

Aula TP - 25/Mar/2019

Cada grupo deve colocar a resposta às perguntas dos seguintes exercícios na área do seu grupo no Github até ao final do dia 12/Abr/2019. Por cada dia de atraso será descontado 0,15 valores à nota desse trabalho.

Note que a máquina virtual disponibilizada pode ser utilizada para fazer estes exercícios.

Exercícios

1. Blockchain

Experiência 1.1

Neste exemplo siga o artigo [Building a blockchain](#) e os vários passos indicados no mesmo.

Notas:

1. Para instalar o node.js e o npm na máquina virtual disponibilizada execute os seguintes comandos na máquina virtual:
 - `sudo su`
 - `curl -sL https://deb.nodesource.com/setup_11.x | bash -`
 - `apt-get install -y nodejs`
 - `exit`
2. O código em javascript encontra-se na diretoria [Aula8/koreCoin](#), no ficheiro main.experiencia1.1.js .

Pergunta 1.1

Na experiência anterior, altere o método que cria o Genesis Block, de modo a que o timestamp seja a data do dia de hoje e o dado incluído nesse Bloco seja "Bloco inicial da koreCoin".

Pergunta 1.2

Na experiência anterior, adicione alguns blocos simulando várias transações em cada um deles.

Experiência 1.2

Neste exemplo siga o artigo [Let's Build the Tiniest Blockchain in Less Than 50 Lines of Python](#) e os vários passos indicados no mesmo.

Nota:

1. O código em python encontra-se na diretoria [Aula8/snakeCoin](#), no ficheiro snakecoin.experiencia1.2.py .

2. Proof of Work Consensus Model

Experiência 2.1

Esta experiência tem por base a Experiência 1.1.

Neste exemplo siga o artigo [Implementing a simple 'proof of work' algorithm for the Blockchain](#) e os vários passos indicados no mesmo.

Nota:

1. O código em javascript encontra-se na diretoria [Aula8/koreCoin](#), no ficheiro `main.experiencia2.1.js`.

Pergunta 2.1

Na experiência anterior, altere a dificuldade de minerar para 2 e veja qual o tempo que demora, utilizando o comando `time` do Linux (ou similar no seu sistema operativo), por exemplo `time node main.experiencia2.1.js`. Repita o exemplo para dificuldade de minerar 3, 4 e 5.

Apresente os tempos e conclua sobre os mesmos.

Experiência 2.2

Esta experiência tem por base a Experiência 1.2.

Neste exemplo siga o artigo [Let's Make the Tiniest Blockchain Bigger - Part 2: With More Lines of Python](#) e os vários passos indicados no mesmo.

Nota:

1. Para instalar o flask na máquina virtual disponibilizada execute os seguintes comandos na máquina virtual:
 - `sudo apt-get install python-pip`
 - `pip install flask`
2. O código em python encontra-se na diretoria [Aula8/snakeCoin](#), no ficheiro `snakecoin-server.experiencia2.2.py`.
3. Para arrancar com o servidor snakeCoin execute o seguinte comando na máquina virtual:
`python snakecoin-server.experiencia2.2.py &`
4. Para adicionar uma transação (pending transaction) na snakeCoin blockchain, execute o seguinte comando na máquina virtual:
`curl "localhost:5000/txion" \ -H "Content-Type: application/json" \ -d '{"from": "akjflw", "to": "fjlkadj", "amount": 3}'`
5. Para minerar a transação (pending transaction), execute o seguinte comando na máquina virtual:
`curl localhost:5000/mine`

Pergunta 2.2

1. Na experiência anterior, qual é o algoritmo de 'proof of work' ?
2. Parece-lhe um algoritmo adequado para minerar? Porquê?

Projeto de Engenharia de Segurança

Pode utilizar o resto da aula para o projeto de Engenharia de Segurança