

Atividade 27/08/2024 - Análise de Blockchain - Parte I
Código completo no final do arquivo !!!

Joao Otavio Manieri - 12021BSI263

Instruções solicitadas

Utilizando a API do blockchain.info

(https://www.blockchain.com/explorer/api/blockchain_api) desenvolva um script para:

- Pegar o último bloco da blockchain

```
# Função para obter o hash do último bloco
def get_latest_block():
    response = requests.get(base_url + 'latestblock')
    return response.json()
```

```
# Obter o hash do último bloco
latest_block = get_latest_block()
block_hash = latest_block['hash']
print(f"Hash do último bloco: {block_hash}")
```

```
PS C:\Users\jotam\OneDrive\Área de Trabalho\BTC > & C:\Users\jotam\AppData\Local\Programs\Python\Python312\python.exe "c:/Users/jotam/OneDrive/Área de Trabalho/BTC/aula.py"
```

- Encontrar a transação de Coinbase: é a primeira transação do bloco...nao tem segredo

```
# Transação de Coinbase (Primiera transação do bloco)
coinbase_tx = block['tx'][0]
print (f"Hash da transação de Coinbase: {coinbase_tx['hash']}")
```

Hash da transação de Coinbase: 2460f56ea3df58c0b9b9720a7cedad4699ef36338ddf016fa449b0751afdec45

- Listar os endereços que estão recebendo os Bitcoins minerados
+ taxas

```
# Calculando a soma das taxas de todas as transações no bloco (exceto a Coinbase)
total_fees = calculate_total_fees(block['tx'])

# Lista de endereços que recebem os bitcoins minerados + taxas (limitado a 10)
addresses = [out['addr'] for out in coinbase_tx['out'] if 'addr' in out][:10]

# Imprimir endereços (limite de 10 endereços)
print("Endereços que receberam os Bitcoins minerados + taxas (limite de 10):")
for address in addresses:
    print(address)

# Imprimir o valor total de bitcoins recebidos (incluindo taxas)
total_received = sum(out['value'] for out in coinbase_tx['out']) + total_fees
print(f"\nTotal de bitcoins recebidos (incluindo taxas): {total_received} satoshis")
```

```
Endereços que receberam os Bitcoins minerados + taxas (limitado de 10):
1K6KoYC69NnafWJ7YgtrpwJxBLiijWqwa6
1KGG9kvV5zXiQyQAMfY32sGt9eFLMmgpgX
```

- Imprimir a taxa paga por cada transação do bloco: lembrando que a primeira transação não tem taxa!

```
# Função para calcular a taxa de uma transação
def calculate_transaction_fee(tx):
    input_value = sum(input['prev_out']['value'] for input in tx['inputs'])
    output_value = sum(output['value'] for output in tx['out'])
    return input_value - output_value
```

```
# Iterar pelas transações do bloco (exceto a Coinbase) para calcular e imprimir as taxas (limitado de 5)
print("\nTaxa paga por cada transação no bloco (limitado a 5):")
for tx in block['tx'][1:6]: # Limitar a 5 transações
    fee = calculate_transaction_fee(tx)
    print(f"TxID: {tx['hash']}, taxa: {fee} satoshis")
```

```
Taxa paga por cada transação no bloco (limitado a 5):
TxID: bf7cb3e606017d8b775a1bd731d491587c48aae49f166c898ace1df6d3322c6d, taxa: 30600 satoshis
TxID: 0b052ec55ffb663c10440053acf5f7e80c3d6c55a94c0d3bed8d148fa3940b92, taxa: 30600 satoshis
TxID: 20918cbe21b938c8172bac0d1a1709227eb015729c94cee24fa295a39dc326e2, taxa: 38000 satoshis
TxID: 4ebd36f39f2171d3f72fd4cffc238c78f0488c29cf91157eacc84c9712342f89, taxa: 29052 satoshis
TxID: a8792e620660c4b6f7f8f0c1b261d41fa409ce50a71161b92e41e2ef07f2b925, taxa: 23829 satoshis
```

obs: limitei a 5 transações para não sobrecarregar o terminal

- Imprimir o número de entradas e o número de saídas de cada transação

```
# Imprimir o número de entradas e saídas de cada transação (limitado a 5)
print("\nNúmero de entradas e saídas de cada transação (limitado de 5):")
for tx in block['tx'][:5]: # Limitar a 5 transações
    num_inputs = len(tx['inputs'])
    num_outputs = len(tx['out'])
    print(f"TxID: {tx['hash']}, Inputs: {num_inputs}, Outputs: {num_outputs}")
```

```
Número de entradas e saídas de cada transação (limite de 5):
TxID: 2460f56ea3df58c0b9b9720a7cedad4699ef36338ddf016fa449b0751afdec45, Inputs: 1, Outputs: 6
TxID: bf7cb3e606017d8b775a1bd731d491587c48aae49f166c898ace1df6d3322c6d, Inputs: 1, Outputs: 2
TxID: 0b052ec55ffb663c10440053acf5f7e80c3d6c55a94c0d3bed8d148fa3940b92, Inputs: 1, Outputs: 2
TxID: 20918cbe21b938c8172bac0d1a1709227eb015729c94cee24fa295a39dc326e2, Inputs: 2, Outputs: 1
TxID: 4ebd36f39f2171d3f72fd4cffc238c78f0488c29cf91157eacc84c9712342f89, Inputs: 1, Outputs: 2
PS C:\Users\jotam\OneDrive\Área de Trabalho\BTC>
```

''' Python

import requests

Definindo a URL base da API do blockchain.info

base_url = 'https://blockchain.info/'

Função para obter o hash do último bloco

def get_latest_block():

response = requests.get(base_url + 'latestblock')

return response.json()

Função para obter os detalhes de um bloco pelo hash

def get_block_by_hash(block_hash):

response = requests.get(base_url + f'rawblock/{block_hash}')

return response.json()

Função para calcular a taxa de uma transação

def calculate_transaction_fee(tx):

input_value = sum(input['prev_out']['value'] for input in tx['inputs'])

output_value = sum(output['value'] for output in tx['out'])

return input_value - output_value

Função para calcular a soma das taxas de todas as transações (exceto a Coinbase)

def calculate_total_fees(transactions):

total_fees = 0

for tx in transactions[1:]: # Excluir a transação de Coinbase

total_fees += calculate_transaction_fee(tx)

return total_fees

Obter o hash do último bloco

latest_block = get_latest_block()

block_hash = latest_block['hash']

```

# Obter detalhes do bloco
block = get_block_by_hash(block_hash)

# Transação de Coinbase é sempre a primeira
coinbase_tx = block['tx'][0]

# Calculando a soma das taxas de todas as transações no bloco (exceto a Coinbase)
total_fees = calculate_total_fees(block['tx'])

# Lista de endereços que recebem os bitcoins minerados + taxas (limitado a 10)
addresses = [out['addr'] for out in coinbase_tx['out'] if 'addr' in out][:10]

# Imprimir endereços (limite de 10 endereços)
print("Endereços que receberam os Bitcoins minerados + taxas (limite de 10):")
for address in addresses:
    print(address)

# Imprimir o valor total de bitcoins recebidos (incluindo taxas)
total_received = sum(out['value'] for out in coinbase_tx['out']) + total_fees
print(f"\nTotal de bitcoins recebidos (incluindo taxas): {total_received} satoshis")

# Iterar pelas transações do bloco (exceto a Coinbase) para calcular e imprimir as taxas
(limite de 5)
print("\nTaxa paga por cada transação no bloco (limite de 5):")
for tx in block['tx'][1:6]: # Limitar a 5 transações
    fee = calculate_transaction_fee(tx)
    print(f"TxID: {tx['hash']}, Fee: {fee} satoshis")

# Imprimir o número de entradas e saídas de cada transação (limite de 5)
print("\nNúmero de entradas e saídas de cada transação (limite de 5):")
for tx in block['tx'][:5]: # Limitar a 5 transações
    num_inputs = len(tx['inputs'])
    num_outputs = len(tx['out'])
    print(f"TxID: {tx['hash']}, Inputs: {num_inputs}, Outputs: {num_outputs}")
'''

```