



**Universidade do Minho**

Mestrado Integrado em Engenharia Informática

## ENGENHARIA DE SEGURANÇA

### Trabalho TP4

#### **Grupo 2**

Paulo Gameiro - A72067  
Pedro Rodrigues - PG41092  
Rafaela Soares - A79034

Braga, Portugal  
18 de Março de 2020

## Conteúdo

<b>1</b>	<b>Exercício 1: <i>Blockchain</i></b>	<b>2</b>
1.1	Pergunta P1.1 . . . . .	2
1.2	Pergunta P1.2 . . . . .	2
<b>2</b>	<b>Exercício 2: <i>Proof of Work Consensus Model</i></b>	<b>2</b>
2.1	Pergunta 2.1 . . . . .	2
2.2	Pergunta 2.2 . . . . .	4

## 1 Exercício 1: *Blockchain*

### 1.1 Pergunta P1.1

O código do ficheiro "main.experiencia1.1.js" foi alterado de modo a que o primeiro bloco do método que cria o *Genesis Block* tivesse como *timestamp* a data em que o mesmo foi criado sendo, assim, o código seguinte a maneira de fazer o mesmo:

```
createGenesisBlock(){  
  var ts = new Date();  
  return new Block(0, ts.toString(), "Bloco inicial da  
    koreCoin", "0");  
}
```

Sendo que o *timestamp* fica da seguinte maneira: Sun Mar 15 2020.

### 1.2 Pergunta P1.2

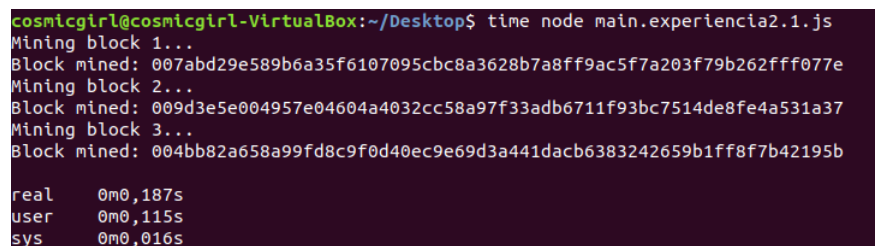
```
koreCoin.addBlock(new Block (1, "01/01/2020", {amount: 20}));  
koreCoin.addBlock(new Block (2, "02/01/2020", {amount: 40}));  
koreCoin.addBlock(new Block (3, "02/01/2020", {amount: 60}));  
koreCoin.addBlock(new Block (4, "03/01/2020", {amount: 80}));  
koreCoin.addBlock(new Block (5, "03/01/2020", {amount: 100,  
Bloco1: 40, Bloco2: 20}));
```

## 2 Exercício 2: *Proof of Work Consensus Model*

### 2.1 Pergunta 2.1

A partir do ficheiro disponibilizado, o main.experiencia2.1.js, alterou-se a dificuldade de minerar para 2, 3, 4 e 5.

Entre cada alteração, executou-se **time node main.experiencia2.1.js**. Os respetivos resultados podem ser observados nas figuras abaixo.



```
cosmicgirl@cosmicgirl-VirtualBox:~/Desktop$ 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    0m0,187s  
user    0m0,115s  
sys     0m0,016s
```

Figura 1: Dificuldade de minerar a 2

```
cosmicgirl@cosmicgirl-VirtualBox:~/Desktop$ time node main.experiencia2.1.js
Mining block 1...
Block mined: 0007c25169f5256ebb80fcb423b582a8e699a759e41abb28906295973f9de4c
Mining block 2...
Block mined: 0004af57b70136e9f4b5309f92d518b3f297168d4cf47d4f9354808daadce457
Mining block 3...
Block mined: 00075796ab77bb5289bb9a952b91047482b3348d54d915ff5e5bce28256d01fe

real    0m0,734s
user    0m0,584s
sys     0m0,012s
```

Figura 2: Dificuldade de minerar a 3

```
cosmicgirl@cosmicgirl-VirtualBox:~/Desktop$ 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    0m2,942s
user    0m2,712s
sys     0m0,028s
```

Figura 3: Dificuldade de minerar a 4

```
cosmicgirl@cosmicgirl-VirtualBox:~/Desktop$ 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    0m44,505s
user    0m44,142s
sys     0m0,327s
```

Figura 4: Dificuldade de minerar a 5

Como se pode evidenciar, quanto maior foi o nível de dificuldade, maior será o tempo necessário para resolver o *puzzle*.

## 2.2 Pergunta 2.2

Na experiência anterior, o algoritmo de '*proof of work*' é o seguinte apresentado:

```
def proof_of_work(last_proof):  
    # Create a variable that we will use to find  
    # our next proof of work  
    incrementor = last_proof + 1  
    # Keep incrementing the incrementor until  
    # it's equal to a number divisible by 9  
    # and the proof of work of the previous  
    # block in the chain  
    while not (incrementor % 9 == 0 and incrementor % last_proof == 0):  
        incrementor += 1  
    # Once that number is found,  
    # we can return it as a proof  
    # of our work  
    return incrementor
```

Figura 5: Algoritmo de '*proof of work*'

Relativamente à utilização deste algoritmo no intuito de minerar, este não é adequado, pois:

- Não é possível definir o nível de dificuldade;
- Utiliza a prova anterior para o cálculo da seguinte.