerador laranja.

2.1.3.7 Resolução de forks

Os *forks* existentes em uma rede *blockchain* são solucionados após algum minerador, de forma unitária, minerar um novo bloco válido e repassar a atualização para o restante dos nós da rede. Esse minerador, propagará sua atualização para a rede forqueada adotar uma *blockchain* única, a que ele propagou. Um *fork* é desfeito ao espalhar uma nova atualização, pois o minerador que resolveu o dilema de duas ou mais *blockchains* válidas, adotou apenas uma, a que se realiza a referência de seu novo bloco. Assim, quem receber sua *blockchain* atualizada, abandonará qualquer variante que contenha blocos divergentes da atualização, dessa forma, a unicidade é novamente estabelecida na *blockchain*. A Figura 8 retrata este processo.

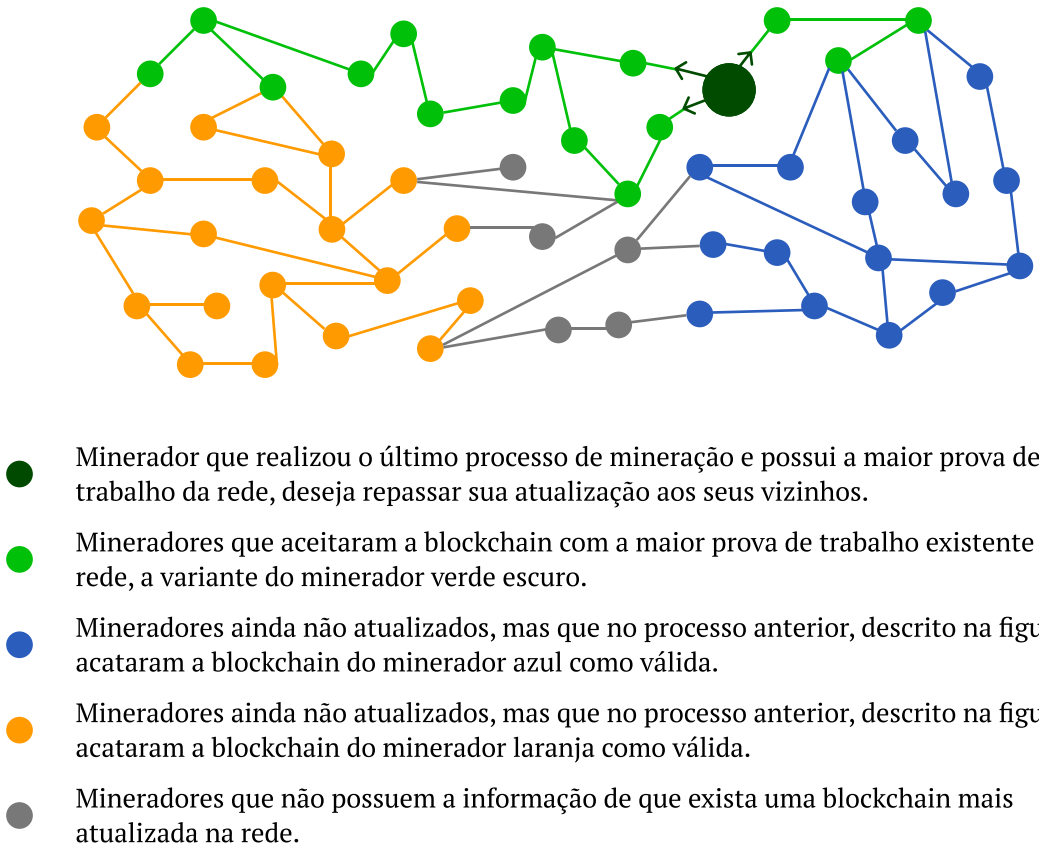

Figura 8 – Desfeita de *forks* na *blockchain*.

Fonte: próprio autor, 2023.

Supõe-se na Figura 8, que o minerador de verde escuro tinha aceito anteriormente, a variante da *blockchain* minerada pelo minerador azul escuro. Ao minerar e propagar

um novo bloco para seus vizinhos, o minerador verde escuro automaticamente, repassa de forma única e sem concorrência, sua atualização. Todos os participantes da rede ao receberam a *blockchain* com o novo bloco, irão acatar que o bloco anterior foi resolvido pelo minerador azul escuro, mesmo que anteriormente, alguns nós tenham admitido que a variante laranja foi válida. Isso acontece devido ao fator da cadeia mais longa, o minerador verde escuro minerou seu bloco ao partir do *hash* do bloco minerado pelo minerador azul, portanto como ele minerou unicamente, sua *blockchain* possui a maior altura existente na rede.

2.1.4 Aspectos de segurança no mecanismo de consenso

Por mais que uma solução baseada em *blockchain* traga um fator de segurança, não significa que isso não pode e não foi explorado ao decorrer dos seus anos de existência. Em busca de novas vulnerabilidades ou fraquezas dos usuários que fazem parte de um sistema baseado em *blockchain*, agentes maliciosos encontraram algumas maneiras de burlar processos para obterem vantagens para si. Nos próximos subtópicos, pode-se observar algumas formas de violação sobre a *blockchain* e práticas desonestas para realização de minerações.

2.1.4.1 Malware de cryptojacking

O processo de mineração demanda o gasto da utilização de recursos computacionais, energia elétrica e tempo, com o objetivo de evitar esses prejuízos, agentes maliciosos, utilizam da prática de *cryptojacking*. Por meio dela conseguem os meios necessários para minerarem criptomoedas ao utilizar equipamentos de pessoas inocentes, que não percebem que foram hackeadas e roubadas (TEKINER et al., 2021). A prática consiste em infectar um aparelho com algum script manipulado e não deixar que o usuário perceba a execução do mesmo em sua máquina, dessa forma, o intermediário com más intenções obterá um maior poder computacional e economizará em despesas próprias.

Este tipo de malware é muito utilizado para mineração da criptomoeda Monero (SECRETARIAT, 2020), cujas inserções de trechos de códigos maliciosos em sites são utilizadas para capturar o navegador da vítima enquanto ela estiver em uso, isto é uma prática recorrente. No geral, o *cryptojacking* é muito utilizado em sites que fazem transmissão de filmes piratas.

Essa prática também pode ocorrer de maneira legalizada, desde que o site informe o usuário sobre os meios utilizados para um determinado fim e esse cliente aceite ceder seus próprios recursos como forma de pagamento por acessar determinado conteúdo.

2.1.4.2 Mineração egoísta

O processo de mineração tradicional tem como valor fundamental uma rede descentralizada e pública, que dá certa liberdade para a rede de mineradores, porém, alguns indivíduos dessa rede se organizam em grupos para realizarem minerações de forma privada e egoísta, ao compartilhar atualizações somente entre eles. O intuito dessa prática é criar uma bifurcação da *blockchain* pública, para minerar novos blocos que apenas esses nós associados terão conhecimento, em algum momento essa *blockchain* privada pode tornar-se mais relevante do que a pública, consequentemente, os mineradores envolvidos revelam seus blocos e repassam para a rede, que aceitará as atualizações, devido ao fato da *blockchain* privada ter uma maior prova de trabalho, o que torna a mineração egoísta algo lucrativo para os mineradores (GOBEL et al., 2015).

2.1.4.3 Ataque de 51%

O ataque de 51% significa dominar mais de 50% dos meios necessários para realizar minerações na rede. Para obter esse tipo de controle, grupos de mineradores podem atuar em conluio, no intuito de fortalecerem seus recursos computacionais. Outra alternativa, é o investimento individual de equipamentos tão poderosos capazes de dominar mais de 50% do poder computacional da rede *blockchain*. A segunda alternativa, em muitos casos, é considerada inviável, como no Bitcoin.

Ao possuir mais da metade da capacidade computacional da rede de mineradores, torna-se possível controlar todas as operações que podem ocorrer em uma *blockchain*, dessa forma, o sistema deixa de ser confiável e converte-se para algo manipulável e centralizado. Se alguma entidade conseguir completar um ataque dessa amplitude, transações poderiam ser invalidadas, adulteradas e passadas como verdadeiras na *blockchain* (ACADEMY, 2018), o que significaria voltar para o processo antigo de confiança em terceiros para realização de transações monetárias.

3 Trabalhos correlatos

Os trabalhos correlatos serão divididos em dois principais subtópicos, a seção 3.1 visa mostrar soluções que mesmo não relacionadas a uma pesquisa científica, auxiliam estudantes e entusiastas a entenderem mais sobre os conceitos de *blockchain* e criptomoedas. A seção 3.2 tem o papel de evidenciar pesquisas científicas similares ao projeto que já foram publicadas.

3.1 Tutoriais sobre blockchain

Esta seção tem o objetivo de mostrar trabalhos que são de grande importância para a sociedade e pesquisas científicas, pois ensinam e auxiliam toda a comunidade que esteja em busca de conteúdo relevantes sobre *blockchain*, Bitcoin e criptomoedas semelhantes que utilizam o algoritmo de PoW. Dessa forma, dois trabalhos principais foram relacionados: a construção de um sistema que de forma prática auxilia quem está a procura de conhecimento; e outro trabalho incluído, não por seu exemplo prático, mas pela quantidade de conteúdos de qualidade focados em ensinar conceitos relevantes sobre o mundo das criptomoedas e *blockchains*.

3.1.1 Blockchain Demo

O Blockchain Demo, foi uma solução de código aberto feita por Anders Brownworth (BROWNWORTH, 2016), com o objetivo de ensinar sobre a tecnologia *blockchain*, de maneira acessível e compreensível para o público de menor grau técnico.

O usuário obtém conhecimento da plataforma por meio de vídeos tutoriais realizados pelo criador e postados no YouTube, além do conhecimento prático que envolve assuntos como:

a) *Hash*, em que se percebe a mudança de um *hash* a cada alteração nos dados a serem criptografados;

b) Bloco, que torna possível a observação de informações como, número, *nonce*, dados e os *hashes*;

c) *Blockchain*, que permite visualizar a conexão entre os blocos;

d) Sistemas distribuídos, dessa forma é possível observar várias divisões de uma blockchain (forks);

e) E para finalizar, os *tokens*, que permite observar as transações realizadas na camada de dados.

Essa solução pode ser encontrada no site de Anders (BROWNWORTH, 2016) ou em seu repositório no Github <https://github.com/anders94/blockchain-demo>.

3.1.2 Blockchain Demo 2.0

Apesar do nome parecido com o trabalho mencionado na subseção anterior, é uma solução diferente criada por (SEAN, 2017). É uma ferramenta web que permite realizar a simulação de operações em uma *blockchain*, bem próximas da realidade.

A solução conta com um tutorial explicativo que descreve mais de 24 tarefas que podem ser realizadas em um ambiente *blockchain*. Suas principais características estão de acordo em como funciona esta solução, como funciona a *blockchain*, o que são blocos, *index*, *timestamp*, *hash*, dados e *nonce*. Detalha também o processo de mineração, as mudança de dados e seus efeitos, como é feita a adição de blocos, o que são as conexões *peer-to-peer*, o que é imutabilidade e para finalizar explicam sobre um conceito de vulnerabilidade, o ataque de 51%.

O repositório desta solução pode ser encontrado no Github <https://github.com/0xs34n/blockchain>.

3.1.3 Binance Academy

Binance Academy é um site com diversos tópicos relacionados às criptomoedas. Assim como esta pesquisa e os demais trabalhos correlatos, busca ensinar conceitos fundamentais sobre a tecnologia de *blockchain* para interessados sobre o tema, ele tem o intuito de auxiliar e também sugerir casos de uso para construção de alguns sistemas nessa linha.

Ele foi criado pela empresa Binance e se encontra hospedado no seguinte endereço da web (BINANCE, 2020). Este site é uma das fontes mais repletas de conteúdos acessíveis, rápidos e confiáveis do meio.

3.2 Simuladores de Blockchain

Essa seção tem como objetivo incluir trabalhos científicos relacionados a criação de ferramentas para simulação ou auxílio na criação de sistemas baseados na tecnologia *blockchain*.

3.2.1 BlockSim: A Simulation Framework for Blockchain Systems

O trabalho feito pelos autores Maher Alharby e Aad van Moorsel (ALHARBY; MOORSEL, 2020) vai de acordo com a proposta deste trabalho de conclusão de curso. Eles desenvolveram um simulador *blockchain* de código aberto que funciona como um *framework* para a construção de sistemas baseados em *blockchain*.

O *framework* permite simular transações, minerações, blocos, *forks*, distribuição de incentivos, rede de mineradores e configurar os nós participantes.

3.2.2 BlockSim: Blockchain Simulator

O trabalho (FARIA, 2018) buscou criar um simulador flexível que possibilita avaliar diferentes soluções baseadas em *blockchain*. O simulador permite a modelagem por meio da extensão dos modelos existentes e possibilita que se realize a alteração das condições de funcionamento do sistema. O artigo científico conta com a modelagem das redes Bitcoin e Ethereum.

Este trabalho pode ser mencionado como um avaliador de alta qualidade para soluções que utilizam da tecnologia *blockchain*, por exemplo, ao criar uma nova criptomoeda, seria possível avaliar o funcionamento do sistema de acordo com diferentes situações, como a propagação das transações e a entrega de novos blocos.

3.2.3 BlockSIM: A practical simulation tool for optimal network design, stability and planning

O artigo (PANDEY et al., 2019) busca apresentar uma ferramenta que visa auxiliar arquitetos e engenheiros de softwares a avaliarem o desempenho das redes de *blockchain* privadas de acordo com o propósito das mesmas e dá a possibilidade de comparação dos resultados obtidos com redes de *blockchain* reais.

A solução visa observar a estabilidade de um sistema e o *Throughput* para as transações. Possibilita a execução de diferentes cenários ao poder alterar os parâmetros ideais de um sistema para os propósitos desejados.

Dessa forma, a ferramenta criada por este trabalho auxilia no planejamento e implementação de estruturas baseadas em *blockchain* a serem escaláveis, estáveis e adaptáveis a demais adversidades.

3.2.4 BlockSim-Net: A Network Based Blockchain Simulator

O artigo (AGRAWAL et al., 2020) é uma solução com um intuito semelhante à solução apresentada por (ALHARBY; MOORSEL, 2020), porém este simulador traz um maior tom de realidade para os experimentos, pois sua solução foi criada para atuar de

forma distribuída, a fim de capturar as adversidades de uma rede real, como atrasos na propagação, por exemplo. Ele é baseado em um sistema de rede *blockchain* simples, porém de alto desempenho e bastante realístico.

Assim, esta solução permite a simulação de transações, minerações, propagação de blocos, *forks*, distribuição de incentivos, rede de mineradores, configurar os nós participantes e obter consenso. Todas essas simulações são realizadas de forma distribuída e não só em uma máquina local.

3.3 Vantagem de utilizar a biblioteca

A biblioteca desenvolvida é um projeto de código aberto, acessível a toda comunidade. Está documentado e publicado em português do Brasil, assim, torna-se uma possibilidade para construção de trabalhos que fortalecem a academia e comunidade *open source* brasileira.

A solução permite a incrementação de diversos algoritmos existentes para obtenção de consenso, como prova de autoridade, prova do tempo decorrido e prova de aposta. Também conta com a possibilidade da experimentação de algoritmos de consenso ainda inexistentes, que podem ser manipulados e executados por meio da biblioteca.

Outras utilidades que podem ser exploradas se referem a eventos que comumente são realizados em *blockchains* realísticas, como:

a) O processo de mineração egoísta;

b) Adaptação dos poderes computacionais dos mineradores de forma dinâmica, que pode ocasionar no aumento gradual dessa característica;

c) Simulação de um ataque de *cryptojacking*;

d) Transformação da biblioteca local, para possibilitar simulações por meio da rede;

e) Envio de recompensas para os nós que realizarem a mineração;

A Tabela 2 busca comparar alguns tópicos da solução proposta neste trabalho com os trabalhos correlatos apresentados neste capítulo. O intuito da tabela é de “apenas” ressaltar o diferencial da biblioteca construída, todos os trabalhos realizados são de extrema importância e colaboram com soluções e aprendizados sobre a tecnologia *blockchain*.

Tabela 2 – Tabela comparativa entre os trabalhos correlatos com a biblioteca desenvolvida.

Solução	**Open source**	**Manipular variáveis e algoritmos de consenso**	**Suporte em português**	**Gerar gráficos para análise de forks**
3.1.1	X	X		
3.1.2				
3.1.3			X	
3.2.1	X	X		
3.2.2	X	X		
3.2.3	X	X		
3.2.4	X	X		
Biblioteca criada	X	X	X	X

Fonte: próprio autor, 2023.

4 Desenvolvimento

O desenvolvimento da biblioteca, se dá na construção de um código fonte com classes e métodos que possibilitam simular de forma simples e eficaz soluções que utilizam a tecnologia de *blockchain*.

Como objetivo final, o intuito é a entrega de uma solução que seja capaz de cumprir os seguintes itens básicos de uma *blockchain* baseada no algoritmo de obtenção de consenso Proof of Work:

a) Possibilitar a criação de mineradores individuais e poder incluí-los em uma base de mineração com mais de um minerador, a fim de simular uma rede *blockchain* com vários nós;

b) Realizar conexões entre os mineradores, para que cada minerador tenha pelo menos um vizinho sempre que existir outros mineradores, o objetivo é de que cada minerador possa propagar suas atualizações na *blockchain* dos outros mineradores da rede;

c) Realizar uma espécie de loteria para a escolha de qual(is) minerador(es) irá(ão) processar a mineração de um bloco, ao possuir como fator fundamental seu poder computacional;

d) Realizar o processo de mineração, a fim de obter consenso com os blocos anteriores, para isso, o minerador deve realizar esta etapa de acordo com o algoritmo de PoW;

e) Ao minerar, atualizar a própria *blockchain* do minerador e propagar as atualizações aos seus vizinhos de acordo com a premissa no item b;

f) Permitir a ocorrência de mais de uma *blockchain* válida ao mesmo tempo, ou seja, possibilidade de *forks*;

g) Possibilidade para realização de ataques que visam encontrar vulnerabilidades na *blockchain*;

h) Exportar informações relevantes para o usuário.

Nos tópicos abaixo serão apresentados os principais métodos e conceitos utilizados na aplicação desenvolvida, sua forma de trabalho e pensamento para resolução de problemas.

4.1 Blockchain

A *blockchain* foi pensada não só em uma estrutura de blocos que simularia um banco de dados distribuído, mas sim em um objeto com características particulares que trazem um maior tom de realidade para o projeto, ao tornar a *blockchain* uma simulação de um verdadeiro livro razão. A Tabela 3 detalha a decomposição da classe Blockchain com seus atributos e a principal função da existência de cada um.

Tabela 3 – Classe Blockchain e seus atributos.

Atributo	**Objetivo**
representante	O representante se refere ao minerador detentor da *blockchain* analisada, cada minerador possui sua própria *blockchain* e ela conta com as informações desse minerador e essa informação se torna relevante para completar o histórico de quem minerou cada bloco.
livro_razao	O livro razão representa o conjunto de blocos interligados por *hashes* em consenso, é nele que se encontra todos os dados referentes às transações, ou seja, cada bloco de informação estará presente dentro do livro razão.
topo	O topo é o último bloco inserido no livro razão, dessa forma é possível verificar se todos os mineradores estão de acordo em relação ao último bloco inserido e, juntamente com o histórico de mineradores, pode-se analisar a ocorrência de *forks* (ato ou efeito de forquear, origem dos termos bifurcações, trifurcações e consequências que resultam em divisões válidas a partir de uma origem).
historico_mineradores	O histórico de mineradores se refere a quem minerou cada bloco existente na *blockchain*, ele é o "dedo duro" que relata o *hash* de um bloco e quem realizou a mineração deste bloco.
fraudada	Este atributo identifica se uma *blockchain* possui blocos fraudados em sua cadeia, definido por padrão como false (falso) e sempre que houver uma fraude é modificado para true (verdadeiro). Usado para facilitar a geração de relatórios.

Fonte: próprio autor, 2023.

4.2 Rede de mineradores

Para que seja possível explorar os conceitos de *blockchain* e realizar uma simulação mais próxima a realidade do assunto, é necessária a criação de uma rede de mineradores que represente o sistema P2P que, costumeiramente, é usado nas principais soluções que envolvem o armazenamento de dados de forma segura e descentralizada, por exemplo, o Bitcoin. Portanto, é necessário que um objeto represente cada um desses nós da rede de forma individual, com o objetivo de dar vida aos mineradores. Além dos mineradores, é crucial que seja criado um outro objeto que tenha o poder de identificar os participantes que fazem parte da rede e efetuar conexões entre eles para que as informações possam transitar entre os envolvidos. Abaixo, esses objetos serão retratados como "Minerador" e "Mundo".

4.2.1 Minerador

O minerador é o objeto responsável por contribuir com as atividades individuais de um nó na rede, por exemplo, sempre tentar minerar um novo bloco e se estiver apto a realizar a mineração, minerar este bloco de acordo com o mecanismo de consenso estabelecido na criação do objeto. Ao minerar, o minerador deve propagar sua atualização aos seus vizinhos (nós que ele possui conexão direta), cada vizinho ao receber a notificação de atualização deve validar essa novidade e se tudo estiver correto, realizar o repasse das informações para seus vizinhos até que a rede toda possa estar atualizada. A Tabela 4 detalha os atributos de um minerador.

Tabela 4 – Classe Minerador e seus atributos.

Atributo	**Objetivo**
identificador	O identificador é o numeral único e individual de cada minerador, é equivalente a identidade utilizada para diferenciar um minerador de outro.
mecanismo	O mecanismo é a forma de obter consenso entre os blocos, por meio dele um minerador pode chamar diferentes métodos para obtenção de consenso de acordo com a rede, na biblioteca tem-se a prova de trabalho o mecanismo já desenvolvido, porém cabe a inserção de novos algoritmos em futuros trabalhos.
poder_computacional	O poder computacional é um valor numérico que representa a disponibilidade de recursos que um minerador pode usar, ou seja, quanto maior o poder computacional de um minerador, maiores são as chances dele minerar um novo bloco.
vizinhos	Os vizinhos são outros mineradores conhecidos por um minerador, é para eles que o minerador propagará suas atualizações na *blockchain*.
blockchain	Cada minerador possui sua *blockchain* e se atualizará à medida em que realiza um processo de mineração ou é notificado sobre a atualização de algum outro minerador.
propagar	O atributo propagar é uma "flag" que sinalizará para os vizinhos de um minerador que realizou uma mineração com sucesso e este minerador estará apto para propagar essa atualização, caso contrário o minerador fica definido como inapto a propagar um novo bloco.
fraudador	Atributo utilizado para identificar um minerador que irá minerar apenas blocos fraudados.

Fonte: próprio autor, 2023.

4.2.2 Mundo

O mundo é o objeto responsável por transformar os mineradores individuais em uma rede, por meio dele que as conexões entre cada minerador são criadas e o grau de vizinhança é estabelecido. Portanto, esse objeto torna possível o compartilhamento das atualizações na *blockchain*.

O mundo também é capaz de detectar se existem mais de uma *blockchain* válida

na rede ao mesmo tempo e registrar quaisquer *forks* no momento em que ocorreram. A Tabela 5 descreve os atributos que o Mundo possui.

Tabela 5 – Classe Mundo e seus atributos.

Atributo	**Objetivo**
mineradores	Os mineradores são representados por um dicionário de dados que identificam o poder computacional de um minerador, esse atributo é o responsável pela criação e manipulação da rede simulada.
poder_mundial	O poder mundial se refere a soma total do poder computacional de todos os mineradores existentes na rede multiplicado por dez. Este dez é um número empírico com o intuito de modelar a dificuldade da rede, assim é possível simular um tempo de mineração entre os participantes, caso ela seja retirado ou seja um número menor, a característica da rede pode mudar e a ocorrência de *forks* aumentar, assim, ele possui o objetivo de que não ocorra uma mineração a cada instante.
bifurcacoes	As bifurcações são representadas por um dicionário de dados que contém as informações sobre a altura da *blockchain* quando houve uma bifurcação, o bloco onde ocorreu a bifurcação, os mineradores que entraram em conflito e o poder computacional que cada um dos mineradores relacionados possui.

Fonte: próprio autor, 2023.

4.3 Processo de mineração

Concluída a criação da base de mineradores e definida as conexões entre os nós, o próximo passo do desenvolvimento é conseguir fazer cada minerador "tentar" realizar o processo de mineração. O intuito das tentativas de minerar é criar uma maneira de ponderar o poder computacional individual, essa métrica tem o objetivo de fazer com que o minerador com maior recurso computacional tenha vantagem para conseguir a mineração em concorrência a um minerador com poucos recursos tecnológicos, assim, mineradores poderosos mineram uma maior quantidade de blocos do que mineradores mais fracos.

4.3.1 Ponderamento

Para realizar essa operação, é necessário conhecer o poder individual de cada minerador que irá efetuar a tentativa de mineração e o poderio total da rede. A divisão que

representa esse ponderamento é dada por:

$$SorteioAleatorio(0..1) <= \frac{PoderMinerador}{PoderRede}$$

Um número aleatório entre 0 e 1 é sorteado, se este número sorteado for menor ou igual a chance do minerador minerar, o minerador iniciará a mineração do bloco. Esse processo enfatiza a importância do poder computacional de um minerador, pois quanto mais poder, maior será o resultado da divisão dos recursos pelo poderio total da rede, o que resulta em uma grande chance desse minerador poder realizar a mineração de um novo bloco. Em termos de codificação tem-se as seguintes operações referentes ao sorteio para mineração:

```
'''
* Classe: Minerador
'''
def tentar_mineracao(self, poder_mundial):
    if ((random.uniform(0, 1)) <= (self.poder_computacional)/
         poder_mundial):
        bloco = Bloco(self)
        self.minerar(bloco)
        return self
```

4.3.2 Mecanismo de prova de trabalho

Após o sucesso na tentativa de minerar, um minerador torna-se apto para realizar o processo completo, desde a construção de um bloco em consenso ao utilizar o algoritmo de prova de trabalho, até a propagação para os seus vizinhos sobre sua atualização.

O algoritmo de prova de trabalho, *proof of work*, busca criar uma métrica de consenso entre os blocos minerados que serão considerados válidos para a rede, portanto, ao criar um novo bloco, é necessário que ocorra uma referência ao *hash* do bloco anterior, para produzir uma cadeia de blocos conectados entre si. Uma exceção ocorre durante a inserção do primeiro bloco na *blockchain* (bloco gênesis), ele é inserido com o *hash* de 64 caracteres do bloco anterior descritos como 0, pois ele não possui blocos anteriores, portanto esse bloco é o início da cadeia.

Após configurada a referência à cadeia de blocos (*blockchain*), os dados (transações) são inseridos no novo bloco e criptografados, o que ocasiona em um *hash* sem consenso. Para torná-lo um bloco válido, é definido um prefixo sobre os *hashes*, por exemplo, o início de cada *hash* de um bloco ser iniciado em “00”, portanto, para obter consenso nessa métrica, os mineradores realizam operações matemáticas utilizando seus próprios recursos computacionais para calcular um *hash* equivalente as informações geradas no

novo bloco que inicie com o prefixo desejado. A quantidade de tentativas realizadas para tentar encontrar um *hash* válido é chamada de *nonce*.

Ao achar um *hash* válido para o bloco, é dito que o minerador conseguiu obter consenso deste bloco e está apto a propagar essa atualização para seus vizinhos, esses que validarão a novidade e definirão se realmente houve consenso, ou seja, se devem recusar ou aceitar o novo bloco. A codificação do mecanismo de consenso se encontra abaixo:

```
'''
* Classe: Mecanismo
'''
def prova_de_trabalho(self):
    quantidade_prefixo = 2
    prefixo = '0'*quantidade_prefixo
    maximo_nonce = 100000000000

    for nonce in range(maximo_nonce):
        informacoes_bloco = str(self.bloco.numero) + \
            self.bloco.dados + \
            self.bloco.hash_anterior + \
            str(self.bloco.fraudado) + \
            str(nonce)

        hash_bloco = sha256(informacoes_bloco.encode('ascii')).hexdigest()

        if hash_bloco.startswith(prefixo):
            self.bloco.nonce = nonce
            self.bloco.hash_proprio = hash_bloco
            self.minerador.blockchain.inserir(self.bloco)

            return self.minerador

    raise BaseException(
        '\nNão foi possível realizar a mineração.'
        'Foram feitas: {valor_maximo_nonce} de tentativas\n')
```

4.4 Forks

Ao compreender que cada minerador tentará realizar o processo de mineração, deve-se levar em consideração que pode acontecer de dois ou mais mineradores realizarem o processo de mineração ao mesmo tempo, o que traz a ocorrência de *forks* (termo em inglês para bifurcação). O termo bifurcação não será usado por não englobar mais de duas divisões válidas de uma *blockchain*, no cenário da biblioteca e do Bitcoin pode haver mais do que uma bifurcação, trazendo trifurcações ou mais ramificações. O *fork* na *blockchain* é um termo inglês, que trata o conceito de divisões válidas que vieram de uma *blockchain* de origem, dessa maneira, há ocorrência de três possíveis cenários:

a) Ninguém conseguir se tornar apto para realizar o processo de mineração;

b) Apenas um minerador estar apto para realizar o processo de mineração, esse é o melhor caso de mineração em *blockchain*, o qual todos são atualizados com apenas um único bloco e o consenso é obtido sem divergências na rede;

c) Dois ou mais mineradores estarem aptos para realizarem o processo de mineração, nesse caso tem-se a ocorrência de *forks*, quando duas ou mais *blockchains* válidas circulam pela rede.

O problema dos *forks* é resolvido quando um único minerador encontrar a prova de trabalho de um novo bloco, assim, não terá concorrentes para competir com sua *blockchain* de maior altura e a rede passa a ignorar as bifurcações desatualizadas para assumir um único livro razão válido.

4.5 Persistência de dados

Para testar a utilidade da biblioteca construída tornou-se necessária a criação de uma classe referente à persistência dos dados relevantes, assim o usuário da solução poderá observar as principais informações do contexto de uma rede que utiliza o armazenamento de dados em *blockchain*. A Tabela 6 detalha os tipos de persistência que a biblioteca permite.

Tabela 6 – Informações sobre os arquivos de persistência da biblioteca.

Método	**Objetivo**
persistir_mineradores	Exportar para um arquivo CSV uma tabela com as informações sobre o identificador de um minerador, a quantidade de vizinhos que ele possui, seu poder computacional, a quantidade de blocos minerados por ele e a razão entre blocos minerados por poder computacional.
persistir_blockchain	Exportar para um arquivo CSV as informações sobre cada bloco existente no livro razão, com destaque para número do bloco, dados da transação, *hash* do bloco anterior, *nonce* (número responsável por identificar a quantidade de tentativas necessárias para encontro de um *hash* válido que esteja em consenso com a rede) e o *hash* do próprio bloco.
persistir_historico	Exportar para um arquivo CSV as informações sobre cada bloco existente no livro razão, com destaque para o *hash* do bloco minerado e o minerador que realizou este processo.
persistir_bifurcacoes	Exportar para um arquivo PNG um gráfico com o eixo das ordenadas referente à altura da *blockchain* e com o eixo das abscissas referente a quantidade de *forks* (quantidade de *blockchains* válidas existentes durante determinada altura) .

Fonte: próprio autor, 2023.

5 Resultados

Os resultados foram obtidos com base na criação de uma biblioteca peso-leve codificada em Python. O código fonte da biblioteca é acessível a toda a comunidade que opta por conhecer sobre o assunto *blockchain* e deseja compartilhar seus conhecimentos com o mundo. Suas principais características são a simplicidade no entendimento, a facilidade de instalação e o pouco consumo de recursos computacionais.

Ela foi desenvolvida para possibilitar de forma prática a simulação dos principais conceitos abordados na fundamentação teórica com aplicações educacionais e científicas. A biblioteca possui código fonte aberto e está disponível para qualquer pessoa que tenha curiosidade em se aventurar pelo tema e sugerir modificações.

O código fonte desta solução está disponível no Github, uma ferramenta de versionamento online e gratuita na web, ao poder ser acessada a partir do repositório (SANTOS, 2022).

5.1 Variações nos poderes computacionais e na quantidade de vizinhos

Este experimento busca trazer um tom realístico e mais próximo à uma simulação do Bitcoin, por meio dele foi criada uma base de 30 mineradores com poderes computacionais variados entre 1 e 100, considerando 1 o mais fraco e 100 o mais poderoso, dessa forma se um minerador possui o poder computacional próximo a 1, é considerado que ele conta com pouquíssimos recursos computacionais e se um minerador possui um poder próximo a 100, significa que ele possui todos os recursos necessários para conseguir minerar um bloco com maior facilidade. Essa hipótese pode ser analisada nas informações exibidas no gráfico da Figura 9, relacionado a distribuição da quantidade de blocos minerados por minerador de acordo com o aumento do poder computacional da rede.

Além dos poderes computacionais alternados, outra característica deste experimento é a variação da quantidade de vizinhos que cada minerador terá, esta quantidade é definida de acordo com o tamanho da base, se uma base de tamanho X for criada, o minerador pode ter de 1 até X-1 de nós conhecidos por ele na rede, no experimento, de 1 até 29 vizinhos. A distribuição da quantidade de blocos minerados por minerador a medida que o número de vizinhos aumenta, está presente no gráfico da Figura 10.

De acordo com essas premissas pode acontecer de haver ocorrências de mineradores com somente 1 vizinho e outros mineradores estarem dispostos de 29 vizinhos, o mesmo

ocorre em relação aos poderes computacionais, em que um minerador pode ter 1 e outro possa ter 100.

O objetivo deste experimento é testar o funcionamento da biblioteca e obter as primeiras análises elaboradas por ela de forma que se obtenha um resultado fidedigno a um ambiente real. Serão realizadas 30 repetições com a mineração de 10 mil blocos. O poder computacional e o número de vizinhos de um minerador será o mesmo durante as 30 execuções do experimento e o resultado médio é exibido na Tabela 7.

Tabela 7 – Após 30 repetições, a tabela apresenta a média dos dados para os 30 mineradores que realizaram a mineração de 10000 blocos. Cada minerador possui variações das características de poder computacional e quantidade de vizinhos.

Poder Computacional (PC)	**Média de Blocos Minerados (BM)**	**Nº de vizinhos**	**Média da Razão (BM/PC)**
1	6,33	3	6,33
2	15,1	24	7,55
2	14,55	13	7,4
5	35,4	3	7,08
6	41,36	23	6,89
14	99,93	25	7,13
17	122,53	17	7,20
17	122,46	16	7,20
27	199,86	21	7,36
31	219,76	8	7,09
38	258,66	2	6,80
41	289,56	8	7,06
43	304,53	23	7,08
45	309,96	5	6,88
46	324,46	24	7,05
51	358,93	16	7,03
55	393,6	22	7,15
58	416,83	7	7,18
62	431,1	12	6,95
65	457,03	28	7,03
68	468,63	4	6,89
70	478,93	17	6,84
74	498,63	1	6,73
76	514,93	8	6,77
77	540,26	27	7,01
78	531,33	28	6,81
89	596,06	5	6,69
91	609,3	13	6,69
99	679,06	6	6,86
100	661,56	12	6,61

Fonte: próprio autor, 2023.

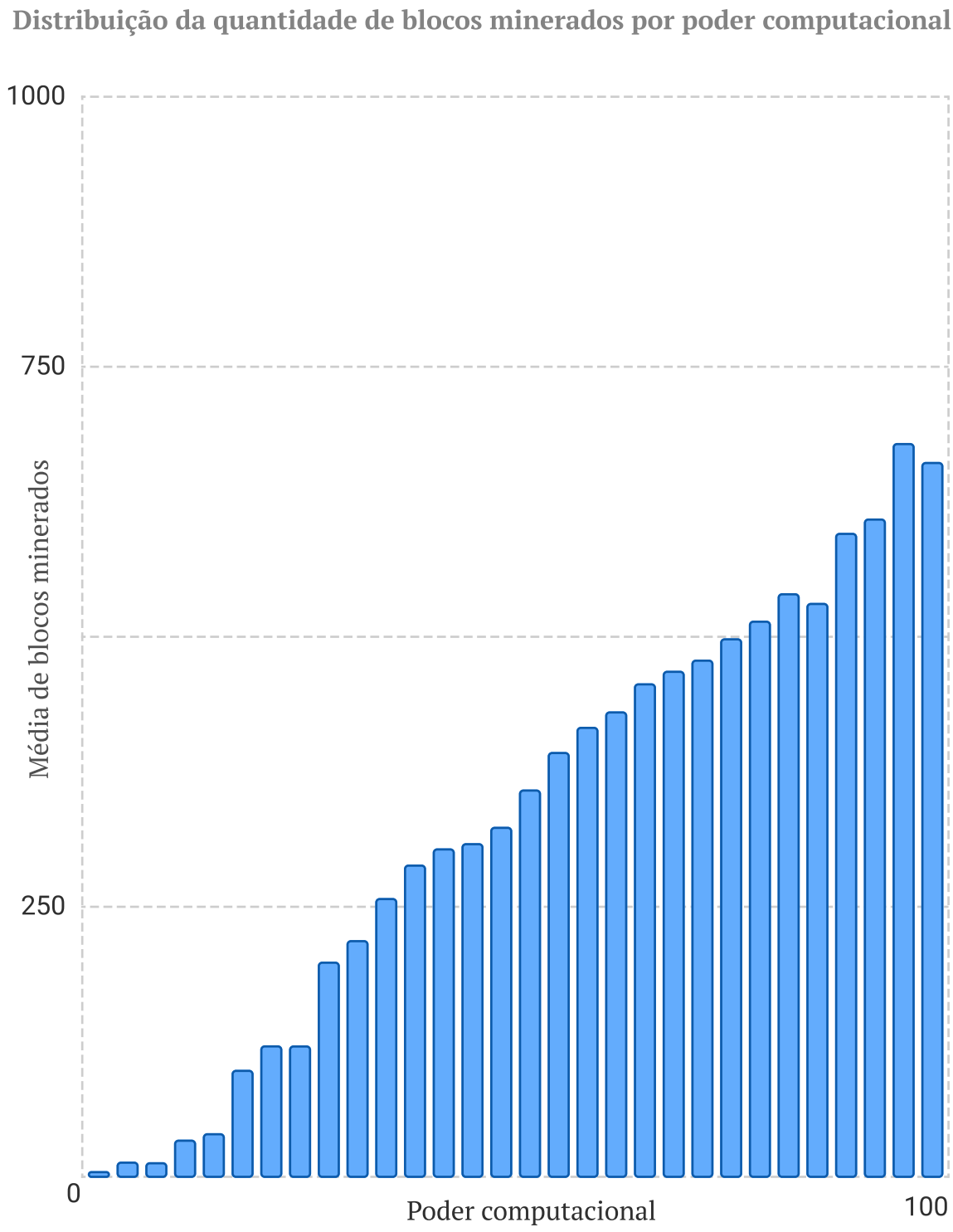

Figura 9 – Distribuição da quantidade de blocos minerados a medida que o poder computacional dos mineradores aumenta.

Fonte: próprio autor, 2023.

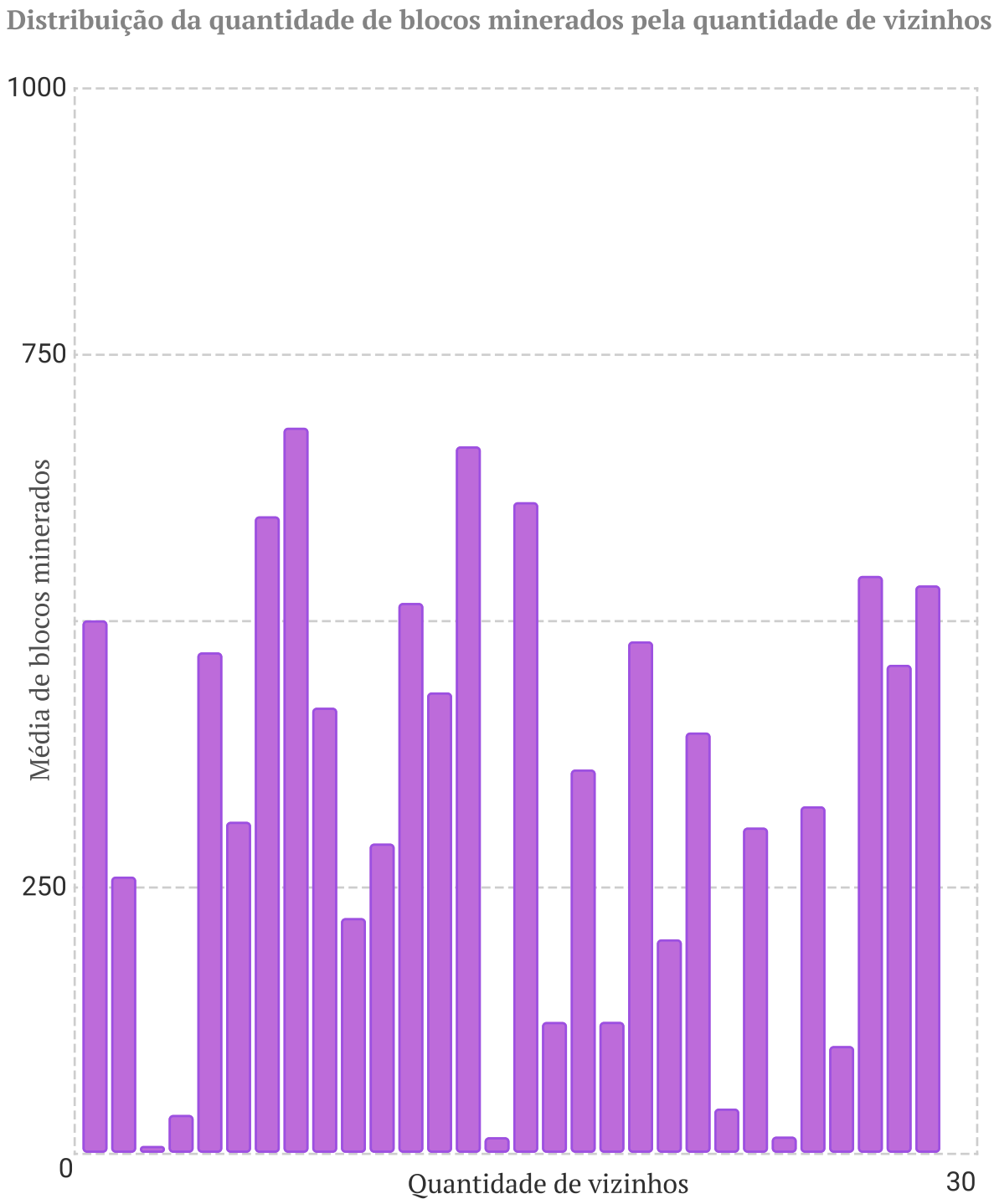

Figura 10 – Distribuição da quantidade de blocos minerados a medida que a quantidade de vizinhos dos mineradores aumenta.

Fonte: próprio autor, 2023.

A primeira percepção ao observar os resultados da Tabela 7 e o gráfico da Figura 9 é notar que, quanto maior o poder computacional de um minerador, mais blocos são minerados por ele, já que quanto maior o poder, mais recursos para mineração estão

disponíveis. Outra análise feita é relacionada à razão de blocos minerados por poder computacional, de forma que se mantêm bem próximas, oscilando entre 6,33 até 7,55 com média de 6,97 e mediana de 7,02, bem próximo ao resultado da média aritmética da razão.

5.2 Variações nos poderes computacionais com a mesma quantidade de vizinhos

Este experimento visa demonstrar a capacidade de influência do poder computacional sobre a quantidade de blocos minerados. Para obter um resultado fiel, os mineradores serão dispostos com a mesma quantidade de vizinhos.

Neste processo foi criada uma base de 30 mineradores com poderes computacionais variados entre 1 e 100, considerando 1 o mais fraco e 100 o mais poderoso, portanto se um minerador possui o poder computacional próximo a 1, é considerado que ele conta com pouquíssimos recursos computacionais e se um minerador possui um poder próximo a 100, significa que ele possui todos os recursos necessários para conseguir minerar um bloco com maior facilidade. Essa hipótese pode ser analisada nas informações exibidas no gráfico da Figura 11, relacionado a distribuição da quantidade de blocos minerados por minerador de acordo com o aumento do poder computacional da rede.

Como o objetivo deste experimento é descobrir a influência do poder computacional, a quantidade de vizinhos de todos mineradores é definida em somente cinco vizinhos conhecidos por minerador. Serão realizadas 30 repetições com a mineração de 10 mil blocos. O poder computacional e o número de vizinhos de um minerador será o mesmo durante as 30 execuções do experimento e o resultado médio é apresentado na Tabela 8.

Tabela 8 – Após 30 repetições, a tabela apresenta a média dos dados para os 30 mineradores que realizaram a mineração de 10000 blocos. Cada minerador possui quantidade de vizinhos fixada em 5.

Poder computacional (PC)	**Média de Blocos Minerados (BM)**	**Média da Razão (BM/PC)**
4	27,7	6,92
8	55,6	6,95
18	130,93	7,27
20	149,63	7,48
22	162,23	7,37
30	212,6	7,08
32	227,9	7,12
34	236,26	6,94
36	260,56	7,23
38	269,03	7,07
39	282,03	7,23
40	273,86	6,84
41	287,3	7,00
43	305,16	7,09
43	295,6	6,87
48	340,53	7,09
49	348,63	7,11
54	375,23	6,95
56	383,7	6,85
57	387,96	6,80
57	394,2	6,91
60	417,83	6,96
64	446,3	6,97
66	450,1	6,82
67	448,93	6,7
68	459,33	6,75
77	518,43	6,73
86	592,83	6,89
92	628,7	6,83
92	630,83	6,85

Fonte: próprio autor, 2023.

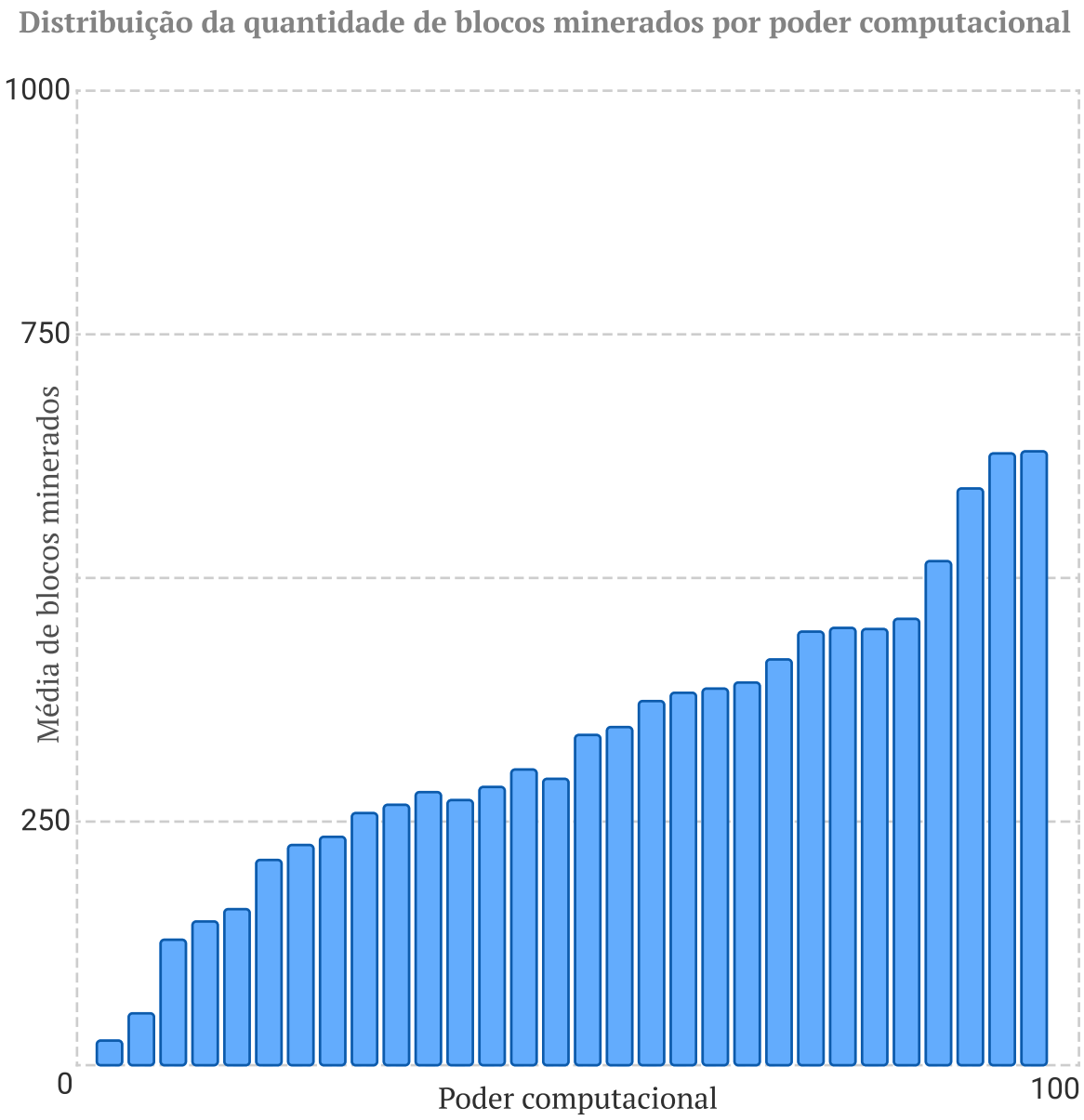

Figura 11 – Distribuição da quantidade de blocos minerados a medida que o poder computacional dos mineradores aumenta sem a influência de vizinhos.

Fonte: próprio autor, 2023.

Ao observar a Tabela 8 e o gráfico da Figura 11, nota-se uma grande semelhança ao experimento anterior, trazendo novamente a percepção de que o poder computacional possui influência sobre a quantidade de blocos minerados. A variação da razão entre blocos minerados por poder computacional também é bem próxima entre os mineradores, com variações de 6,7 até 7,48 com a média aritmética de 6,98, desvio padrão de 0,19 e mediana de 6,95.

5.3 Sem variações nos poderes computacionais e com variações na quantidade de vizinhos

Este é o experimento mais complexo do trabalho, ele busca entender a influência da quantidade de vizinhos sobre a quantidade de blocos minerados por minerador, de forma que se um minerador possui uma maior quantidade de vizinhos, suas atualizações são passadas com maior velocidade e, na ocorrência de *forks*, ele será beneficiado por isso.

Para poder nivelar apenas pela quantidade de vizinhos, todos os mineradores irão possuir capacidade computacional igual, o que é equivalente a dizer que todos os mineradores possuem recursos computacionais iguais. Para poder perceber a diferença, a quantidade de blocos a serem minerados passa de 10 mil para 30 mil blocos e a quantidade de mineradores da rede sai de 30 e vai para 50 mineradores. Esses 50 mineradores são divididos em 2 grupos de 25 mineradores, no primeiro grupo, cada minerador possui 40 vizinhos, no segundo grupo, cada minerador possui apenas 1 vizinho. É possível analisar a distribuição da quantidade de blocos minerados por cada um desses grupos por meio do gráfico da Figura 12.

A partir desses pressupostos, espera-se que o grupo com maior quantidade de vizinhos possa repassar mais blocos na *blockchain* do que o grupo com menor quantidade de vizinhos e também pretende-se comparar a influência da quantidade de vizinhos sobre a influência que o poder computacional teve no experimento anterior. Serão realizadas 30 repetições com a mineração de 30 mil blocos. O poder computacional e o número de vizinhos de um minerador será o mesmo durante as 30 execuções do experimento de acordo com a definição de cada grupo. O resultado médio do experimento é apresentado na Tabela 9 e Tabela 10.

Tabela 9 – Após 30 repetições, a tabela apresenta a média dos dados para os primeiros 25 mineradores que realizaram a mineração de 30000 blocos. Cada minerador possui poder computacional fixado em 100 e esse grupo de mineradores da tabela possui 40 vizinhos cada.

Minerador	**Número de vizinhos**	**Média de Blocos Minerados (BM)**	**Média da Razão (BM/PC)**
1	40	662,83	6,62
2	40	670,56	6,70
3	40	668,56	6,68
4	40	661,83	6,61
5	40	660,4	6,60
6	40	660,83	6,60
7	40	654,26	6,54
8	40	653,1	6,53
9	40	649,93	6,49
10	40	652,33	6,52
11	40	650	6,5
12	40	653,46	6,53
13	40	646,13	6,46
14	40	643,46	6,43
15	40	639,06	6,39
16	40	634,29	6,34
17	40	628,53	6,28
18	40	631,46	6,31
19	40	634,56	6,34
20	40	632,7	6,32
21	40	621,56	6,21
22	40	632,13	6,32
23	40	620,66	6,20
24	40	611,1	6,11
25	40	613,8	6,13

Fonte: próprio autor, 2023.

Tabela 10 – Após 30 repetições, a tabela apresenta a média dos dados para os últimos 25 mineradores que realizaram a mineração de 30000 blocos. Cada minerador possui poder computacional fixado em 100 e esse grupo de mineradores da tabela possui 1 vizinho cada.

Minerador	**Número de vizinhos**	**Média de Blocos Minerados (BM)**	**Média da Razão (BM/PC)**
26	1	614,23	6,14
27	1	589,93	5,89
28	1	572,53	5,72
29	1	511,03	5,11
30	1	600,53	6,0
31	1	520,46	5,2
32	1	565,16	5,65
33	1	572,6	5,72
34	1	514,46	5,14
35	1	566,46	5,66
36	1	559,6	5,59
37	1	576,43	5,76
38	1	537,5	5,37
39	1	611,63	6,11
40	1	540,36	5,4
41	1	537,13	5,37
42	1	543,56	5,43
43	1	545,36	5,45
44	1	528,03	5,28
45	1	580,93	5,8
46	1	528,76	5,28
47	1	545,23	5,45
48	1	574,5	5,74
49	1	572,53	5,72
50	1	503,36	5,03

Fonte: próprio autor, 2023.

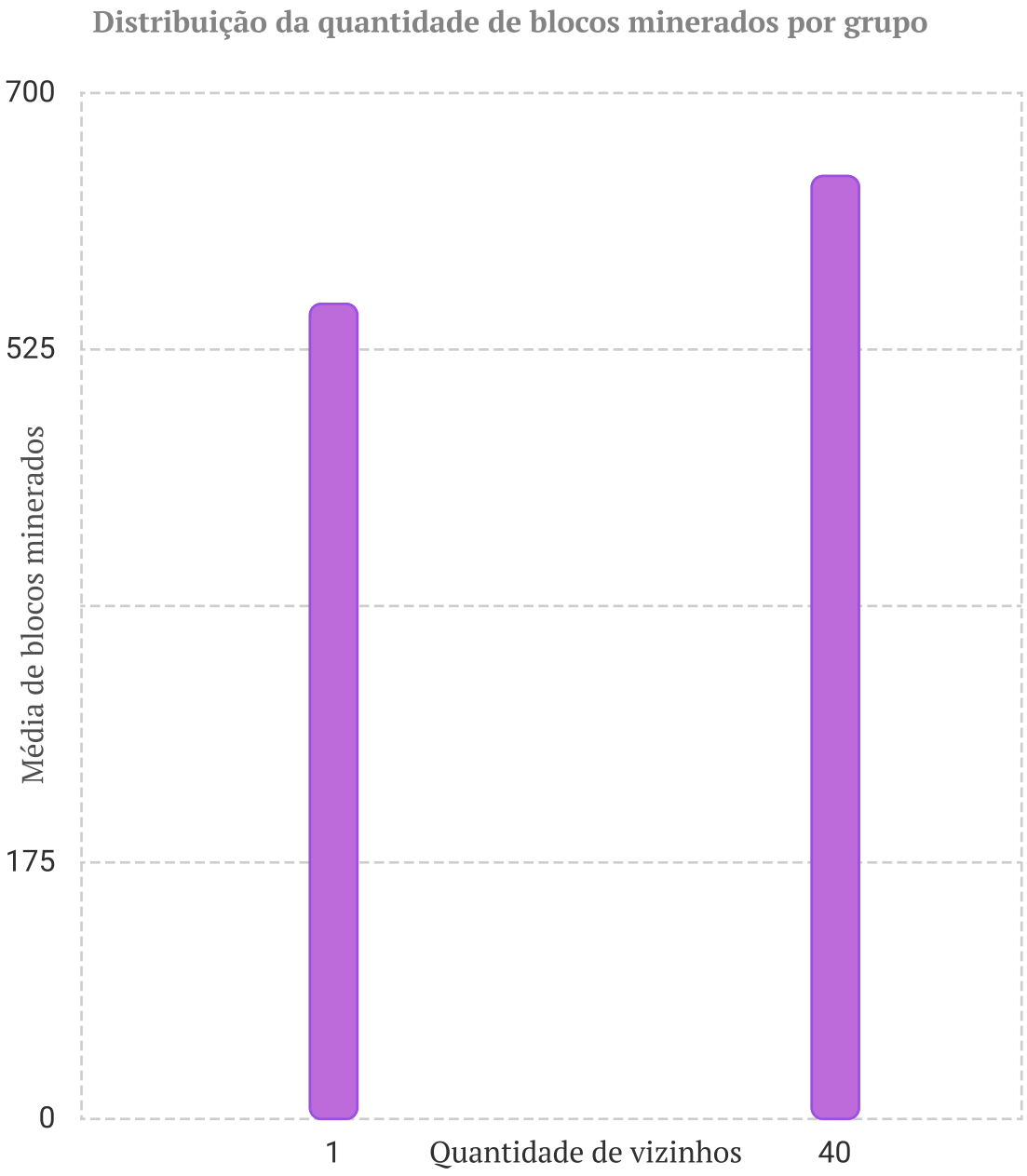

Figura 12 – Distribuição da quantidade de blocos minerados por grupo.

Fonte: próprio autor, 2023.

Ao analisar as duas tabelas geradas no experimento e pelo gráfico da Figura 12, observa-se os seguintes dados, para o primeiro grupo, configurado com uma maior quantidade de vizinhos, tem-se a média de blocos minerados de 643,50 com desvio padrão de 16,91. Para o segundo grupo, configurado com uma menor quantidade de vizinhos, tem-se a média de blocos minerados de 556,49 com desvio padrão de 30,72.

Nota-se que o grupo com uma pequena quantidade de vizinhos escreve uma menor quantidade de blocos na *blockchain*, o que leva a conclusão de que suas atualizações se

propagam com maior lentidão entre os mineradores.

Outro fator importante a ser comentado é que mesmo com 30 mil blocos minerados a diferença da razão de blocos mineradores é bem pequena, chegando a uma segunda conclusão de que o poder computacional dos mineradores é o principal diferencial para a mineração dos blocos.

5.4 Análise de forks

Todos os experimentos anteriores geraram gráficos para análises dos *forks* que poderiam acontecer durante o processo de mineração dos blocos. Por meio meio deste experimento, torna-se possível realizar análises sobre cada forma de *fork* (bifurcações, trifurcações, etc.).

Os gráficos gerados detalham a altura de cada *blockchain* até 100 primeiros blocos minerados em uma execução. Essa execução foi realizada com menor quantidade de blocos pelo intuito de facilitar a visualização. Através da análise gráfica das imagens é possível identificar o comportamento de alguns *forks*, desde seu início, até o fim.

5.4.1 Análise gráfica: variações nos poderes computacionais e na quantidade de vizinhos

O gráfico da Figura 13 tem o objetivo de relatar o aparecimento de *forks*, independente do seu formato (bifurcações, trifurcações ou mais variações) do experimento relacionado às variações nos poderes computacionais e na quantidade de vizinhos.

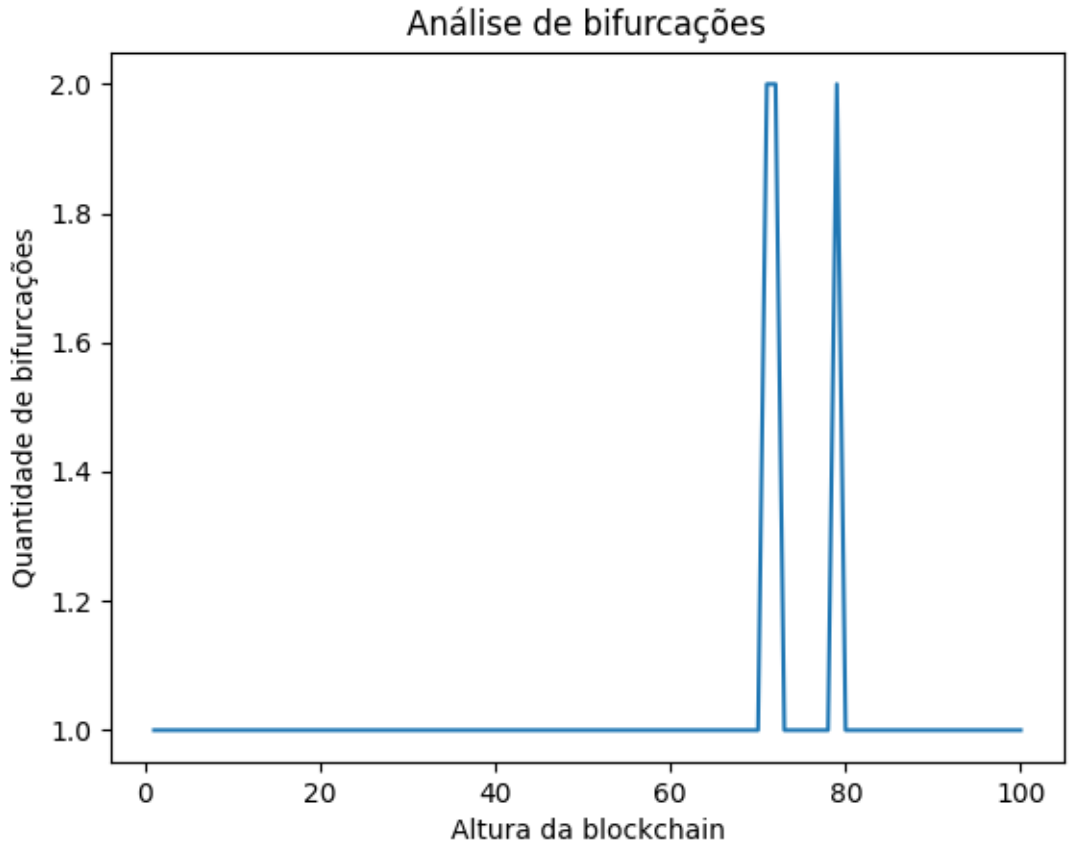

Figura 13 – Análise de *forks* no experimento 5.1 - Variações nos poderes computacionais e na quantidade de vizinhos.

Fonte: próprio autor, 2023.

Após análise do gráfico, observa-se dois *forks* no formato de bifurcações (apenas duas *blockchains* válidas ao mesmo tempo), sendo corrigidas em um pequeno intervalo de tempo.

5.4.2 Análise gráfica: variações nos poderes computacionais com a mesma quantidade de vizinhos

O gráfico da Figura 14 tem o objetivo de relatar o aparecimento de *forks*, independente do seu formato (bifurcações, trifurcações ou mais variações) do experimento relacionado às variações nos poderes computacionais com a mesma quantidade de vizinhos.

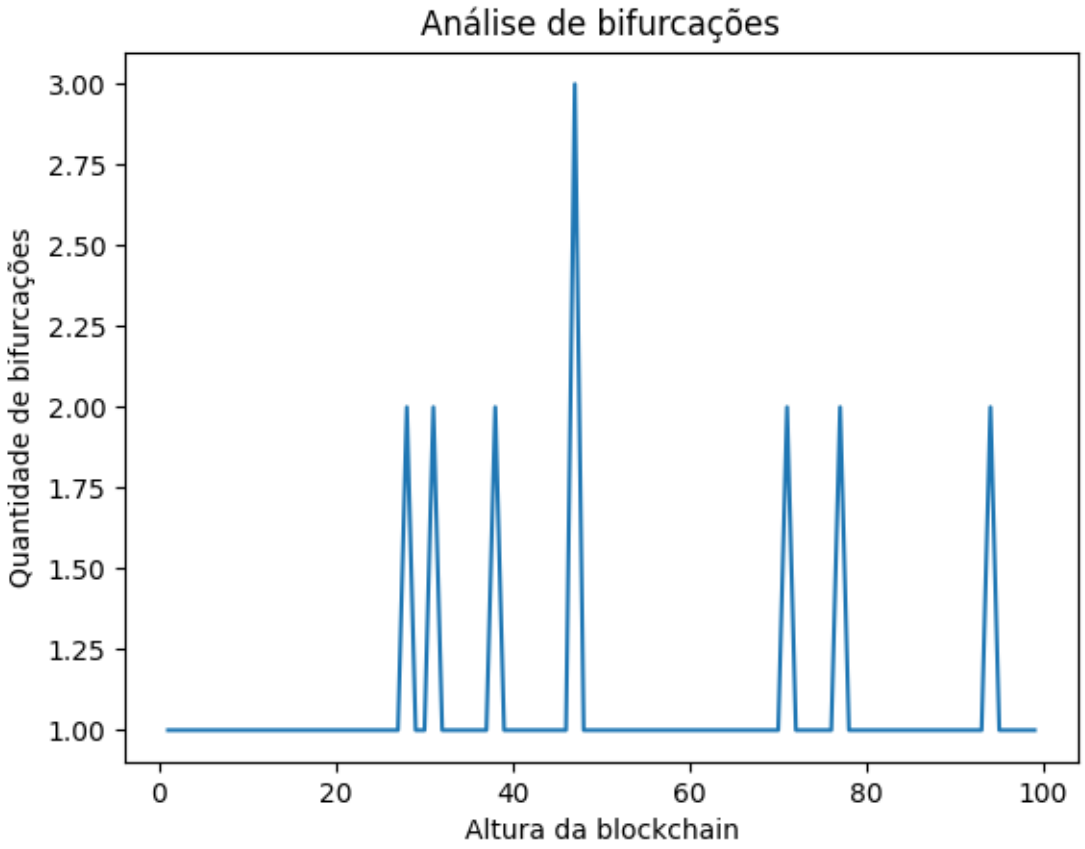

Figura 14 – Análise de *forks* no experimento 5.2 - Variações nos poderes computacionais com a mesma quantidade de vizinhos.

Fonte: próprio autor, 2023.

Após a análise do gráfico, nota-se uma maior quantidade de *forks*, ocorrendo bifurcações e uma trifurcação, neste experimento observa-se uma maior quantidade de *forks* em relação ao experimento anterior no começo da mineração dos blocos.

5.4.3 Análise gráfica: sem variações nos poderes computacionais e com variações na quantidade de vizinhos

O gráfico da Figura 15 tem o objetivo de relatar o aparecimento de *forks*, independente do seu formato (bifurcações, trifurcações ou mais variações) do experimento sem variações nos poderes computacionais e com variações na quantidade de vizinhos.

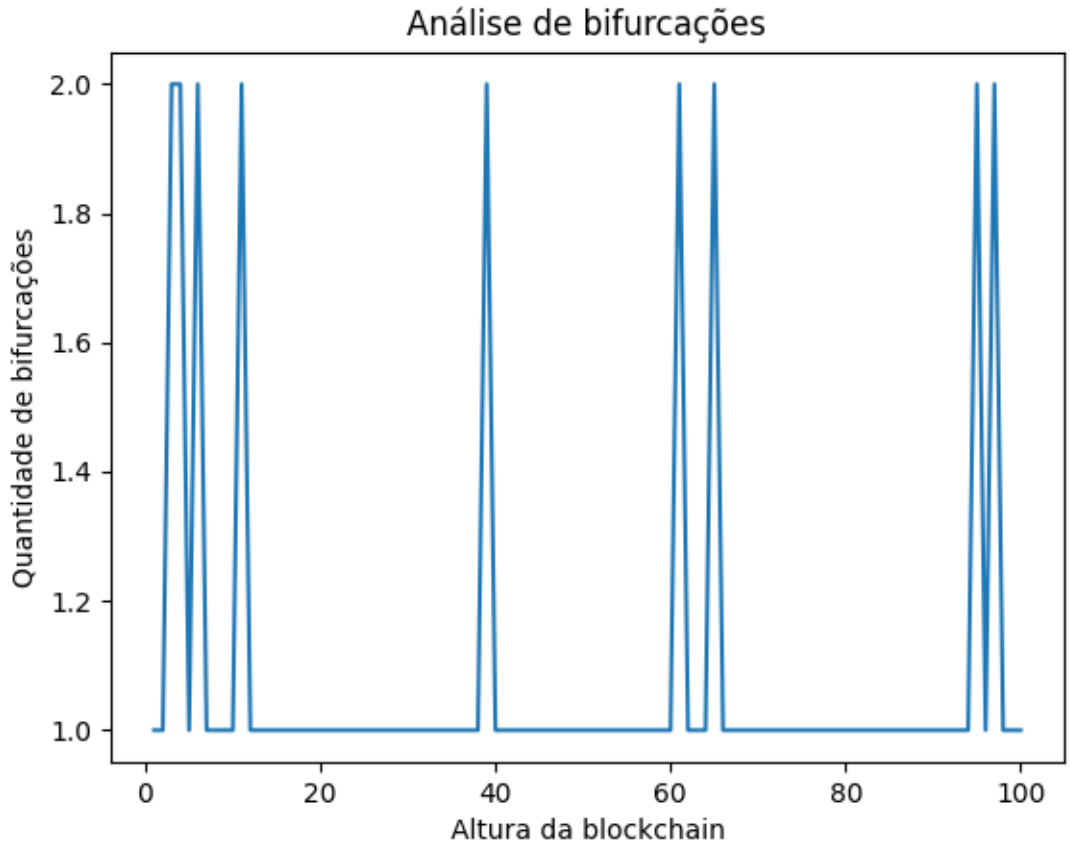

Figura 15 – Análise de *forks* no experimento 5.3 - Sem variações nos poderes computacionais e com variações na quantidade de vizinhos.

Fonte: próprio autor, 2023.

Após a análise do gráfico, é encontrada bifurcações logo no princípio da *blockchain* e ao longo do tempo surge-se outros *forks* que são resolvidos rapidamente neste começo da *blockchain*.

5.5 Ataque de maioria com dominância de um minerador com mais de 50% do poder computacional da rede

Este experimento visa simular um ataque de maioria, popularmente conhecido como ataque de 51%, esse tipo de vulnerabilidade tem como característica a dominância de mais de 50% do poder computacional por um ou mais mineradores, neste último caso, necessita-se de uma atuação em conluio.

Em um cenário real de criptomoedas, ao obter mais de 50% do poder computacional da rede, um minerador mal intencionado poderia manipular transações feitas por ele, de forma a acarretar propositalmente gastos duplos (quando um usuário consegue transacionar as mesmas moedas mais de uma vez) e impedir que outros mineradores continuem o processo de mineração normalmente, pois não conseguem criar um *fork* de maior altura do que o fraudado pelo minerador malicioso.

No experimento, o ataque será analisado por meio da comparação na média dos resultados de 30 repetições, com 30 mineradores que buscam minerar 10 mil blocos. Um minerador estará disposto com 51% do poder computacional da rede, portanto ele possuirá a capacidade de minerar blocos mais rapidamente do que todos os outros mineradores, mesmo que ocorra uma atuação de mineração em conluio contra ele. Dessa maneira, quando ele assumir o processo de mineração ele impedirá que outros mineradores continuem seu trabalho e propositalmente, ele colocará blocos fraudados na *blockchain*. Esses blocos fraudados não serão aceitos pelos outros mineradores, mas como ninguém irá retirar o fraudador de sua atuação na *blockchain*, espera-se que sua *blockchain* atinja os 10 mil blocos minerados primeiro do que o restante da rede, o que faz com que seu *fork* seja o principal por possuir a cadeia mais longa.

Para que seja possível realizar o ataque de maioria de todos os tipos e com qualquer quantidade de mineradores, é aconselhável que haja uma equação matemática que possibilite descobrir o poder computacional dos mineradores que violarão a rede. Dessa forma, diferentes casos podem ser explorados, por exemplo, um ataque de 51% com um único minerador ou um ataque de 60% com dois mineradores.

Em casos com mais de um minerador que participa do ataque, deve ser informada a porcentagem que cada fraudador deve possuir individualmente, ou seja, para um ataque de 60% com dois mineradores, deve ser informado que cada fraudador possuirá 30% de poder computacional cada. A partir dessas premissas, a equação é definida por 4 variáveis:

a) **PCR (Poder Computacional da Rede)** - O poder computacional da rede faz referência a soma de todos os mineradores presentes na rede sem contar com os fraudadores. No experimento proposto nesta seção e de acordo com a análise feita na Tabela 11 a soma do PCR é de 1357, ou seja, para este experimento, o poder

computacional dos 29 mineradores honestos é de 1357 e se refere aos 49% do poder total existente entre os 30 mineradores, assim sabe-se a quantidade equivalente à parte honesta da rede;

b) **QF – Quantidade de Fraudadores** - A quantidade de fraudadores se refere ao número de mineradores que fraudarão a *blockchain*, ou seja, os mineradores que participarão do ataque de maioria;

c) **PPCCF – Porcentagem do Poder Computacional de Cada Fraudador** - A porcentagem de cada fraudador se refere à porcentagem de poder computacional que o(s) minerador(es) fraudador(es) deve(m) ter individualmente para completar o ataque, pois ela é acompanhada da quantidade de fraudadores, se um ataque for de 60% para dois mineradores, essa variável deve estar preenchida com apenas 30% (0,30) pois será multiplicada pela quantidade de fraudadores, assim, 30% (0,30) multiplicado por 2 é 60% (0,60) e o ataque pode ser realizado corretamente;

d) **PCMF - Poder computacional do minerador fraudador** - Se refere ao resultado da equação, que é o poder computacional ideal para cada fraudador.

$$PCMF = \frac{\frac{PCR(PPCCF*QF)}{1-(PPCCF*QF)}}{QF} + 1$$

Substituindo os valores para o ataque de 51% de um único minerador tem-se:

$$PCMF = \frac{1357(0.51 * 1)}{1 - (0.51 * 1)} + 1$$

$$PCMF = 1413{,}\cancel{38}$$

A Tabela 11 exibe o resultado médio das 30 execuções do experimento.

Tabela 11 – Após 30 repetições, a tabela apresenta a média dos dados para os 30 mineradores que realizaram a mineração de 10000 blocos. Um minerador domina mais de 50% do poder computacional da rede.

Poder computacional (PC)	**Média de Blocos Minerados (BM)**	**Média da Razão (BM/PC)**
2	1,03	0,5
3	1,1	0,37
3	1,63	0,54
6	3	0,5
11	5,83	0,53
14	6,36	0,46
16	6,83	0,43
20	8,93	0,45
21	7,93	0,38
29	12,13	0,42
35	15,63	0,45
38	16,36	0,43
42	17,43	0,42
44	19,03	0,43
47	20,8	0,44
47	21,06	0,45
62	25,13	0,41
63	25,63	0,41
64	26,23	0,41
69	29,9	0,43
70	29,13	0,42
71	29,43	0,41
74	31,03	0,42
74	31,73	0,43
76	31,5	0,41
86	36,4	0,42
88	36,33	0,41
91	38,86	0,43
91	39,46	0,43
1413	9424,06	6,67

Fonte: próprio autor, 2023.

Ao observar a Tabela 11, nota-se uma grande diferença na relação de poderes entre o minerador mais poderoso (fraudador) e o restante da rede (mineradores honestos). O minerador mais poderoso consegue ter maior poder computacional do que todo o restante dos mineradores juntos, proporcionalmente essa divisão pode ser observada no gráfico da Figura 16.

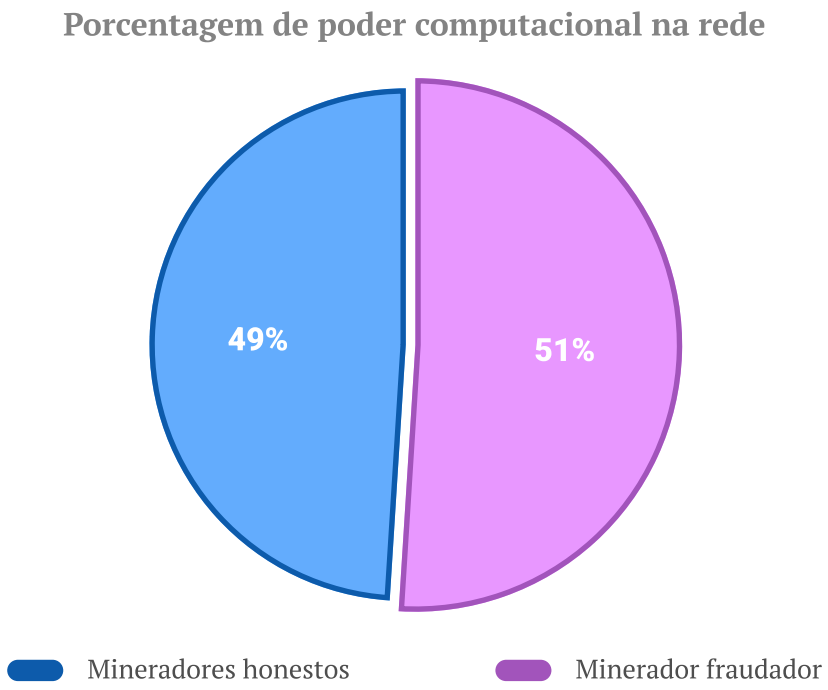

Figura 16 – Porcentagem de poder computacional na rede.

Fonte: próprio autor, 2023.

Devido ao fato dele possuir uma maior capacidade computacional, ele conseguiu minerar mais blocos do que o restante da rede durante as a média das 30 repetições, mas ressalta-se que a quantidade de blocos minerados por ele é abundantemente superior, que chega a 94,24%. Os 100% de blocos minerados não são alcançados pois ele não realiza a mineração de forma egoísta, assim, recebe blocos de mineradores que conseguiram minerar no momento inicial da *blockchain*. O gráfico da Figura 17 traz a informação relacionada ao histórico de mineração, por meio dele é possível comparar a quantidade de blocos honestos com a quantidade de blocos fraudados de acordo com a altura em que a *blockchain* se encontrava.

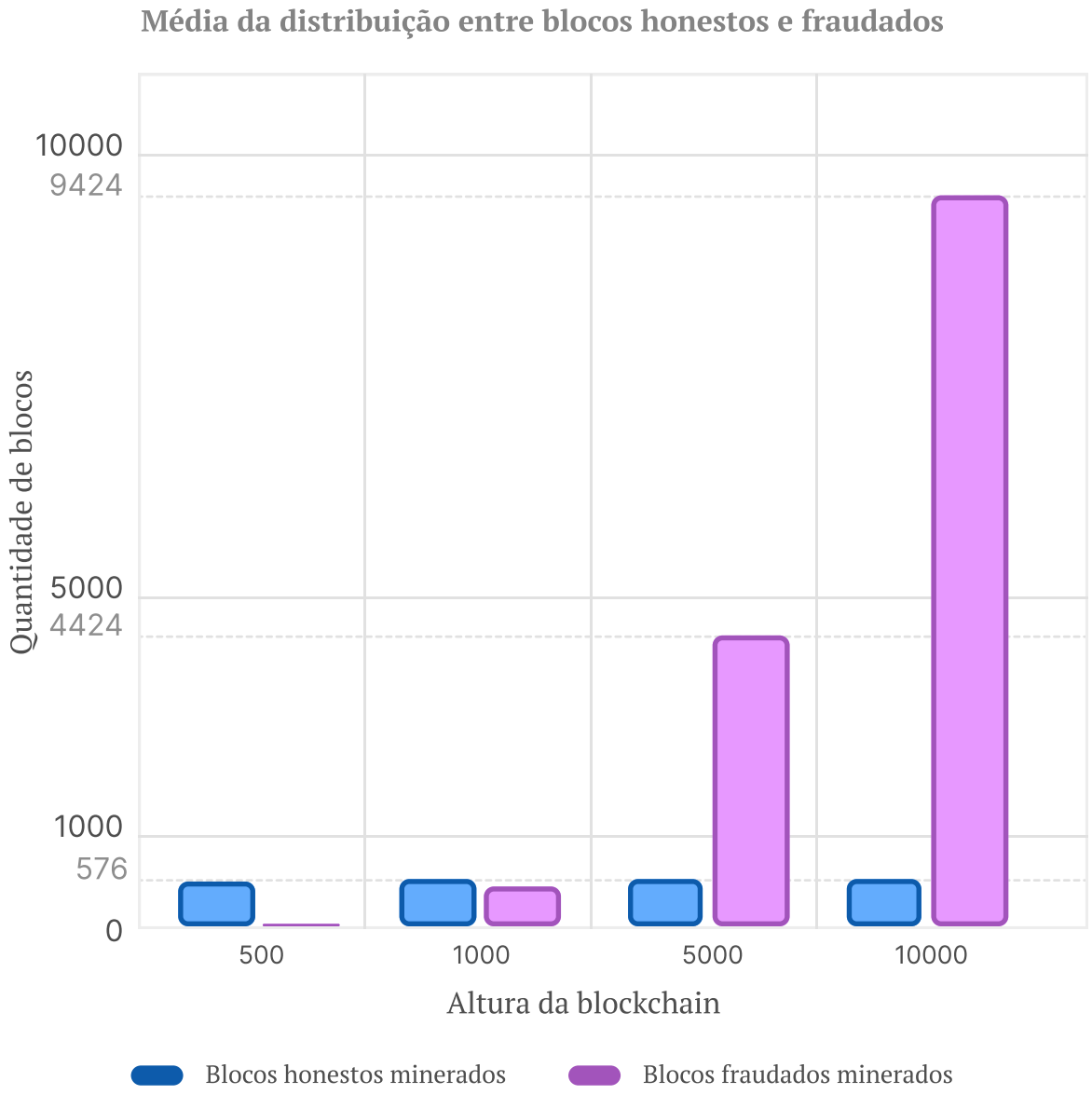

Figura 17 – Relação entre blocos honestos e blocos fraudados de acordo com o resultado médio de 30 repetições.

Fonte: próprio autor, 2023.

Observa-se que a partir do momento que o minerador fraudador começa a minerar seus blocos, somente ele minera na *blockchain* principal, isso se deve ao fato de sua dominância na rede ser superior a de todos os outros, ninguém consegue minerar blocos com maior rapidez do que ele.

Por mais que todos os mineradores honestos não aceitam os blocos fraudados, eles não expulsam o minerador fraudador e assim sua *blockchain* é considerada a de maior altura na rede e qualquer outro trabalho de mineração se torna impossibilitado de conseguir tomar a frente deste minerador por meio da prova de trabalho.

5.6 Falha no ataque de maioria com 60% do poder computacional da rede dividido entre dois mineradores concorrentes

Este experimento visa simular um ataque de maioria em que dois mineradores mal intencionados possuem por volta de 60% do poder computacional da rede juntos, porém executarão o trabalho de mineração de forma individual, ou seja, nenhum dos dois irá aceitar blocos fraudados pelo outro, assim a divisão do experimento irá contar com 2 mineradores com 30% do poder computacional da rede cada um.

Neste experimento, esses dois mineradores serão os mais poderosos da rede, porém nenhum nó aceitará blocos fraudados, assim, os fraudadores não poderão compartilhar blocos nem mesmo entre si, o que faz com que um bloco fraudado só possa ser inserido na própria *blockchain* do fraudador.

Pelo fato dos fraudadores não atuarem em conluio e continuarem a aceitar blocos honestos de outros mineradores, diferentemente do ataque de 51% esses mineradores não dominarão a rede, pois a parte honesta terá por volta de 40% do poder computacional da rede e cada um dos fraudadores terão apenas 30%. Pelo motivo citado, devem minerar uma quantidade de blocos inferior ao restante da rede já que não conseguirão criar um *fork* que dominará a parte honesta da rede.

A equação criada para o experimento anterior na seção 5.5 - Ataque de maioria com um minerador dominando mais de 50% do poder computacional da rede - é reaproveitada para esse experimento com a alteração das seguintes variáveis:

$$PCMF = \frac{\frac{1367(0.30*2)}{1-(0.30*2)}}{2} + 1$$

$$PCMF = 1026{,}\cancel{25}$$

A Tabela 12 exibe a média para o resultado do experimento realizado com 30 repetições.

Tabela 12 – Após 30 repetições, a tabela apresenta a média dos dados para os 30 mineradores que realizaram a mineração de 10000 blocos. No experimento, 2 mineradores possuem 60% do poder computacional da rede, divididos em 30% cada um.

Poder computacional (PC)	**Média de Blocos Minerados (BM)**	**Média da Razão (BM/PC)**
10	72,3	7,2
10	71,1	7,1
23	166,27	7,23
30	223,56	7,45
33	241,87	7,33
34	249,2	7,33
38	274,13	7,21
39	285,2	7,31
42	308,23	7,34
42	310,1	7,38
44	324,76	7,38
45	331,8	7,37
46	336,83	7,32
46	334,63	7,28
47	347,56	7,39
47	345,06	7,34
55	404	7,35
56	404,26	7,22
57	415,86	7,3
57	419,83	7,36
60	432,4	7,21
64	470	7,34
64	467,13	7,3
64	466,43	7,29
65	470,06	7,23
66	481,33	7,29
87	630	7,24
96	698,8	7,28
1026	4,06	0,004
1026	13	0,012

Fonte: próprio autor, 2023.

Ao observar a Tabela 12, nota-se dois mineradores mais poderosos do que o restante dos mineradores da rede. Essa divisão de poderes na rede é demonstrada no gráfico da Figura 18.

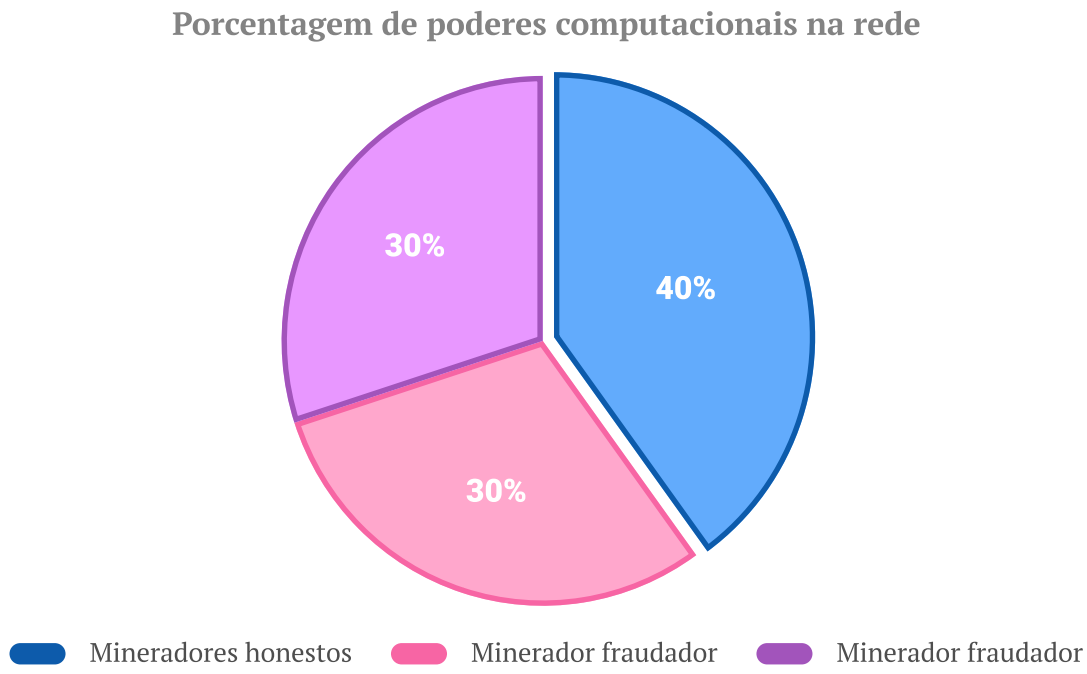

Figura 18 – Porcentagem de poderes computacionais na rede.

Fonte: próprio autor, 2023.

O gráfico da Figura 18 busca exibir as informações referentes a divisão dos poderes computacionais da rede, ele isola os mineradores fraudadores com apenas 30% e mostra a dominância dos mineradores honestos com 40% de poder computacional. Mesmo com uma parte menor da rede, fica evidente como a união dos honestos os beneficiam para propagarem com maior dominância suas atualizações, já que individualmente supera os fraudadores individuais na média dos resultados para as 30 repetições do experimento. Por meio do gráfico da Figura 19 será possível verificar a média da quantidade de blocos honestos e fraudados minerados.

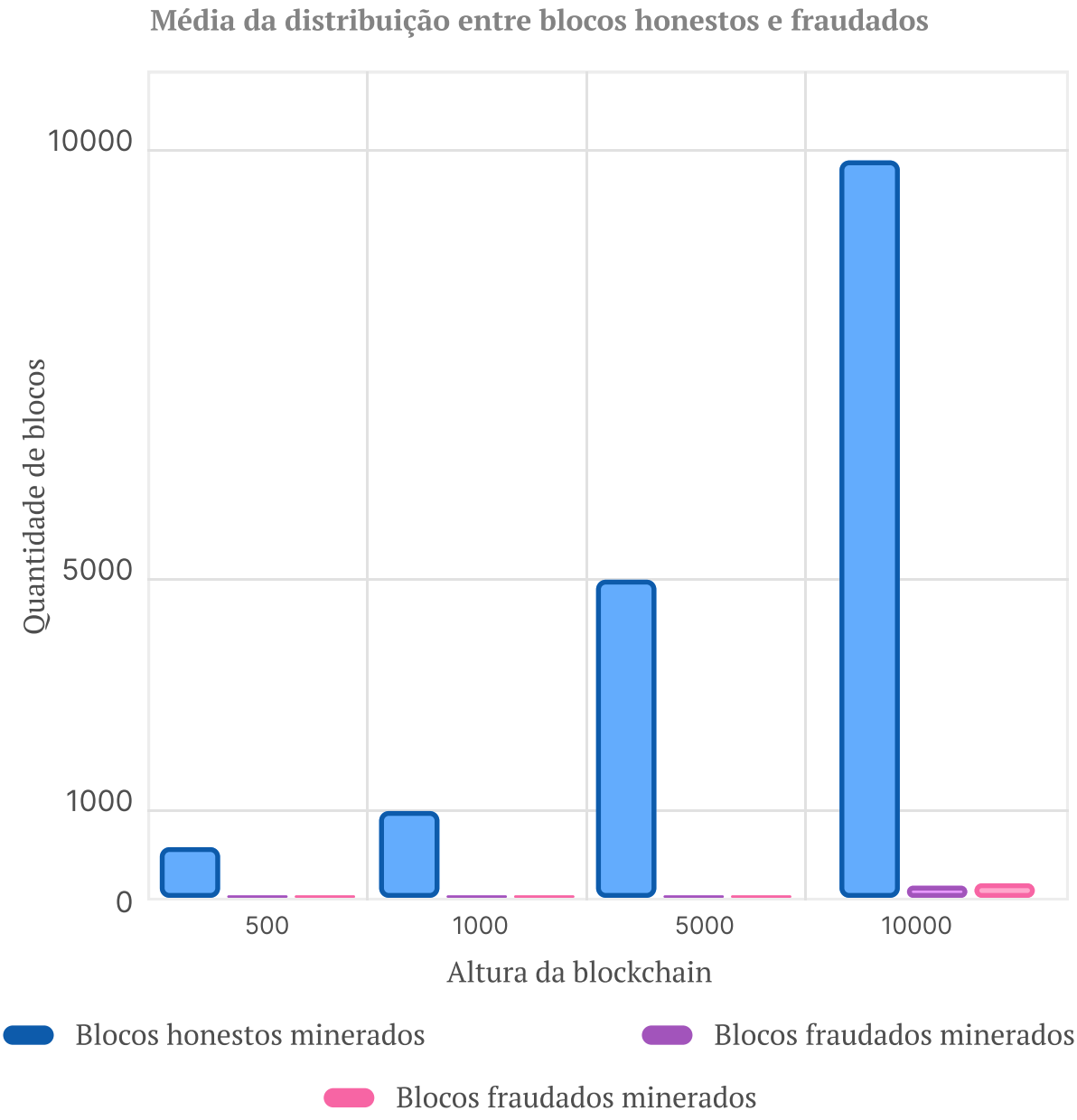

Figura 19 – Relação entre blocos honestos e blocos fraudados de acordo com o resultado médio de 30 repetições para um ataque sem conluio.

Fonte: próprio autor, 2023.

Ao retornar para a análise da Tabela 12 e do gráfico da Figura 19, observa-se que os dois mineradores mais poderosos antagonizam o experimento anterior já que foram os que mineraram a menor quantidade de blocos da rede, isso ocorre pois tentam propagar blocos fraudados de forma individual e não em conluio, assim não dominam a rede e os mineradores honestos que possuem apenas 40% do poder computacional da rede se sobressaem em relação aos mineradores mais poderosos que atuaram de forma individual. O motivo dos fraudadores possuírem alguns blocos minerados é devido ao fato de algu-

mas das execuções possuírem *blockchains* que conseguiram ser as primeiras a chegarem aos 10000 blocos minerados em algumas repetições. Por possuir um poder de mineração bastante elevado, esses são os últimos blocos inseridos. Porém somente eles possuirão esses últimos blocos, os blocos fraudados nunca serão aceitos por outros mineradores e caso a *blockchain* continuasse no experimento, além de 10 mil blocos, esses blocos seriam inválidos para o restante dos mineradores e dariam sequência corretamente.

6 Conclusão

Este trabalho buscou realizar a criação de uma biblioteca peso-leve a fim de simular o consenso com o algoritmo de PoW em ambientes baseados na tecnologia *blockchain*, essa solução deveria ser criada por meio de uma linguagem de programação com maior facilidade de abstração das informações, para que seja um recurso que possa colaborar com o ambiente científico e simultaneamente com entusiastas do assunto. A partir dessas premissas foi realizada a pesquisa e documentação dos principais conceitos sobre *blockchain* que deveriam ser codificados.

Como conceito fundamental da maioria das *blockchains*, a estrutura de uma rede de mineração foi codificada com mineradores ativos que concorrem entre si a cada novo bloco a ser minerado, todos os mineradores que forem aptos a minerar, ao realizar este processo devem utilizar a metodologia de prova de trabalho, que realiza a conversão de um *hash* inválido para outro *hash* válido sem danificar o conteúdo dos dados colocados no bloco. Para que um minerador possa encontrar a prova do trabalho em meio ao seus concorrentes, é interessante que ele possua uma capacidade de recursos computacionais elevada, para obter uma vantagem e conseguir realizar o processo de mineração antes dos demais, nota-se nos experimentos realizados que a principal influência para uma mineração ocorrer é o poderio computacional do minerador, como resultado, mineradores mais poderosos costumeiramente possuem uma maior taxa de mineração de blocos do que mineradores com menor capacidade computacional, salvos os casos em que aspectos de vulnerabilidades foram explorados, como o ataque de maioria mal sucedido, em que os dois mineradores mais poderosos que criavam blocos fraudados, não conseguiram tanto sucesso diante da rede honesta.

Após a mineração, o objetivo do minerador é sinalizar aos seus vizinhos que existe uma nova *blockchain* mais atualizada, dessa forma, naturalmente, devem ocorrer *forks* (bifurcações, trifurcações ou mais formas de divisão de *blockchains* válidas). Esses *forks* são resolvidos em uma próxima mineração realizada apenas por um único nó, a análise dos *forks* e o desfecho do experimento de mineradores que possuem poucos vizinhos contra mineradores que possuíam muitos vizinhos, foi possível concluir que mineradores com uma maior quantidade de vizinhos possuíam maior influência no momento em que disputavam *forks* com mineradores que possuíam poucos vizinhos, o que relaciona a importância do número de vizinhos para uma mineração efetiva.

Esses conceitos foram abordados nos experimentos relacionados às variações e não variações de poderes computacionais, variações e não variações da quantidade de vizinhos e análises de *forks* de cada um desses experimentos. Para cada experimento que

é demonstrado, o resultado permite entender as principais características que envolve uma rede *blockchain* baseada em prova de trabalho.

Além dos conceitos fundamentais a serem desenvolvidos, para estudar alguns aspectos de vulnerabilidades em *blockchains*, a biblioteca explora um dos conceitos de segurança abordados na monografia e a escolha foi explorar o ataque de 51% ou ataque de maioria, ao criar experimentos relacionados a atacar a rede com mais de 50% do poder computacional, tem-se um experimento de dominação da rede por um único minerador que frauda seus blocos, nele pode-se observar uma soberania na mineração dos blocos e outro experimento com dois mineradores que fraudam blocos e dividem um poder de maioria de forma concorrente e não em conluio, o que resulta em uma fraca adesão de seus blocos na *blockchain* principal. Dessa maneira conclui-se que um ataque de maioria só funciona quando os atacantes estão agindo de forma cooperativa.

Todos os experimentos abordados testam e comprovam a funcionalidade da biblioteca para exemplificar de forma prática os conceitos redigidos durante a monografia, eles são analisados e explicados de acordo com seus intuitos de demonstração.

Para futuros projetos de pesquisas, o aprimoramento da biblioteca em adaptações de acordo com a forma de obtenção de consenso para diferentes tipos de mecanismos, como prova de autoridade, prova de participação podem agregar a solução para diversos tipos de tecnologia que abordam o conceito de *blockchain*. A biblioteca também suporta o acréscimo de experimentos como mineração egoísta e testes de algoritmos para obtenção de consenso que ainda não existem, mas podem ser criados com a inserção e manipulação de alguns trechos de código.

7 Referências

ACADEMY, B. *O que é um Ataque de 51%?* 2018. Disponível em: <https://academy.binance.com/pt/articles/what-is-a-51-percent-attack>. Acesso em: 21 nov. 2022. Citado na página 30.

AGRAWAL, N. et al. BlockSim-Net: A Network Based Blockchain Simulator. *Ashoka University and Koç University*, p. 1–5, 2020. Citado na página 33.

ALHARBY, M.; MOORSEL, A. v. BlockSim: An Extensible Simulation Tool for Blockchain Systems. *School of Computing, Newcastle University, Newcastle upon Tyne, United Kingdom. Department of Computer Science, Taibah University, Medina, Saudi Arabia*, p. 1–16, 2020. Disponível em: <https://doi.org/10.3389/fbloc.2020.00028>. Citado na página 33.

ANTONOPOULOS, A. M. *Mastering Bitcoin: Programming the Open Blockchain*. [S.l.: s.n.], 2017. ISBN 978-1491954386. Citado 4 vezes nas páginas 12, 13, 18 e 26.

BINANCE. *Binance Academy*. 2020. Disponível em: <https://academy.binance.com/pt>. Acesso em: 10 jan. 2023. Citado na página 32.

BROWNWORTH, A. *Blockchain Demo*. 2016. Disponível em: <https://andersbrownworth.com/blockchain>. Acesso em: 25 nov. 2022. Citado 2 vezes nas páginas 31 e 32.

CHICARINO, V. R. L. Uma heurística para a detecção de ataques ao mecanismo de consenso por Prova de Trabalho em Blockchain. *Universidade Federal Fluminense*, p. 38–46, 2019. Citado 2 vezes nas páginas 22 e 24.

FARIA, C. S. F. BlockSim: Blockchain Simulator. *Instituto Superior Técnico da Universidade de Lisboa*, p. 1–72, 2018. Disponível em: <https://doi.org/10.1109/Blockchain.2019.00067>. Citado na página 33.

GOBEL, J. et al. Bitcoin Blockchain Dynamics: the Selfish-Mine Strategy in the Presence of Propagation Delay. *Department of Informatics, University of Hamburg, 22527 Hamburg, Germany; Department of Mathematics and Statistics, University of Melbourne, Vic 3010, Australia; Department of Mathematical Sciences, Stellenbosch University, 7600 Stellenbosch, South Africa*, p. 3, 2015. Citado na página 30.

GREENBERG, A. *Crypto Currency*. 2011. Disponível em: <https://www.forbes.com/forbes/2011/0509/technology-psilocybin-bitcoins-gavin-andresen-crypto-currency.html>. Acesso em: 26 out. 2022. Citado na página 17.

GUPTA, M. *Blockchain For Dummies, 2nd IBM Limited Edition*. [S.l.: s.n.], 2018. ISBN 978-1-119-54593-4. Citado na página 21.

HAFF, G.; HENRY, W. *From Pots and Vats to Programs and Apps: How Software Learned to Package Itself*. [S.l.: s.n.], 2017. ISBN 978-1548927264. Citado na página 12.

HAWKES, P.; PADDON, M.; ROSE, G. G. On Corrective Patterns for the SHA-2 Family. *Qualcomm Australia, Level 3, 230 Victoria Rd, Gladesville, NSW 2111, Australia*, p. 1–2, 2004. Citado na página 23.

LAMOUNIER, L. *Algoritmos De Consenso: A Raiz Que Sustenta A Tecnologia Blockchain*. 2018. Disponível em: <https://101blockchains.com/pt/algoritmos-de-consenso/>. Acesso em: 05 jul. 2022. Citado na página 13.

MERKLE, R. C. Secrecy, authentication, and public key systems. *Stanford Electronics Laboratories, Department of Electrical Engineering, Stanford University, Stanford / CA - 94305*, p. 1–182, 1979. Citado na página 22.

NAKAMOTO, S. Bitcoin: A Peer-to-Peer Electronic Cash System. p. 1–9, 2008. Citado 3 vezes nas páginas 13, 16 e 17.

PANDEY, S. et al. BlockSIM: A practical simulation tool for optimal network design, stability and planning. p. 1–6, 2019. Disponível em: <https://doi.org/10.1109/BLOC.2019.8751320>. Citado na página 33.

PENARD, W.; WERKHOVEN, T. van. On the Secure Hash Algorithm family. *Stanford Electronics Laboratories, Department of Electrical Engineering, Stanford University, Stanford / CA - 94305*, p. 2–3, 2008. Citado na página 23.

RANGER, S. *What is cloud computing? Everything you need to know about the cloud explained*. 2022. Disponível em: <https://www.zdnet.com/article/what-is-cloud-computing-everything-you-need-to-know-about-the-cloud/>. Acesso em: 14 mai. 2022. Citado na página 12.

REBELLO, G. A. F. et al. Correntes de Blocos: Algoritmos de Consenso e Implementação na Plataforma Hyperledger Fabric. *Universidade Federal do Rio de Janeiro - GTA/PEE/COPPE, Universidade Federal da Bahia - GAUDI/DCC e Universidade de Brasília - COMNET/CIC*, p. 93–148, 2019. Disponível em: <https://doi.org/10.5753/sbc.471.7.03>. Citado na página 13.

SANTOS, G. H. de A. *biblioteca-blockchain*. 2022. Disponível em: <https://github.com/guilhermehenriquesantos/biblioteca-blockchain>. Acesso em: 19 dez. 2022. Citado na página 45.

SEAN. *Blockchain Demo 2.0*. 2017. Disponível em: <https://blockchaindemo.io/>. Acesso em: 25 nov. 2022. Citado na página 32.

SECRETARIAT, I. G. *Cryptojacking*: Cybercriminals can unknowingly use your computer to generate cryptocurrency. 2020. Disponível em: <https://www.interpol.int/Crimes/Cybercrime/Cryptojacking>. Acesso em: 20 nov. 2022. Citado na página 29.

SERHACK. *Mastering Monero: The future of private transactions*. [S.l.: s.n.], 2018. ISBN 978-1731079961. Citado 3 vezes nas páginas 12, 16 e 17.

SOCIETY, N. G. *Y2K bug*. 2022. Disponível em: <https://education.nationalgeographic.org/resource/Y2K-bug>. Acesso em: 14 mai. 2022. Citado na página 12.

TEKINER, E. et al. SoK: Cryptojacking Malware. *Florida International University*, p. 1–20, 2021. Disponível em: <https://doi.org/10.1109/EuroSP51992.2021.00019>. Citado na página 29.