Article

Atividade Prática sobre Blockchain

Pedro Bernardi Alves [1]

[1] Engenharia de Computação; (Bernardi)

1. Introdução

Criar uma aplicação de blockchain local, com um algoritmo de prova por trabalho local onde a dificuldade é definida pelo usuário da aplicação, que permita armazenar os blocos validados, com uma rotina que verifica a integridade dos blocos.

2. Método

O desenvolvimento da blockchain começa pela lista de objetos que o compõe. Ele deve possuir informações básicas, data/hora, transações e hash. E uma parte importante é imutabilidade.

Para construção dessa blockchain, foi utilizado a linguagem C++, pois possui benefícios em termos de polimorfismo para tempo de execução , sobrecarga de funções e multi-threading.

A classe Blockchain é constituida pela dificuldade, possui um número, de forma que o hash de cada bloco contenha zeros à esquerda que correspondem a essa dificuldade. Garantir que o hash de cada bloco comece com o número de zeros definido na dificuldade, requer muito poder de computação. Quanto maior o nível de dificuldade, mais tempo leva para minerar novos blocos.

A partir da cadeia de blocos, defino o hash anterior do novo bloco para ser igual ao hash do último bloco na cadeia, garantindo assim que a cadeia seja à prova de adulteração. O hash utilizado, SHA-256, é uma função de criptografia , que recebe alguma string de texto (armazenada como um valor Unicode) e retorna uma string criptografada de 64 caracteres. Em uma blockchain, o texto que criptografamos é na verdade nosso bloco. O bloco de gênese refere-se ao primeiro bloco criado. Sempre que um bloco é integrado ao restante da cadeia, ele deve referenciar o bloco anterior.

Como as propriedades do novo bloco são alteradas a cada novo cálculo, é importante calcular seu hash criptográfico novamente. Depois de atualizar seu hash, o novo bloco é enviado para o array blockchain. E o bloco é salvo com o uso da blibioteca nlohmann.

Uma característica chave de uma blockchain é que uma vez que um bloco tenha sido adicionado à cadeia, ele não pode ser alterado sem invalidar a integridade do resto da cadeia. Os hashes são críticos para garantir a validade e a segurança de um blockchain porque qualquer alteração no conteúdo de um bloco resultará na produção de um hash totalmente novo e na invalidação do blockchain.

O código tem como base o artigo [1] e a série de videos de [2].

3. Procedimentos

A execução do projeto pode ser dado pelo comando abaixo em terminal.

```
./Make
```

Citation: Atividade Prática sobre Blockchain. *Appl. Sci.* **2022**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

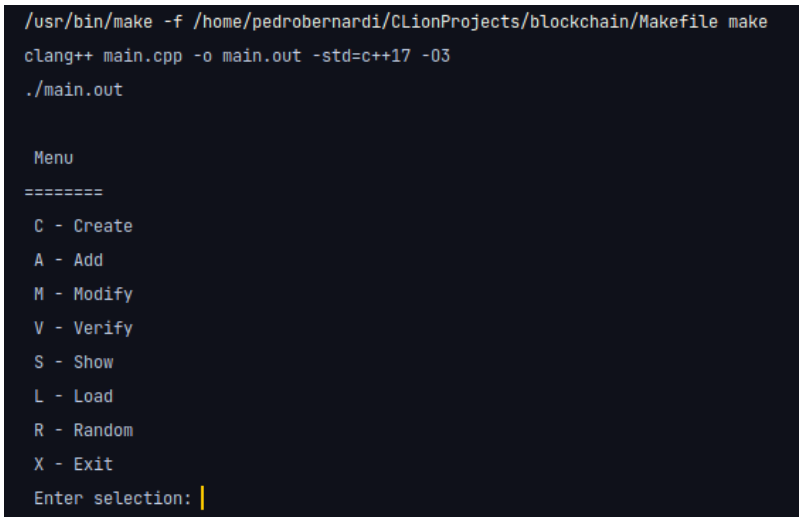

Figure 1

- Create - Criar um novo bloco
  - Digitamos a dificuldade escolhida para criação de nosso bloco
  - E seguinte, o número máximo de transações
- Add - Adicionar uma transação
  - O valor a ser feito na transação
  - Digitar o remetente
  - Digitar o destinário
- Modify - Modificar uma transação
  - O valor a ser alterado na transação
  - Digitar o indice do bloco
  - Digitar o indice da transação
- Verify - Verificar integridade dos blocos
- Show - Visualizar os blocos
- Load - Carregar blocos do sistema de armazenamento
- Random - Carregar dados aleatórios para testes

4. Resultados

A criação de um blockchain com dificuldade 2 e número máximo de transições, sendo salvo em um arquivo json, e também pode viszualdo a transação genesis, sendo provado ser o primeiro bloco.


```
    {
"blocks": [
    {
        "blockHash": "",
        "difficulty": 2,
        "index": 0,
        "maxTransactionCount": 2,
        "minedAt": "not mined yet",
        "nounce": 0,
        "previousBlockHash": "81ddc8d248b2dccdd3fdd5e84f0cad62b08f2
        "timeMined": 139982780073304,
        "transactions": [
            {
                "amount": 0.0,
                "receiverKey": "Genesis",
                "senderKey": "Genesis",
```

```
                    "timestamp": "Tue May 17 21:23:30 2022"
                }
            ]
        }
    ],
    "difficulty": 2,
    "maxTransactionsCount": 2
}
```


A escolha de dificuldade altera o tempo de mineração dos blocos, através da adição de transações pela escolha da opção Random no menu, podemos obter o tempo de mineração desses blocos.

A partir de cada dificuldade, foi feito a mineração de 10 blocos, e gravado em um arquivo json. A partir do arquivo json, foi utitilizado a blibioteca matplotlib do python, e assim verificando uma média em milisegundos dos blocos. As dificuldades escolhidas foram 2, 3, 4 e 5, outras dificuldades foram descartas dado ao hardware.

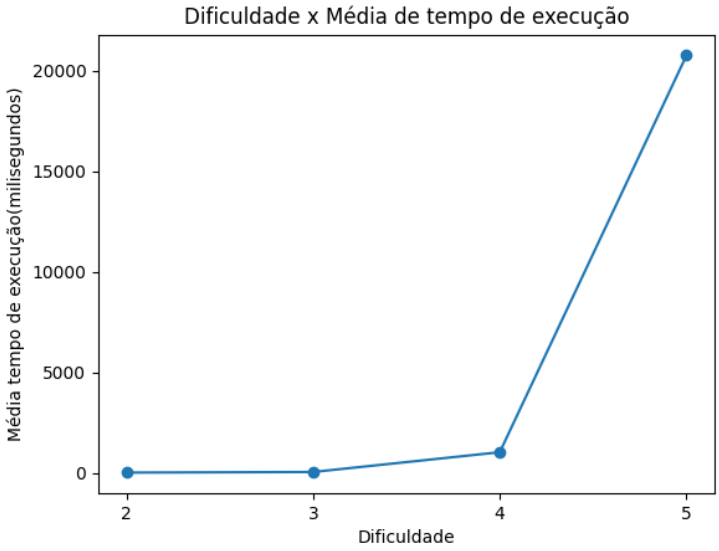

Figure 2

A veririficação do integridade do bloco foi realizada da seguinte forma:

Figure 3

Como mostrado na imagem, o bloco é válido, portando dentro do esperado pois não houve nenhuma alteração.

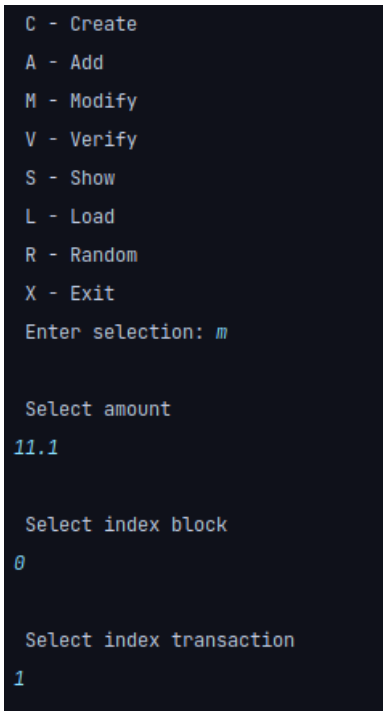

Figure 4

Figure 5

Dada a modificação do primeiro bloco, e a segunda transação para o valor 11.1, observamos que já não temos um bloco válido.

5. Conclusões

A prova de trabalho visa identificar um número que encontra uma solução para um problema matemático complicado após a conclusão de uma certa quantidade de trabalho de computação. A ideia principal do trabalho de prova é que qualquer participante da rede blockchain deve achar esse número difícil de identificar, mas facilmente verificável. Consequentemente, desencoraja o spam e a adulteração da estrutura do blockchain. O seguinte trabalho pode ser ajustado para atender em rede, e assim ter uma prova de trabalho mais robusta. Os testes realizados foram realizados num processador i5-10210u, melhores resultados podem ser obtidos em sistemas superiores.

References

1. Social Network for Programmers and Developers, 2022.
2. Explained, S. Creating a blockchain with Javascript (Blockchain, part 1), 2017.