

APUSCOIN: Criando uma criptomoeda do zero com Python e Linux

#crypto #python #linux #blockchain #apus



Histórico Tchelinux

- **2018 [bioinformática]:**
 - Software Livre e Bioinformática: do DNA ao medicamento
- **2019 [web scraping]:**
 - Web Scraping: a arte de automatizar a recuperação de informação na Web
- **2020 [deep learning]:**
 - IA versus COVID-19
- **2021 [high performance]:**
 - Alta performance em IA: uma abordagem prática em Linux



O que é criptografia?

O que é criptografia?

Definição

A criptografia é o estudo e a prática de enviar mensagens seguras e encriptadas entre duas ou mais partes. A criptografia permite a realização de transações de moedas digitais sob pseudônimo, seguras e "trustless", ou seja, sem um banco ou outro intermediário.



O que é criptografia?



Definição

A criptografia é o estudo e a prática de enviar mensagens seguras e encriptadas entre duas ou mais partes. A criptografia permite a realização de transações de moedas digitais sob pseudónimo, seguras e "trustless", ou seja, sem um banco ou outro intermediário.

Porque é que a criptografia é importante?

As criptomoedas baseiam-se totalmente em ideias criptográficas. O Bitcoin foi inventado por uma pessoa (ou grupo de pessoas) sob o pseudónimo Satoshi Nakamoto, que propôs a ideia sob a forma de um livro branco publicado num fórum de discussão de criptografia em 2009.

O que é



BLOCKCHAIR Search for transactions, addresses, blocks and embedded text data...

Bitcoin · Transactions **Bitcoin transaction** API 200% on Deposit \$

A criptografia é baseada em hashes.

Transaction hash
1810e267c0f7ac947672a70992b1a7266bb12a04eed616d40f33f5e60f1c1ffc

Transaction status
Confirmed · 2 of 6 confirmations
Block id 733,184

Amount transacted ?
0.00353392 BTC · 140.59 USD

Transaction fee ?
0.0000741 BTC · 2.95 USD

Fee per vbyte
1 satoshi

Additional info Transaction receipt

Senders 50

1JMNxue9Z497qgxNbGC3md3mFQuQZ
Unsx
0.00001698 BTC · 0.68 USD

Recipients

bc1qppekrijpe0rt
p4apdjs0mu5n
0.00353392 BTC · 140.59 USD

BLOCKCHAIR – Um dos primeiros exploradores de Blockchain.

O q

English | [Español](#) | [Français](#) | [ελληνικά](#) | [italiano](#) | [Deutsch](#)
[Česky](#) | [Magyar](#) | [日本語](#) | [简体中文](#) | [Русский](#) | [português](#)



bitaddress.org

Geração de hashes
online



Open Source JavaScript Client-Side Bitcoin Wallet Generator



6%	6%	6%	Brain Wallet
6%	6%	Wallet Details	
<p>Generating Bitcoin Address... MOVE your mouse around to add some extra randomness... 6% OR type some random characters into this textbox <input type="text"/></p> <p>67431fae2cc1ad609156457b42bebb25214218381d4ecec118466b4393c713832 456f86eba555d51a2bcfb4aa9260bfcdf657600665c5ac306c2e2f8b72f7ce914 f7c02158bd4d3eb1afe9c2ac06a960c367d1f999ac3fe0c9d6802aade5a932322 99d860d1ce4dfd2ba6c590c2162b5234455e67e7f251261ee5b53e9d627e517e1 b1e719b3f1ce423ba972e2776384d836438b83291b8b87e4df8414f6e51fc0a17 6dce5b033e07a3bdaf0fc1a8c7b0f08ab52dd2c9c56bba013d0a94d6ddac413a8 939151f539b15d428d23733cd3408c0857c9927838aace6ec3600b19098d41fda 9ad6ca2c0b936477ff26081bbd84181ea5b4609f83cd4702a2bb9dfc5</p>			

BITADDRESS – Gerador de hashes para carteiras Bitcoin.

O qu



 **bitaddress.org**
Open Source JavaScript Client-Side Bitcoin Wallet Generator

Single Wallet Paper Wallet Bulk Wallet Brain Wallet
Vanity Wallet Split Wallet Wallet Details

Generate New Address Print

Bitcoin Address

Chave Pública



SHARE

14f1aMNF9Pgav8yRyS8xSpwG3zXmuY6o6b

Chave Privada

SECRET



Private Key

Kyye4LJLWXjom1R37R6oBVpuEFSCqzeLsiJz5mkEVpDc4SxNUyFL

ENTENDENDO AS CHAVES DO BITCOIN – Chave pública versus Chave privada.

O que é chave privada?

Onde deve guardar as suas chaves privadas?



À semelhança de qualquer palavra-passe, é crucial manter as suas chaves privadas em segurança. As duas principais formas de as manter sob controlo são:

O que é chave privada?

Onde deve guardar as suas chaves privadas?



À semelhança de qualquer palavra-passe, é crucial manter as suas chaves privadas em segurança. As duas principais formas de as manter sob controlo são:

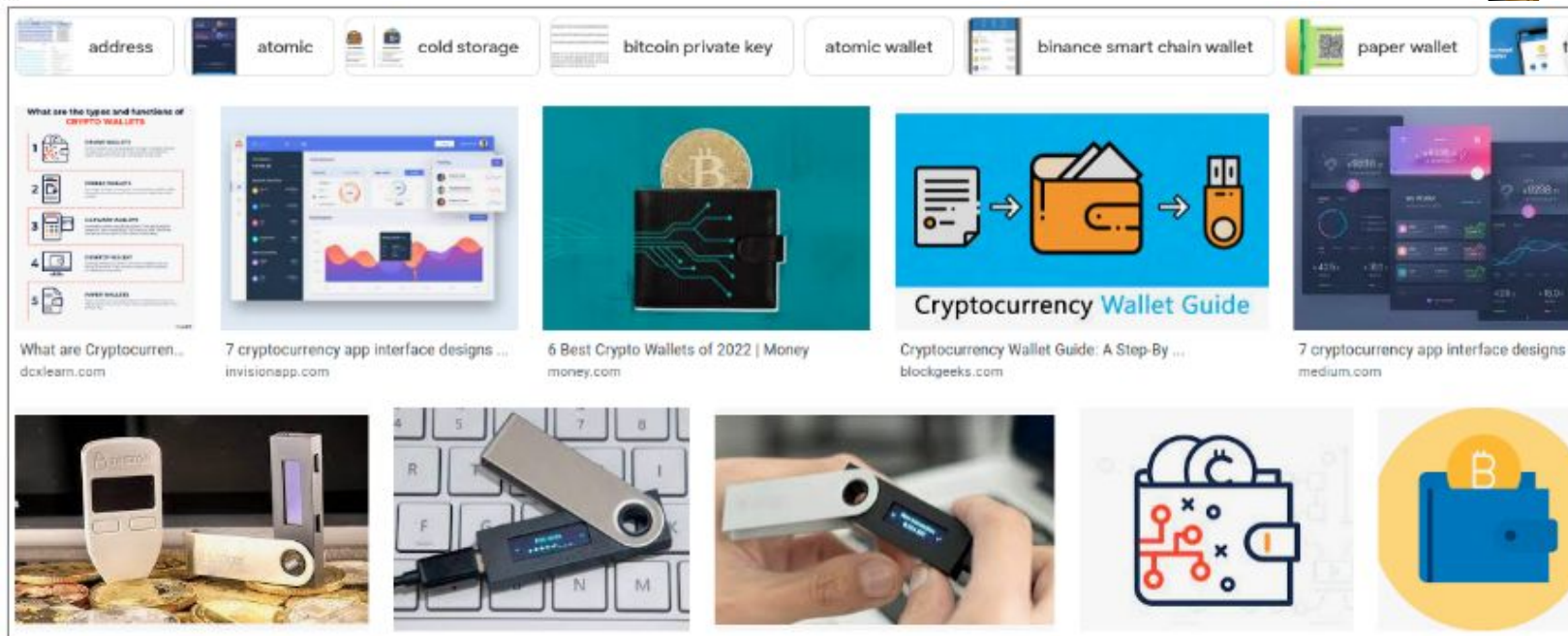
- **Armazená-las online numa carteira de criptomoedas:** a opção melhor e mais simples para a maior parte das pessoas consiste em usar uma carteira virtual, como a oferecida pela Coinbase, para gerir as suas chaves privadas. Estas são conhecidas como carteiras "quentes".

← **Hot wallet**

- **Armazená-las offline num local seguro:** alguns investidores optam por manter as respetivas chaves privadas num computador que não esteja ligado à internet, escritas em papéis ou até memorizadas.

← **Cold wallet**

O que é chave privada?



CARTEIRAS DO BITCOIN – Quentes “hot” ou Frias “cold”.



O que é criptomoeda?

O que é criptomoeda?



O que é mineração de criptomoedas?

A maioria das criptomoedas é "minada" através de uma rede descentralizada (também conhecida como peer-to-peer) de computadores. Mas a mineração não gera apenas mais Bitcoin ou Ethereum - é também o mecanismo que atualiza e protege a rede verificando constantemente o livro razão do blockchain público e adicionando novas transações.



bitmain



All

Images

News

Shopping

Videos

More

Tools

Collections SafeSearch



antminer



miner



mining



farm



bitcoin miner



logo



antpool



ethereum



asic



c



BITMAIN
bitmain.com



Bitmain Antminer S9 13.5 TH ...
shopee.com.br



Bitmain lança minerador de Bitcoin que ...
cointelegraph.com.br



BITMAIN
bitmain.com



Bitcoin Bitmain Antminer Differnet L...
pt.dhgate.com



BITMAIN – Um dos maiores pools de mineração do mundo.

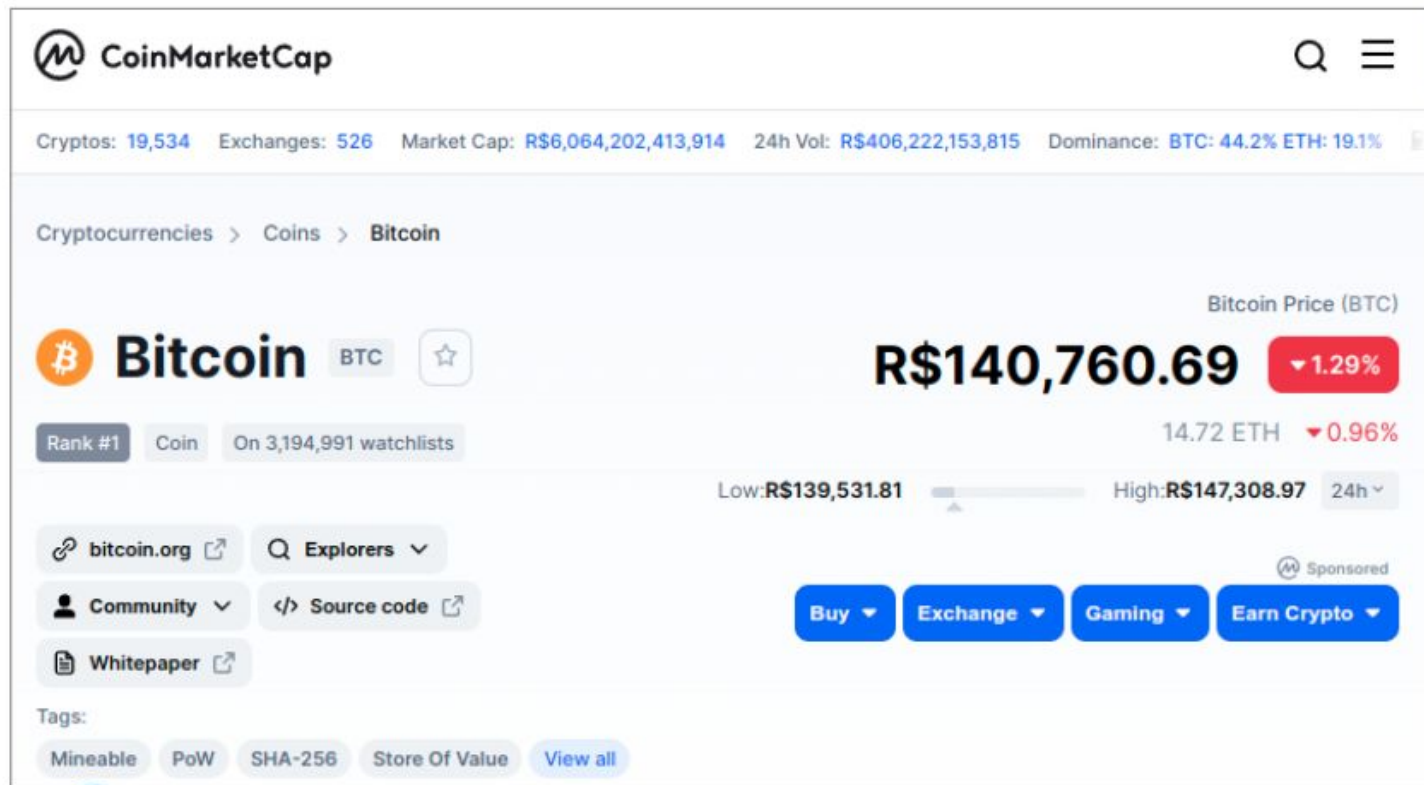
O que é criptomoeda?



#	Nome	Preço	Variação	Gráfico	Negociar
1	 Bitcoin BTC	R\$ 140.399,84	-3,46%		comprar
2	 Ethereum ETH	R\$ 9.529,02	-2,75%		comprar
3	 Cardano ADA	R\$ 2,48	-4,93%		comprar
4	 Solana SOL	R\$ 238,14	-5,44%		comprar

COINBASE – Visualizando as cotações das criptomoedas.

O q



The screenshot displays the CoinMarketCap interface for Bitcoin. At the top, the site's logo and navigation links are visible. Below the header, a summary bar shows market statistics: 19,534 cryptocurrencies, 526 exchanges, a total market cap of R\$6,064,202,413,914, and a 24-hour volume of R\$406,222,153,815. The dominance of Bitcoin is listed as 44.2%, with Ethereum at 19.1%.

The main section is titled 'Cryptocurrencies > Coins > Bitcoin'. It features the Bitcoin logo, the name 'Bitcoin', and the ticker 'BTC'. The current price is R\$140,760.69, with a 1.29% decrease. A secondary price in ETH is shown as 14.72 ETH, also down 0.96%. The price range for the last 24 hours is from R\$139,531.81 to R\$147,308.97.

On the left, there are links to 'bitcoin.org', 'Community', 'Whitepaper', and 'Source code'. Below these are tags for 'Mineable', 'PoW', 'SHA-256', and 'Store Of Value'. On the right, there are buttons for 'Buy', 'Exchange', 'Gaming', and 'Earn Crypto', with a 'Sponsored' label above them.



COINMARKETCAP – Visualizando os detalhes de um projeto cripto.



Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1. Introduction

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for

WHITEPAPER – Uma descrição completa de uma criptomoeda.

O que



ICO – Oferta inicial de ativos virtuais ou oferta inicial de moedas.



O que é um protocolo?

O que é um protocolo?



Definição

Os protocolos são conjuntos de regras básicas que permitem a partilha de dados entre computadores. Para as criptomoedas, estabelecem a estrutura do blockchain: a base de dados distribuída que permite que o dinheiro digital seja trocado de forma segura na internet.

O que é um protocolo?



Definição

Os protocolos são conjuntos de regras básicas que permitem a partilha de dados entre computadores. Para as criptomoedas, estabelecem a estrutura do blockchain: a base de dados distribuída que permite que o dinheiro digital seja trocado de forma segura na internet.

Porque é que os protocolos são importantes?

Os protocolos permitem que as criptomoedas sejam descentralizadas através do blockchain, o que significa que estão distribuídas por uma rede de computadores sem hub central ou autoridade.

O que é um protocolo?



Protocolo no código

Search or jump to... Pull requests Issues Marketplace Explore

bitcoin / bitcoin Public Watch 3.9k Fork 32.2k Star 63.5k

<> Code Issues 602 Pull requests 405 Projects 6 Security Insights

master 7 branches 266 tags Go to file Add file Code

hebasto Merge bitcoin-core/gui#587: refactor: Replace `GUIUtil::ObjectInvo...` ✓ be7a5f2 22 hours ago 33,532 commits

.github	doc: Remove label from good first issue template	2 years ago
.tx	qt: Update transfex resource blob to 23.0	3 months ago
build-aux/m4	build: stop overriding user CXXFLAGS	20 days ago
build_msvc	Revert "build: Specify zeromq port explicitly for MSVC builds"	10 days ago
ci	ci: Make log verbose in error case only	9 days ago

About

Bitcoin Core integration/staging tree

bitcoincore.org/en/download

c-plus-plus cryptography bitcoin p2p

cryptocurrency

Readme MIT License 63.5k stars 3.9k watching

BITCOIN CORE – Projeto disponível no GitHub.



O que é blockchain?

O que é blockchain?



Na sua forma mais básica, um blockchain é uma lista de transações que qualquer pessoa pode ver e verificar. O blockchain Bitcoin, por exemplo, contém um registo de cada vez que alguém enviou ou recebeu bitcoins.

O que é blockchain?



Na sua forma mais básica, um blockchain é uma lista de transações que qualquer pessoa pode ver e verificar. O blockchain Bitcoin, por exemplo, contém um registo de cada vez que alguém enviou ou recebeu bitcoins.

As criptomoedas e a tecnologia de blockchain que as capacita tornam possível transferir valor online sem a necessidade de um intermediário, tal como um banco ou empresa de cartões de crédito.

O que é blockchain?

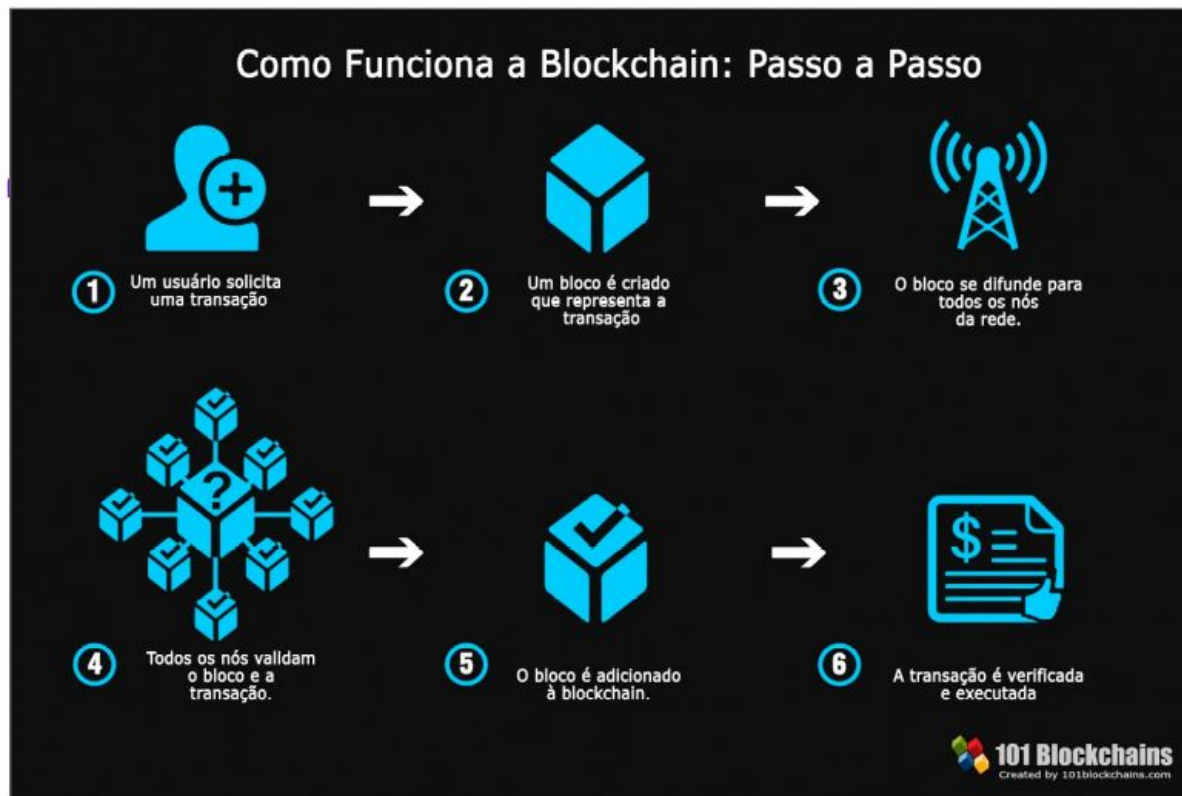


Na sua forma mais básica, um blockchain é uma lista de transações que qualquer pessoa pode ver e verificar. O blockchain Bitcoin, por exemplo, contém um registo de cada vez que alguém enviou ou recebeu bitcoins.

As criptomoedas e a tecnologia de blockchain que as capacita tornam possível transferir valor online sem a necessidade de um intermediário, tal como um banco ou empresa de cartões de crédito.

Imagine uma alternativa aberta e global a todos os serviços financeiros que utiliza atualmente, acessível com pouco mais do que um smartphone e ligação à internet.

O que é



BLOCKCHAIN – Como funciona a tecnologia.



O que é um fork?

O que é um fork?



Porque é que isso é importante?

Um fork ocorre para tornar uma criptomoeda mais segura ou adicionar outras funcionalidades.

O que é um fork?



Porque é que isso é importante?

Um fork ocorre para tornar uma criptomoeda mais segura ou adicionar outras funcionalidades.

- **Soft fork:** foram utilizados para introduzir novas funções ou funcionalidades, geralmente ao nível da programação, de Bitcoin e Ethereum. Uma vez que o resultado final é um só blockchain, as alterações são retrocompatíveis com os blocos pré-fork.
- **Hard fork:** um hard fork ocorre quando o código se altera de tal modo que a nova versão deixa de ser retrocompatível com blocos mais antigos. Neste cenário, o blockchain divide-se em dois: o blockchain original e uma nova versão que segue um novo conjunto de regras.





Criando a nossa própria moeda Códigos



Criando a nossa própria moeda











 Search or jump to... / [Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)

 [apusdigital](#) / [tchelinix-2022](#) Public Edit Pins Watch 0 Fork 0 Star 0

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)


main 1 branch 0 tags Go to file Add file Code


 **alexcamargoweb** initial commit 449a660 3 days ago 2 commits


 README.md	initial commit	3 days ago
 node_01.py	initial commit	3 days ago
 node_02.py	initial commit	3 days ago
 node_03.py	initial commit	3 days ago
 nodes.json	initial commit	3 days ago
 tchelinix-2022.pdf	initial commit	3 days ago
 transactions.json	initial commit	3 days ago


About

Repositório da palestra "APUSCOIN: Criando uma criptomoeda do zero com Python e Linux". Tchelinix Live 2022.

 Readme

 0 stars

 0 watching

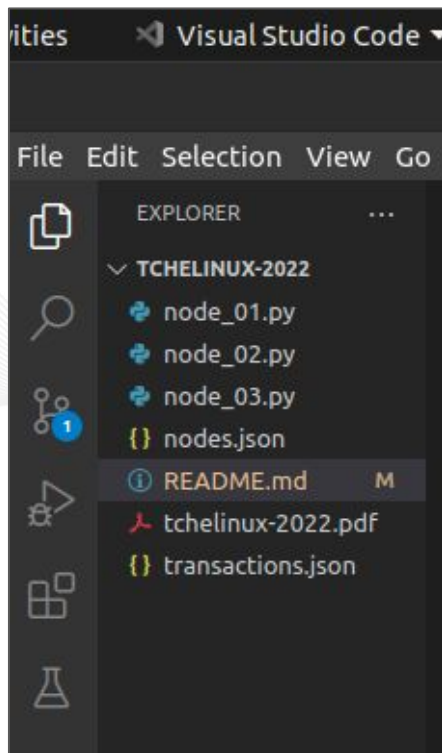
 0 forks

Releases

No releases published

[Create a new release](#)

Criando a nossa própria moeda



Criando a nossa própria moeda



```
① README.md M X  node_01.py  node_02.py  node_03.py  {} nodes.json  {} transactions.json
① README.md > # tchelinix-2022 > ## Sequência de testes via POSTMAN:
1  # tchelinix-2022
2  Repositório da palestra "APUSCOIN: Criando uma criptomoeda do zero com Python e Linux". Tchelinix Live 2022.
3
4  ## Sequência de testes via POSTMAN:
5
6  1. POST /connect_node - Conecta o nó atual aos nós na rede
7  Body >> raw >> JSON >> nodes.json
8  http://127.0.0.1:5001/connect_node
9  http://127.0.0.1:5002/connect_node
10 http://127.0.0.1:5003/connect_node
11
12 2. GET /get_chain - Retorna o blockchain atual
13 http://127.0.0.1:5001/get_chain
14 http://127.0.0.1:5002/get_chain
15 http://127.0.0.1:5003/get_chain
16
17 3. POST /add_transaction - Envia uma transação através via Node 01
18 Body >> raw >> JSON >> transactions.json
19 http://127.0.0.1:5001/add_transaction
20
21 4. GET /mine_block - Minera um bloco
22 http://127.0.0.1:5001/mine_block
23
24 5. GET /replace_chain - Aplica o protocolo de consenso
25 http://127.0.0.1:5002/replace_chain
26 http://127.0.0.1:5003/replace_chain
27
```

Criando a nossa própria moeda



```
{} README.md M node_01.py node_02.py node_03.py {} nodes.json X {} transactions.json
{} nodes.json > ...
1  {
2    "nodes": ["http://127.0.0.1:5001",
3              "http://127.0.0.1:5002",
4              "http://127.0.0.1:5003"]
5  }
```

Tche

Criando a nossa própria moeda



```
{} transactions.json > ...
1  {
2    "sender": "",
3    "receiver": "",
4    "amount":
5  }
```

Tche

Criando a nossa própria moeda



```
node_01.py > mine_block
1 import datetime # lida com as datas
2 import hashlib # cria as hashes
3 import json # cria os arquivos json
4 from flask import Flask, request, jsonify # envia/recebe as requisições
5 import requests # gerencia os nós da rede
6 from uuid import uuid4 # gera identificadores únicos aleatoriamente
7 from urllib.parse import urlparse # trabalha com as URLs
8
```

Criando a nossa própria moeda



```
8
9  # Parte 1 - Cria a moeda Apuscoin (transações)
10
11  class Blockchain:
12      # função de inicialização
13      def __init__(self): # "self" usa as variáveis do objeto
14          self.chain = [] # lista Python como o bloco
15          # cria uma lista para as transações: as transações são adicionadas ao bloco através da mineração
16          self.transactions = []
17          # cria o bloco gênese
18          self.create_block(proof = 1, previous_hash = '0')
19          # adiciona os nós participantes da rede através de um set (mais otimizado q o list)
20          self.nodes = set()
21
```


Criando a nossa própria moeda



```
21
22 # função que cria um bloco com suporte a transações e adiciona ao blockchain (encadear)
23 def create_block(self, proof, previous_hash): # faz o link com o bloco anterior
24     # cria o dicionário com as 5 chaves do bloco: índice, timestamp, proof (nonce), previous_hash, transações
25     # importante: o bloco é criado depois ser de minerado (proof-of-work)
26     block = {
27         'index': len(self.chain) + 1,
28         'timestamp': str(datetime.datetime.now()),
29         'proof': proof,
30         'previous_hash': previous_hash,
31         # adiciona uma chave para as transações na criação do bloco
32         'transactions': self.transactions
33     }
34     # quando o bloco é criado, zera a lista de transações
35     self.transactions = []
36     # inclui o bloco (lista) à blockchain (lista)
37     self.chain.append(block)
38
39     return block # retorna o bloco criado
40
```



Criando a nossa própria moeda

```
40
41 # função que retorna o bloco anterior
42 def get_previous_block(self): # recebe o próprio objeto (lista inicializada)
43     return self.chain[-1] # retorna o bloco anterior
44
45 # função que faz o proof-of-work (deve ser difícil de resolver e fácil de recuperar)
46 def proof_of_work(self, previous_proof): # recebe: o próprio objeto e a prova anterior
47     new_proof = 1 # inicializa o proof (prova)
48     check_proof = False # verifica se o proof está correto
49     # resolve o hash (quanto mais zeros à esquerda, mais difícil o problema)
50     while check_proof is False:
51         # cria uma hash em hexadecimal
52         hash_operation = hashlib.sha256(
53             str(new_proof**2 - previous_proof**2).encode()).hexdigest()
54         # verifica se resolveu o problema/puzzle
55         if hash_operation[:4] == '0000':
56             check_proof = True
57         else:
58             new_proof += 1 # incrementa o proof antes de testar novamente
59
60     return new_proof # retorna o proof
61
```

Criando a nossa própria moeda



```
66 # cria uma função de hash própria
67 def hash(self, block): # recebe o próprio objeto e o próprio bloco (lista)
68     # gera o json codificado do bloco
69     # ordena pela chave e codifica como string
70     encoded_block = json.dumps(block, sort_keys=True).encode()
71     # retorna o hash SHA256 do json
72     return hashlib.sha256(encoded_block).hexdigest()
73
```

Criando a nossa própria moeda



```
100
101 # função que define o formato da transação
102 def add_transaction(self, sender, receiver, amount): # recebe a si própria, o decto, o receptor e o valor
103     # define o formato da transação no dicionário Python
104     self.transactions.append({
105         'sender': sender,
106         'receiver': receiver,
107         'amount': amount
108     })
109     # verifica em qual bloco a transação foi adicionada
110     previous_block = self.get_previous_block()
111     # retorna o bloco atual
112     return previous_block['index'] + 1
113
```



Criando a nossa própria moeda

```
197 # cria a rota que adiciona uma transação ao bloco
198 @app.route('/add_transaction', methods = ['POST'])
199 # cria a função que adiciona uma transação ao bloco
200 def add_transaction(): # cria a transação
201     json = request.get_json() # pega o arquivo json enviado
202     # verifica se o json é válido, ou seja, se as chaves existem
203     transaction_keys = ['sender', 'receiver', 'amount']
204     # se não há chaves no arquivo
205     if not all(key in json for key in transaction_keys):
206         # retorna um aviso e o status de resposta http (Bad request)
207         return 'Alguns elementos estão faltando', 400
208     else:
209         # se a transação está ok, adiciona as transações ao próximo bloco
210         index = blockchain.add_transaction(
211             json['sender'], json['receiver'], json['amount'])
212         # cria a mensagem de output
213         response = {
214             'message': f'Esta transação será adicionada ao Bloco {index}.'}
215         # retorna a mensagem no formato json e o status de resposta http (Created)
216         return jsonify(response), 201
```


Criando a nossa própria moeda



```
159 # cria a rota de mineração
160 @app.route('/mine_block', methods = ['GET'])
161 # função que vai minerar o bloco a partir da instância criada
162 def mine_block():
163     previous_block = blockchain.get_previous_block() # pega o bloco anterior
164     previous_proof = previous_block['proof'] # pega o proof-of-work do bloco
165     # adiciona uma transação de recompensa
166     blockchain.add_transaction(sender = node_address, receiver = 'miner', amount = 0.5)
167     # pega o proof-of-work do bloco anterior
168     proof = blockchain.proof_of_work(previous_proof)
169     # pega o hash anterior
170     previous_hash = blockchain.hash(previous_block)
171     # cria o bloco
172     block = blockchain.create_block(proof, previous_hash)
173     # exibe o resultado da mineração na página
174     response = {
175         'message': 'Parabéns! Você minerou um bloco.',
176         'index': block['index'],
177         'timestamp': block['timestamp'],
178         'proof': block['proof'],
179         'previous_hash': block['previous_hash'],
180         # inclui as transações do bloco
181         'transaction': block['transactions']
182     }
183     # retorna no formato json e o status de resposta http (OK)
184     return jsonify(response), 200
```

Criando a nossa própria moeda



```
120
121 # função do protocolo de consenso (substitui o blockchain se encontrar um bloco maior)
122 def replace_chain(self):
123     network = self.nodes # recebe todos os nós (conjunto)
124     longest_chain = None # encontra a cadeia mais longa
125     max_length = len(self.chain) # verifica o tamanho da cadeia
126     # percorre todos os nós da rede (busca o blockchain mais longo)
127     for node in network:
128         # pega o comprimento da rede
129         response = requests.get(f'http://{node}/get_chain')
130         # verifica se a resposta está correta
131         if response.status_code == 200:
132             # pega o comprimento do blockchain
133             length = response.json()['length']
134             # pega o blockchain em si
135             chain = response.json()['chain']
136             # verifica se o comprimento atual é maior que o tamanho máximo
137             if length > max_length and self.is_chain_valid(chain):
138                 # atualiza o maior valor com o blockchain atual
139                 max_length = length
140                 longest_chain = chain
141     # verifica se foi encontrado um bloco maior
142     if longest_chain:
143         self.chain = longest_chain
144         return True # o blockchain foi substituído
145     else:
146         return False # não houve substituição no blockchain
147
```

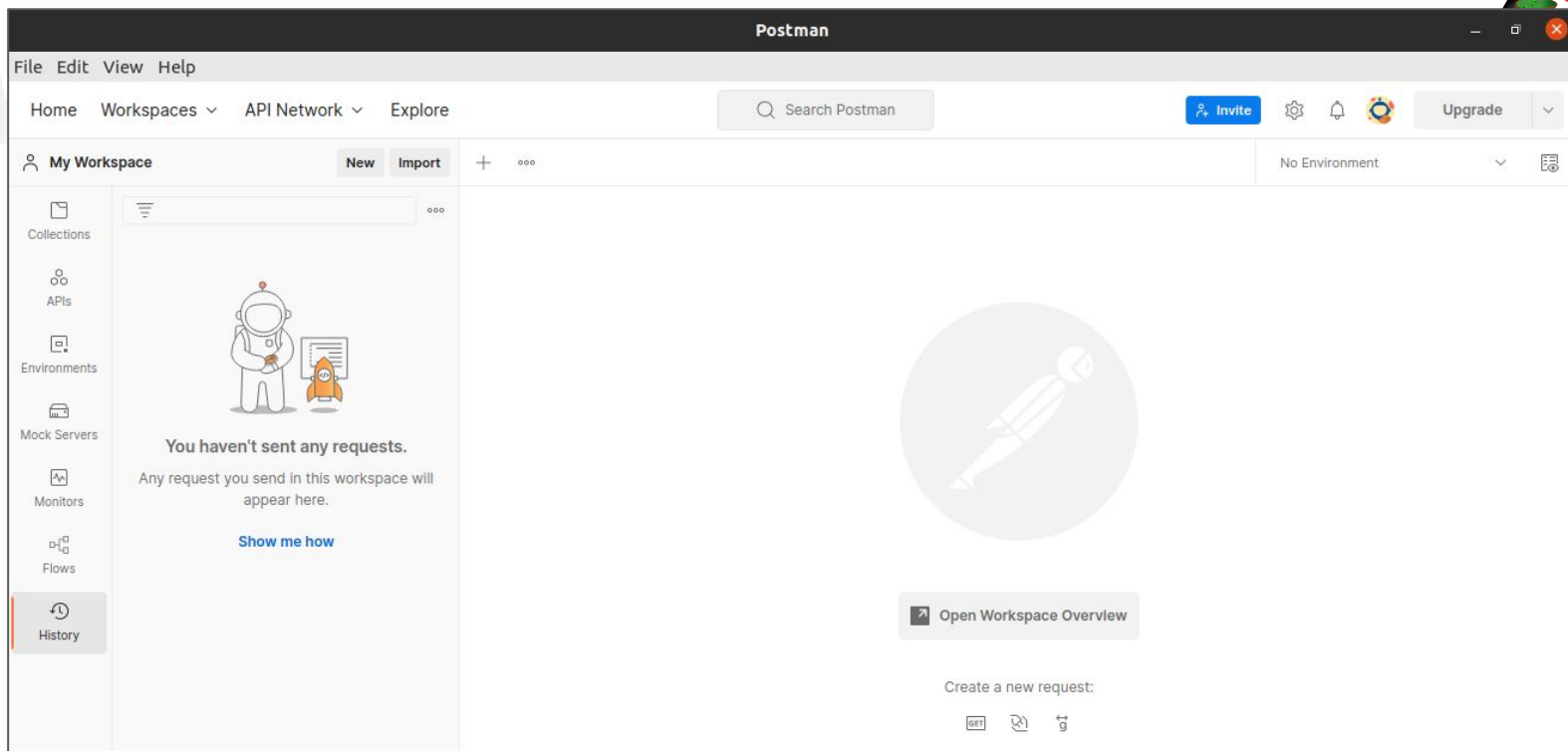


Criando a nossa própria moeda

Parte prática



Criando a nossa própria moeda



Principais referências

<https://www.coinbase.com/pt/learn/crypto-basics/what-is-cryptography>

<https://www.coinbase.com/pt/learn/crypto-basics/what-is-a-private-key>

<https://www.coinbase.com/pt/learn/crypto-basics/what-is-cryptocurrency>

<https://www.coinbase.com/pt/learn/crypto-basics/what-is-a-blockchain>

<https://www.coinbase.com/pt/learn/crypto-basics/what-is-a-fork>

<https://www.coinbase.com/pt/learn/crypto-basics/what-is-a-bitcoin-halving>

<https://www.coinbase.com/pt/learn/crypto-basics/what-is-a-protocol>

<https://polkadot.network/>

<https://coinmarketcap.com/currencies/polkadot-new/>

<https://holdbtc100.com/>

<https://www.pagcripto.com.br/>

<https://coinmarketcap.com/currencies/bitcoin/>

<https://mempool.space/pt/>