Atividade Prática sobre Blockchain - Segurança e Auditoria de Sistemas

Caio Augusto de Souza Muniz, 2050889

4 de julho de 2022

1 Introdução

A aplicação relatada neste relatório de atividade assíncrona tem como objetivos:

- Criação de uma aplicação de blockchain local
- Implementação de um algoritmo de prova por trabalho local onde a dificuldade é definida pelo usuário da aplicação
- Armazenamento os blocos validados em um arquivo
- Implementação de rotina de verificação de integridade dos blocos.
- O conteúdo presente nos blocos são *strings*.

2 Ferramentas

Para o desenvolvimento da aplicação foram utilizadas as seguintes ferramentas:

- A linguagem escolhida foi o *Javascript*, através do compilador *Node.js* em sua versão 16.15;
- Foi utilizado como base o repositório *Savjee/SavjeeCoin*, disponível no *GitHub*.

3 Metodologia

O código-fonte da aplicação está disponível em: *caioasmuniz/local-blockchain*

A interface da aplicação é feita pela linha de comando, esta apresenta um menu com as seguintes opções:

3.1 Ler blockchain de um arquivo;

3.2 Minerar blocos pendentes;

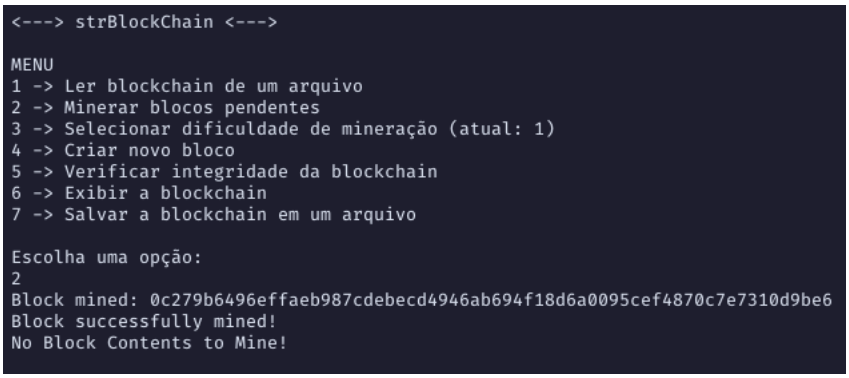

Figure 1: Captura de tela da função de mineração do bloco

3.3 Selecionar dificuldade de mineração;

Figure 2: Captura de tela da função de seleção da dificuldade de mineração

3.4 Criar novo bloco;

Figure 3: Captura de tela da função de criação de um bloco

3.5 Verificar integridade da *blockchain*;

Figure 4: Captura de tela da função de verificação de integridade da *blockchain*

3.6 Exibir a *blockchain*;

```
<---> strBlockChain <--->

MENU
1 -> Ler blockchain de um arquivo
2 -> Minerar blocos pendentes
3 -> Selecionar dificuldade de mineração (atual: 1)
4 -> Criar novo bloco
5 -> Verificar integridade da blockchain
6 -> Exibir a blockchain
7 -> Salvar a blockchain em um arquivo

Escolha uma opção:
6
case 6
[
  Block {
    previousHash: '',
    timestamp: 1483228800000,
    blockContent: 'Genesis Block',
    nonce: 0,
    hash: 'c5ad0b3062fc67060065d7f7b9fb84c7892c677ed0f3006fe26add1dd9c231dd'
  },
  Block {
    previousHash: 'c5ad0b3062fc67060065d7f7b9fb84c7892c677ed0f3006fe26add1dd9c231dd',
    timestamp: 1656976686106,
    blockContent: 'teste',
    nonce: 3,
    hash: '03338cbf9f31cf174651762f87418127f021b5c93989c2c045abdea213ea6a09'
  }
]
```

Figure 5: Captura de tela da função exibição da *blockchain*

3.7 Salvar a *blockchain* em um arquivo;

Figure 6: Conteúdo do arquivo contendo a *blockchain*

4 Análise do Impacto da Dificuldade

Para a observação do impacto da dificuldade no tempo de execução do algoritmo, foi utilizado o *script* "*test-difficulty.js*", presente no repositório da atividade. O *script* continuamente insere e minera uma quantidade de blocos na *blockchain*, enquanto monitora e loga a duração de cada inserção. Este processo foi realizado para cada uma das dificuldades de 1 a 6, tendo 20 blocos gerados e minerados em cada uma das dificuldades. Após a execução do script, foi gerada a seguinte tabela com média e desvio padrão do tempo de execução:

Dificuldade	Tempo Médio de execução (ms)	Desvio Padrão (ms)
1	0.3	0.4775
2	2.95	3.4301
3	15.4	15.6277
4	287.4	207.9394
5	4366.15	4004.0907
6	87177.7	59671.2204

Table 1: Tabela de tempo médio de execução e desvio padrão de acordo com a dificuldade

Além disso, foi gerado o seguinte gráfico do tempo médio de mineração pela

dificuldade:

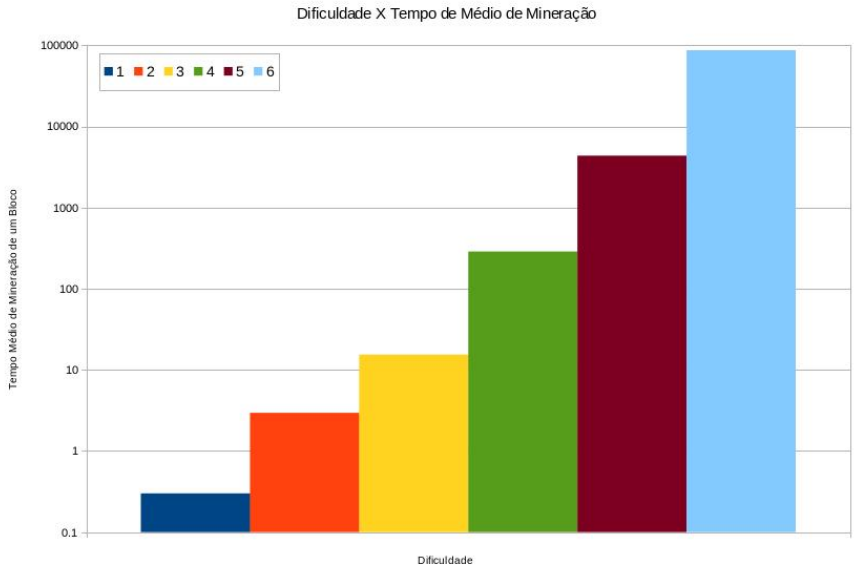

Figure 7: Gráfico de Dificuldade pelo tempo de mineração

A partir dos dados coletados, é possível se observar uma relação direta entre a dificuldade de mineração de um bloco e o tempo de mineração deste. Pode-se inferir também que esta relação é exponencial, tendo em vista a escala logaritmica utilizada para a representação dos dados no gráfico.