**Diferentes Funções de Prova de Trabalho em uma Blockchain**

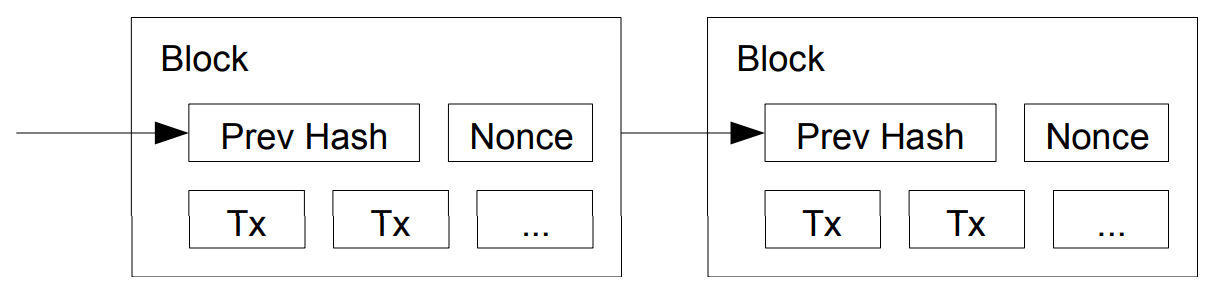
Henrique Chaves Lopes | RA 11202130079

1. **Introdução**

O advento da blockchain, uma tecnologia proposta no artigo publicado por Satoshi Nakamoto (cuja identidade é desconhecida) em 2008, representou uma ruptura no modo como dados sensíveis, como o registro de transações financeiras, são armazenados. Mas, para além das famosas criptomoedas, cujo surgimento foi concomitante à invenção da blockchain, novas aplicações da tecnologia estão sendo desenvolvidas, inclusive no campo da saúde. Fica, portanto, cada vez mais evidente a grande relevância deste tópico.

As blockchains são, em poucas palavras, bancos de dados descentralizados e confiáveis, implementados sobre redes nas quais cada nó representa um usuário. Na blockchain de Nakamoto, não há requisitos para participar da rede (*permissionless*) e cada usuário tem uma cópia do registro de transações. Grupos de transações são guardados em blocos cujos cabeçalhos referenciam, por meio de um hash, o bloco anterior na *chain* (corrente ou cadeia), mantendo a ordem cronológica dos eventos. Embora seja uma rede *permissionless*, e possam haver usuários mal-intencionados, a confiabilidade da blockchain é assegurada por seu algoritmo de consenso e seus mecanismos de provas criptográficas, desde que a maior parte do poder computacional da rede esteja nas mãos de usuários honestos. Pois todo usuário pode minerar blocos — i.e., criar novos blocos e adicioná-los na cadeia — caso resolva uma tarefa computacional, conhecida como prova de trabalho (*Proof of Work*, PoW), mais rápido que os demais. O algoritmo de consenso, por sua vez, garante que todos os nós estejam trabalhando sobre a mesma cadeia, e não sobre um *fork* (bifurcação) gerado por atrasos na rede ou quando dois blocos são minerados ao mesmo tempo.

*Figura 1. Diagrama de uma blockchain*



*Fonte: Satoshi Nakamoto*

Dada a sua importância central nas blockchains, o foco deste trabalho são os mecanismos de prova de trabalho. Uma blockchain foi implementada e diferentes funções de prova de trabalho foram testadas. Assim algumas considerações puderam ser feitas.

1. **Descrição do código**

A implementação da blockchain foi feita com a linguagem de progamação Python, de acordo com o guia explicitado abaixo, nas referências bibliográficas deste trabalho. O código fonte contém a classe **Blockchain**, onde foram definidos:

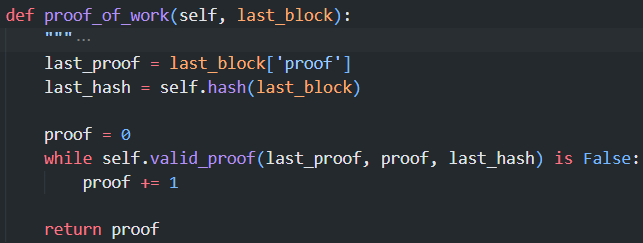
* O método de inicialização, onde cria-se o bloco de gênese.
* O método **new\_block** que cria um novo bloco e cujos parâmetros são: o hash do bloco anterior e o *nonce* (número) da prova de trabalho.
* O método **new\_transaction** que cria uma nova transação e cujos parâmetros são: o endereço do pagador, o endereço do recebedor e o valor da transação.
* O método **hash** que calcula o hash do bloco passado como parâmetro.
* O método **last\_block** que retorna o último bloco da blockchain.
* O método **proof\_of\_work** que recebe, como parâmetro, o último bloco e calcula a prova de trabalho.
* O método **valid\_proof** que valida uma prova de trabalho e cujos parâmetros são: o *nonce* do bloco anterior, o *nonce* atual que será verificado e o hash do bloco anterior.

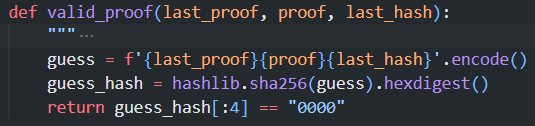
Em outro arquivo, usando a biblioteca Flask do Python, construiu-se uma *Application Programming Interface* (API), permitindo a comunicação entre blockchain e usuário. O usuário pode enviar requisições do protocolo *Hyper Text Transfer Protocol* (HTTP) para URLs específicas do servidor, neste caso local, para realizar cada tarefa desejada, como criar uma nova transação ou minerar um bloco.

1. **Hashcash (SHA-256)**

A função de mineração usada no bitcoin é uma variação do hashcash, proposto para conter *spam* de e-mails e ataques de negação de serviço em 1997 por Adam Back. Nesta função, um número (*nonce*) é incrementado, ou gerado randomicamente, até que o hash de uma string concatenando o próprio *nonce*, o hash e o *nonce* do bloco anterior na blockchain, começe com uma determinada quantidade de zeros. O algoritmo de hash usado é o SHA-256.

*Figura 2 e 3. Implementação do hashcash em Python*





*Fonte: De autoria própria*

A quantidade de zeros representa o nível de dificuldade da operação, visto que o processamento exigido e o tempo de execução aumentam exponencialmente com essa restrição. A exemplo do bitcoin, a dificuldade de sua prova de trabalho é ajustada de tempos em tempos, de modo que o tempo de mineração seja sempre próximo de 10 minutos. O primeiro teste realizado envolveu minerar um bloco em diferentes dificuldades do algoritmo. O *hardware* usado foi um *notebook* particular com um processador Intel i5 da 13ª geração. Na tabela 1 vêem-se os resultados.

*Tabela 1. Teste de dificuldades do hashcash*

|  |  |
| --- | --- |
| Dificuldade | Tempo de mineração |
| 4 zeros | 0,80s |
| 5 zeros | 4,98s |
| 6 zeros | 66,62s |
| 7 zeros | 806,40s ≈ 13,44 min |

*Fonte: De autoria própria*

É importante ressaltar que, do ponto de vista energético, este algoritmo tem se provado muito ineficiente. Atualmente, para minerar seus blocos, a rede de bitcoins consome mais energia elétrica do que uma gama de países. Por isso, a tecnologia tem sido criticada por ambientalistas.

1. **Primecoin**
2. **Conclusão**

\*Algumas funções de prova de trabalho para pesquisar: scrypt (usado no litecoin); equihash (usado no zcash); ethash (ethereum); cuckoo cycle (eficiência energética); blake2b (menor consumo de energia do que o hashcash); primecoin; verifiable delay functions (reduz a vantagem de hardware, mas ainda está sendo desenvolvido).

**Referências**

NAKAMOTO, Satoshi. Bitcoin: a peer-to-peer electronic cash system. **Bitcoin**, 2008. Disponível em: <https://bitcoin.org/bitcoin.pdf>. Acesso em: 12 de dez. de 2024.

CONTERNO, Ivan. Tecnologia blockchain pode reduzir risco de roubos e fraudes na área da saúde. **Jornal da USP**, São Paulo, 29 de mai. de 2023. Disponível em: <https://jornal.usp.br/ciencias/tecnologia-blockchain-pode-reduzir-risco-de-roubos-e-fraudes-na-area-da-saude/>. Acesso em: 10 de dez. de 2024.

GAYVORONSKAYA, T.; MEINEL, C. **Blockchain**: hype or innovation. Springer, 2020.

TRISTIANTO, Bimo Putro. Build your own blockchain in Python: a practical guide. **Medium**, 2022. Disponível em: <https://bimoputro.medium.com/build-your-own-blockchain-in-python-a-practical-guide-f9620327ed03>. Acesso em: 10 de dez. de 2024.

Proof of work. **Wikipedia**, 2024. Disponível em: <https://en.wikipedia.org/wiki/Proof_of_work>. Acesso em: 12 de dez. de 2024.

Hashcash. **Wikipedia**, 2024. Disponível em: <https://en.wikipedia.org/wiki/Hashcash>. Acesso em: 12 de dez. de 2024.

HUYNH, A. N. Q. et al. Energy consumption and bitcoin market. **Asia-Pacifc Financial Markets**, 29, 79-93, 2022. Disponível em: <https://link.springer.com/article/10.1007/s10690-021-09338-4>. Acesso em: 13 de dez. de 2024.

GUIDO, Gabriela. Mineradoras de bitcoin gastaram a mesma quantidade de energia que a Austrália no último ano. **Um Só Planeta**, 10 de fev. de 2024. Disponível em: <https://umsoplaneta.globo.com/energia/noticia/2024/02/10/mineradoras-de-bitcoin-gastaram-a-mesma-quantidade-de-energia-que-a-australia-no-ultimo-ano.ghtml>. Acesso em: 13 de dez. de 2024.