Universidade do Minho
Criptografia e Segurança da Informação
Departamento de Informática

Engenharia de Segurança

TP5 - Aula 8

Grupo 7
Carlos Pinto Pedrosa A77320
José Francisco Gonçalves Petejo e Igreja Matos A77688
31 de Março de 2019

Conteúdo

1 Blockchain

1.1 Pergunta 1.1

Depois de analisar o código fonte, verificou-se que o método que cria o *Genesis Block* se chama *createGenesisBlock()*. Assim, para responder ao enunciado apenas foi necessário alterar o corpo esse método para o seguinte:

```
createGenesisBlock(){
    var today = new Date();
    var dd = String(today.getDate()).padStart(2, '0');
    var mm = String(today.getMonth() + 1).padStart(2, '0');
    var yyyy = today.getFullYear();

    today = dd + '/' + mm + '/' + yyyy;

    return new Block(0, today, "Bloco inicial da koreCoin", "0");
}
```

Como se pode observar pelo código acima, apenas se constrói a data atual e se cria um bloco com essa data e com a informação que é o bloco inicial da *koreCoin*.

1.2 Pergunta 1.2

Para responder à segunda pergunta, apenas foi necessário localizar o código onde eram adicionados blocos à *Blockchain* e adicionar mais alguns e mais transações a cada bloco.

```
let koreCoin = new Blockchain();
koreCoin.addBlock(new Block (1, "20/03/2019", {amount1: 20, amount2: 40}));
koreCoin.addBlock(new Block (2, "21/03/2019", {amount1: 60, amount2: 80}));
koreCoin.addBlock(new Block (3, "22/03/2019", {amount1: 100, amount2: 120,
    amount3: 140}));
koreCoin.addBlock(new Block (4, "23/03/2019", {amount1: 160, amount2: 180,
    amount3: 200}));
koreCoin.addBlock(new Block (5, "24/03/2019", {amount1: 220, amount2: 240,
    amount3: 260, amount4: 280}));
```

2 Proof of Work Consensus Model

2.1 Pergunta 2.1

Para a realização desta atividade, foi necessário analisar o *source-code* para encontrar o que queríamos alterar. Assim, depois de devidamente analisado, verificou-se que a dificuldade era alterada numa variável com o mesmo nome. De seguida, a dificuldade foi alterada para os valores 2, 3, 4 e 5 e verificados os tempos.

```
cpp@MacBookPro:koreCoin$ time node main.experiencia2.1.js
Mining block 1...
Block mined: 007abd29e589b6a35f6107095cbc8a3628b7a8ff9ac5f7a203f79b262fff077e
Mining block 2...
Block mined: 009d3e5e004957e04604a4032cc58a97f33adb6711f93bc7514de8fe4a531a37
Mining block 3...
Block mined: 004bb82a658a99fd8c9f0d40ec9e69d3a441dacb6383242659b1ff8f7b42195b

real    0m0.245s
user    0m0.125s
sys     0m0.045s
cpp@MacBookPro:koreCoin$ time node main.experiencia2.1.js
Mining block 1...
Block mined: 0007c25169f5256ebb80fcbb423b582a8e699a759e41abb28906295973f9de4c
Mining block 2...
Block mined: 0004af57b70136e9f4b5309f92d518b3f297168d4cf47d4f9354808daadce457
Mining block 3...
Block mined: 00075796ab77bb5289bb9a952b91047482b3348d54d915ff5e5bce28256d01fe

real    0m0.377s
user    0m0.383s
sys     0m0.033s
cpp@MacBookPro:koreCoin$ time node main.experiencia2.1.js
Mining block 1...
Block mined: 0000023168f87d968813b22c4dc92f60c127ff5084af8487d913d497ea7a7900
Mining block 2...
Block mined: 00008e0c291aaf728e015855328b14e651231cce209e6413503fd299e0df6c5e
Mining block 3...
Block mined: 0000fb4a126ef4c1c3c93bf2ed25e8db4c7da2ec89a46aad4f7bf092afd8b6b4

real    0m1.630s
user    0m1.703s
sys     0m0.057s
cpp@MacBookPro:koreCoin$ time node main.experiencia2.1.js
Mining block 1...
Block mined: 0000023168f87d968813b22c4dc92f60c127ff5084af8487d913d497ea7a7900
Mining block 2...
Block mined: 000000b950a180294edddf4340a2d5834119a41bf89e4b2027f341f0fc02365e
Mining block 3...
Block mined: 0000088dc9c3115ee6ab7e95d2e8836f932d75d303ab451a825241acde589a58

real    0m20.761s
user    0m21.549s
sys     0m0.361s
```

Figura 1: Mineração de Blocos com grau de dificuldade 2, 3, 4 e 5

Para melhor visualização dos resultados, estes podem ser encontrados na tabela abaixo.

Dificuldade	2	3	4	5
Tempo	0.245s	0.377s	1.630s	20.761s

Analisando os resultados podemos verificar que estes não crescem de forma linear, levando-nos a concluir que para dificuldades relativamente grandes os tempos possam chegar a dias ou até mais.

2.2 Pergunta 2.2

De facto, o algoritmo de *Proof-of-Work* é relativamente simples neste exemplo. Para conseguir criar um novo bloco, a pessoa que está a minar apenas terá de incrementar um número. Quando esse número é divisível por 9 e pelo número de prova do último bloco, então o bloco irá ser minerado e o mineiro será recompensado. De facto, este não é o algoritmo mais apropriado para minerar pois é bastante fácil de calcular, levando a que haja um grande número de minerações.