

UniBlock Documentation

Release 1.0.0

UniBlock

May 15, 2019

Table of Contents

1	UniBlock Main	3
2	UniBlock Comunnication	3
3	UniBlock BlockChain	4

1 UniBlock Main

main.main ()

Função principal do programa.

main.parseArguments ()

Função que identifica os argumentos passados.

Returns parser – objetos contendo os argumentos.

2 UniBlock Comunnication

class communication.Connection (myIp, listClients)

Classe base para conexoes

getMinersAndTraders ()

Adiciona nas respectivas listas os ips que são mineradores e ips que são traders.

listenConnection (port=5055)

Coloca o servidor para rodar de fato. Após, fica escutando a porta e quando chegar alguma conexão, cria um thread para o cliente. Trata envia para a função que irá tratar a requisição.

Parameters

- **Ip** – Endereço Ip que o servidor irá rodar
- **Port** – Porta em que o servidor irá rodar

myIp

Metodo getter do myIp.

Returns str – ip.

printClients ()

Imprime na tela a lista de clientes mineradores e a lista de clientes negociadores.

class communication.Miner (myIp, listClients, rich)

Classe do minerador.

filterCommunication (conn, addr)

Trata as conexões dos clients... Recebe uma mensagem, filtra e envia uma resposta de acordo ou executa ações.

TypeOfClient => retorna se o cliente é um minerador ou um negociador. Rich => retorna se o cliente é um minerador que está armazenando transações na carteira. NewTransaction => recebe uma nova transação a ser adicionada na carteira. NewBlock => recebe a noticia que a chain foi atualizada, entao o cliente deve validar a sua chain.

Parameters

- **conn** – Socket de conexão com o cliente
- **addr** – Endereço da conexão deste cliente

sendBlock (block)

Envia o ultimo bloco minerado para todos na rede.

Parameters **block** – ultimo bloco minerado.

sendTransactionsToMiners ()

Envia as transacoes para os mineradores.

class communication.Trader (myIp, listClients)

Classe do usuario comum.

discoverMiner ()

Método que descobre qual o minerador que está aceitando transações para adicionar na carteira.

filterCommunication (conn, addr)

Trata as conexões dos clients... Recebe uma mensagem, filtra e envia uma resposta.

Parameters

- **conn** – Socket de conexão com o cliente.
- **addr** – Endereço da conexão deste cliente.

runMethods ()

Metodo em loop que fica pedindo ao usuário o texto que deve ser adicionado na chain. Esta função acaba criando uma cadeia de chamadas de funções.

sendToMiner (transaction)

Envia a transação para o minerador. Utiliza pickle para serializar o dado e enviar.

Parameters transaction – Transaction.

userInput ()

Inicia a transação. Descobre o ip do minerador que está com a flag rich. Envia esta transação para o minerador com a flag rich.

3 UniBlock Blockchain

class Blockchain.BlockChain

Classe pai da blockchain. Implementa os metodos essenciais para a blockchain.

chain

Método getter para a cadeia de blocos.

Returns list – cadeia de blocos.

static hash (block)

Método estatico que gera a hash do bloco.

Parameters block – bloco da cadeia de blocos.

Returns str – hash do bloco.

last_block

Método getter para o ultimo bloco da chain.

Returns list – ultimo bloco da cadeia de blocos.

last_proof

Método getter para a ultima prova de trabalho adicionada na chain.

Returns int – ultima prova de trabalho.

rule

Metodo getter para a regra da prova de trabalho.

Returns int – regra da prova de trabalho.

valid_chain (chain)

Confere se a chain eh valida atraves das hashes da chain.

Parameters chain – cadeia de blocos.

Returns bool – flag da validade do ultimo bloco da cadeia de blocos.

static valid_proof (last_proof, proof, rule)

Metodo estatico que valida a proof gerada.

Parameters • **last_proof** – ultima prova.

 • **proof** – prova atual.

 • **rule** – regra da prova de trabalho.

Returns bool – flag da prova de trabalho gerada.

class Blockchain.MinerChain

Classe que estende a classe Blockchain Classe que implementa os metodos da chain utilizada pelos mineradores da blockchain

block

Metodo getter do block.

Returns list – bloco da cadeia de blocos.

current_transactions

Metodo getter para as transacoes atuais.

Returns dict – transacao atual.

finish_transactions

Metodo getter para as transacoes fechadas.

Returns dict – transacao.

mine ()

Minera a carteira se ja estiver pronto para minerar. Muda a flag para false e retorna o block minerado.

new_block (proof, previous_hash=None)

Cria um novo bloco com as informacoes.

Parameters • **proof** – prova de trabalho.

 • **previous_hash** – hash do bloco anterior.

Returns dict – novo bloco.

new_transaction (transaction)

Metodo que recebe uma nova transacao e adiciona nas transacoes atuais. Se o numero maximo de transacoes da carteira for atingido. Uma nova carteira e adicionada na lista de transacoes.

Parameters transaction – nova transacao.

Returns int – indice do proximo bloco.

proof_of_work (*last_proof*)

Metodo de prova de trabalho. Determina a dificuldade de minerar um block.

Parameters **last_proof** – ultima prova de trabalho.

Returns int – prova de trabalho.

start_miner

Metodo getter para retornar o valor da flag _start_miner. Responsavel por dizer (return True) quando uma carteira esta pronta para ser minerada.

Returns bool – flag de inicio da mineracao.

transactions

Metodo getter para as transacoes.

Returns list – lista de transacoes.

class Blockchain.**TraderChain**

Classe que implementa a chain dos traders

new_transaction (*myIp*)

Cria uma nova transacao que sera enviada para a carteira ativa

Parameters **myIp** – ip da maquina.

Returns dict – lista de transacoes.