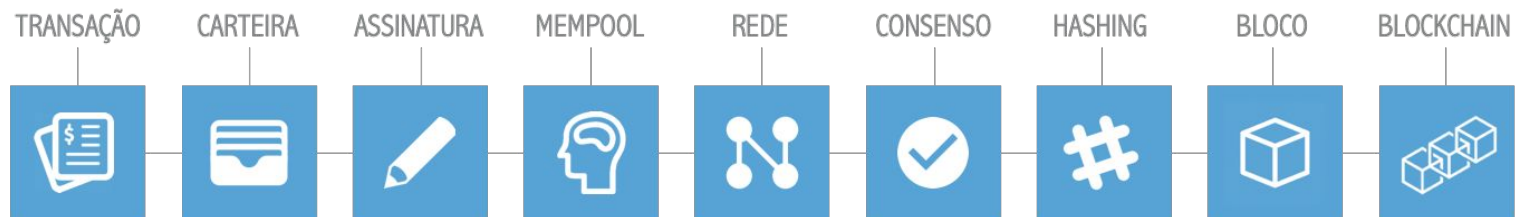


IMD0913

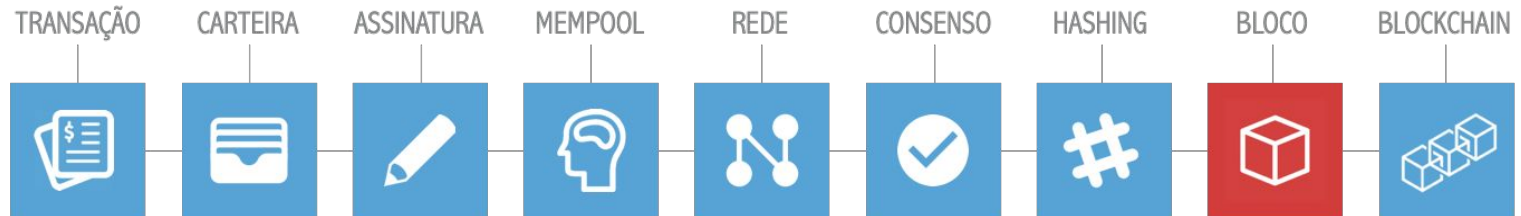
ARQUITETURA DE UM BLOCKCHAIN

BLOCO

ARQUITETURA DE UM **BLOCKCHAIN**



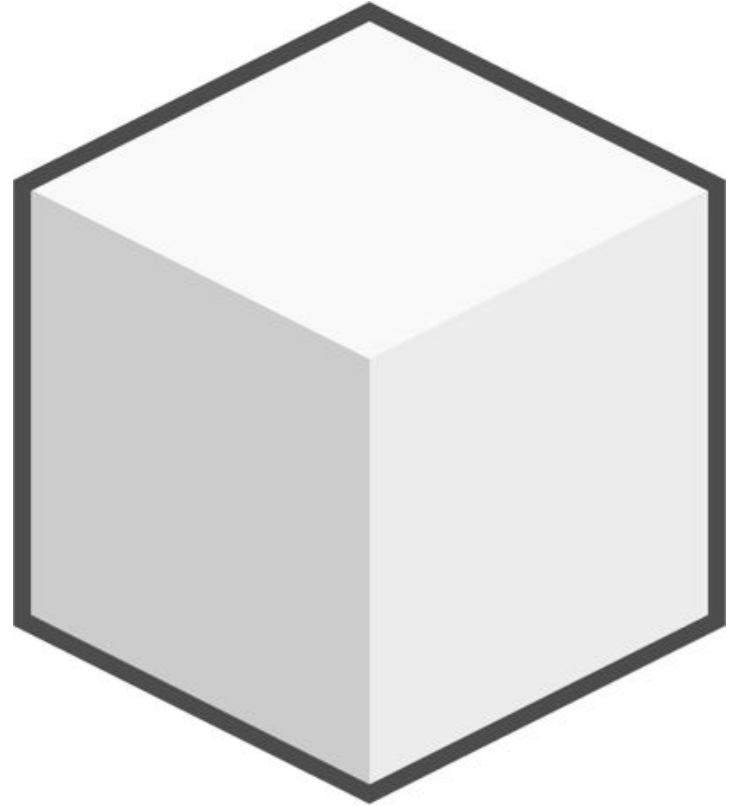
ARQUITETURA DE UM **BLOCKCHAIN**



Bloco

Componente elementar do *blockchain*

Segmentação do *blockchain* em unidades
mais elementares



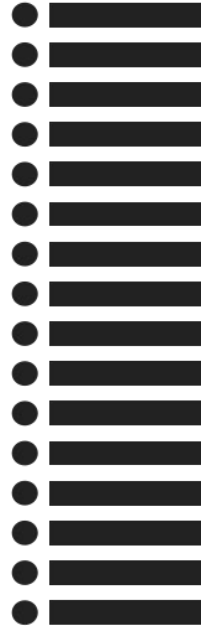
Bloco

Um *container* que armazena uma lista de transações para serem adicionadas ao *blockchain*.

Blockchain

Um livro-razão digital e compartilhado que registra uma lista de transações no formato de uma sequência de blocos.

transações



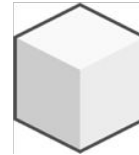
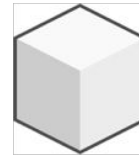
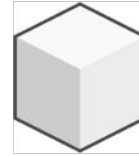
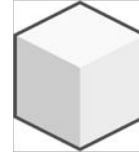
transações



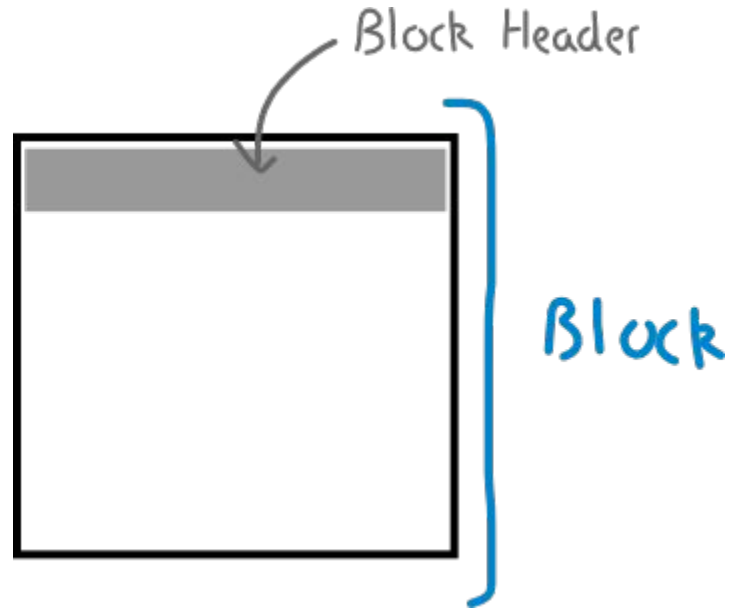
transações



blocos



Cabeçalho (*header*) de um bloco



Cabeçalho (*header*) de um bloco

O **número de versão** do bloco

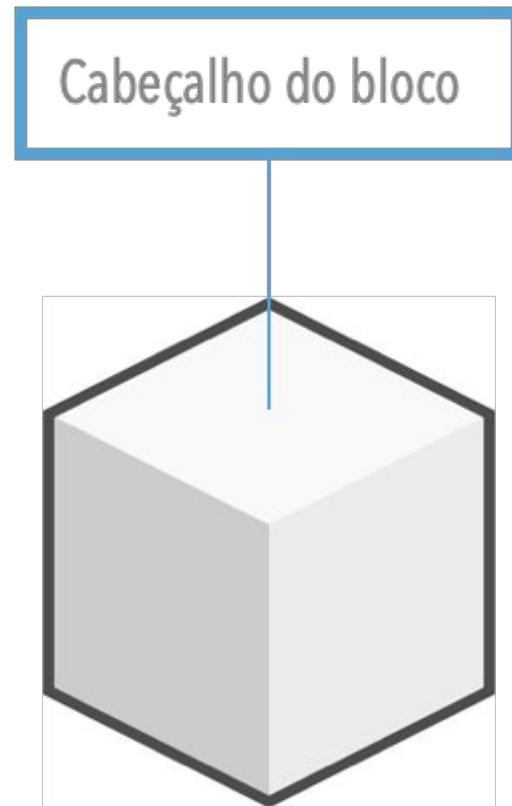
O **hash do bloco anterior** (*prevBlockHash*) na cadeia

Um código gerado pelos dados transacionais (**merkle root**)

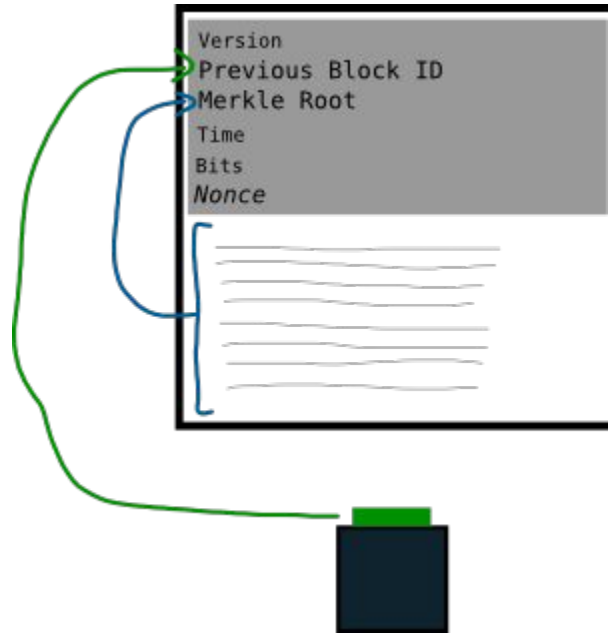
Um **timestamp** de quando o bloco foi criado

O alvo de **dificuldade** do bloco (*bits*)

Um valor aleatório chamado **nonce**



Cabeçalho (*header*) de um bloco

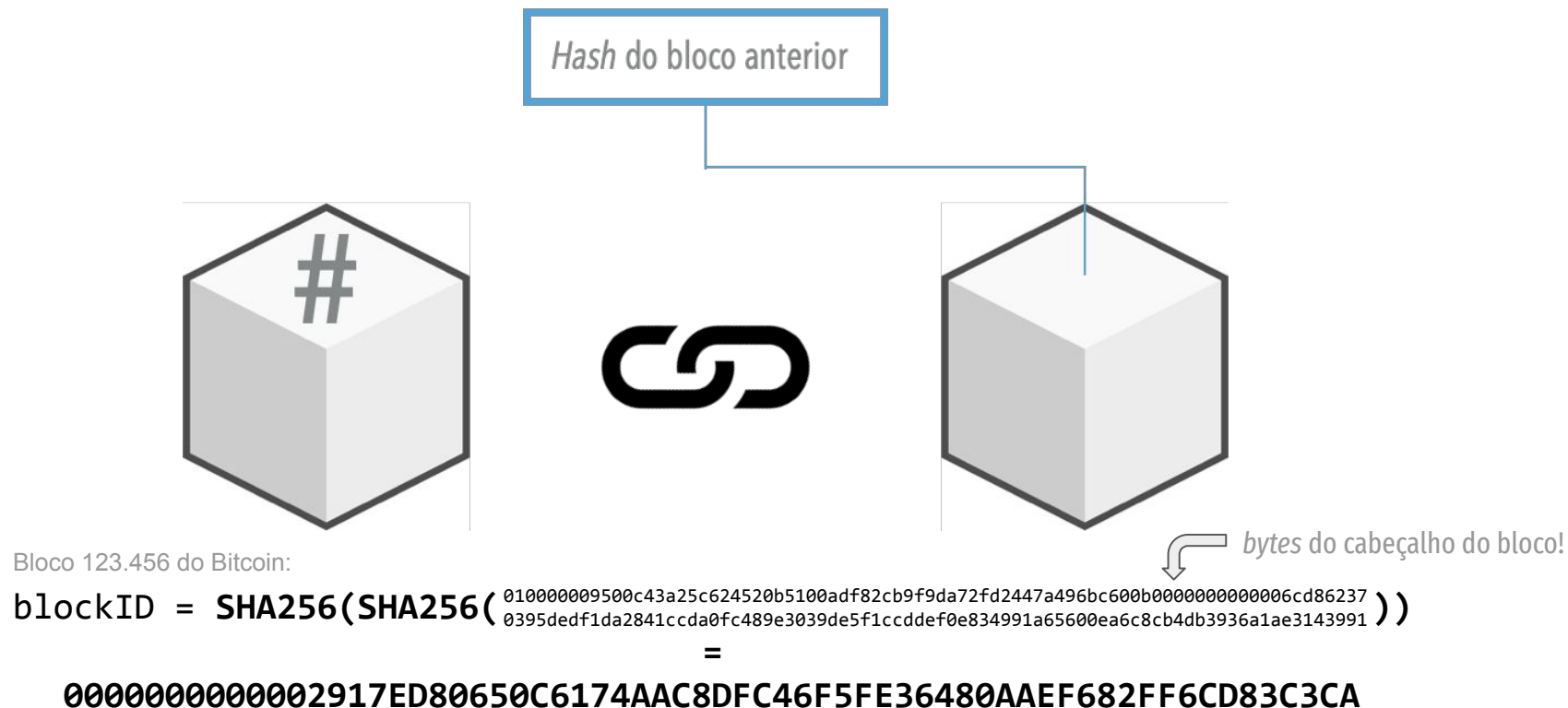


Cabeçalho (*header*) de um bloco

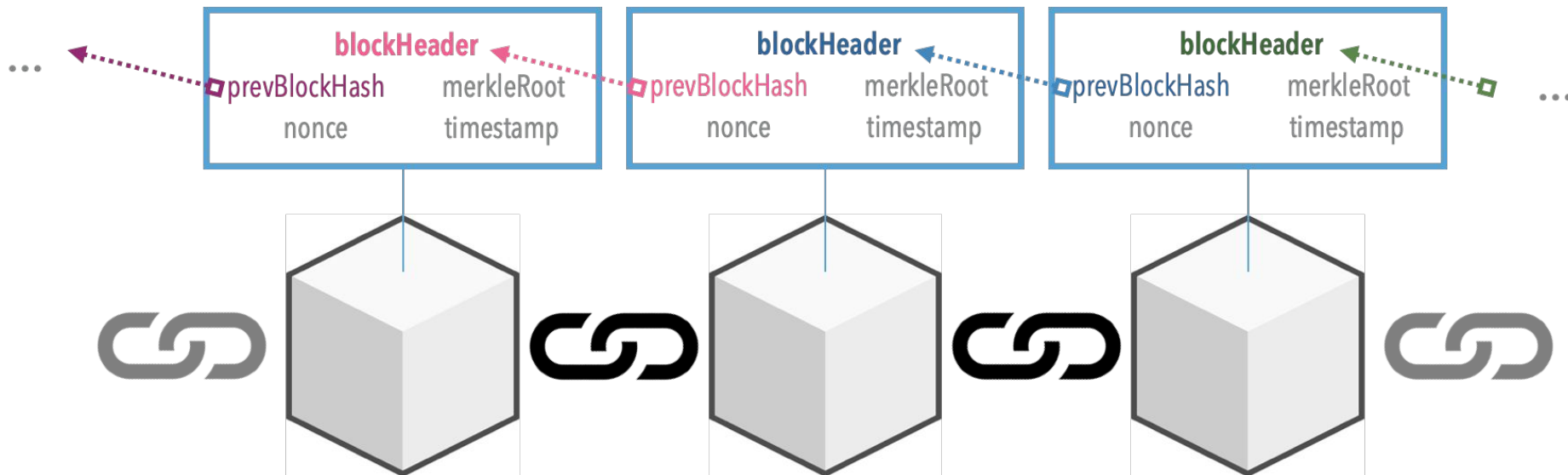


$\text{blockID} = H(\text{blockHeader}) = H(\text{prevBlockHash} || \text{merkleRoot} || \text{time} || \text{nonce} || \dots)$

Cabeçalho (*header*) de um bloco



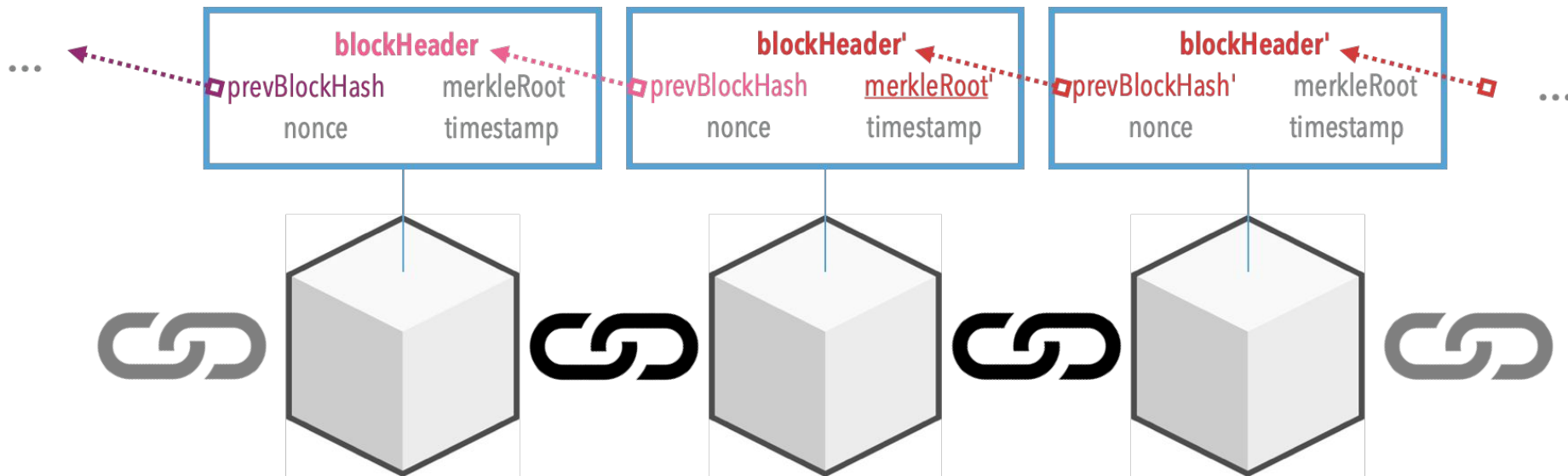
Cabeçalho (*header*) de um bloco



SHA256(SHA256(x))
→

$\text{prevBlockHash} = H(\text{prevBlockHash} || \text{merkleRoot} || \text{time} || \text{nonce})$

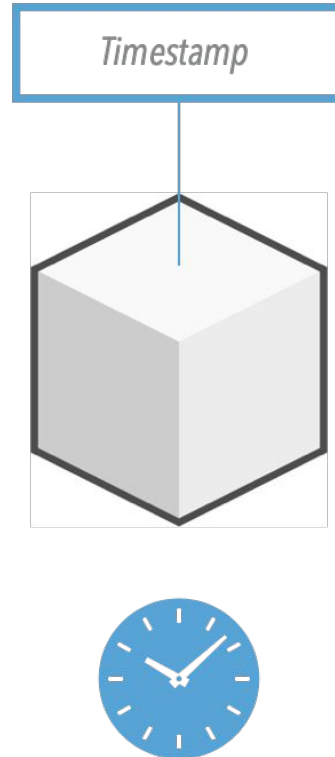
Cabeçalho (*header*) de um bloco



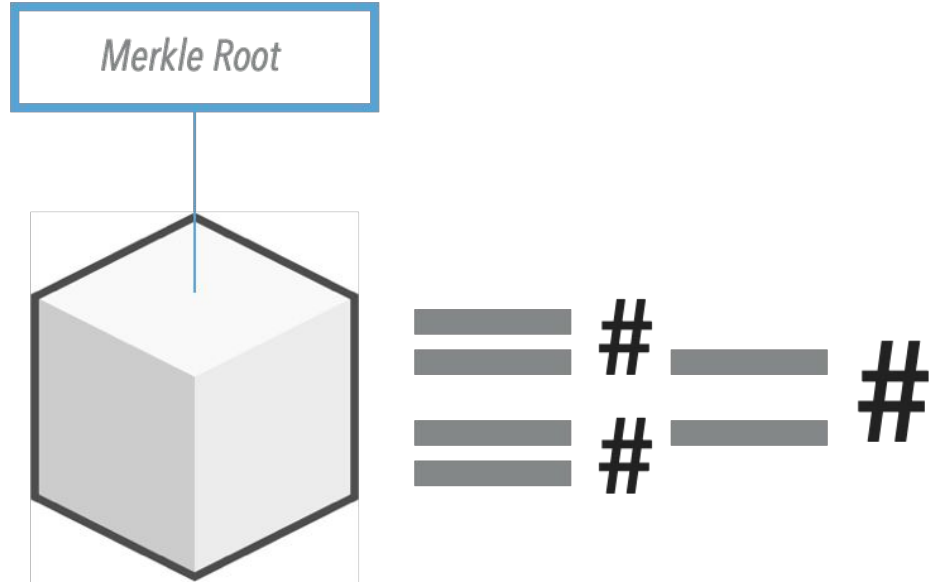
SHA256(SHA256(x))
→

prevBlockHash = H(**prevBlockHash** || merkleRoot || time || nonce)

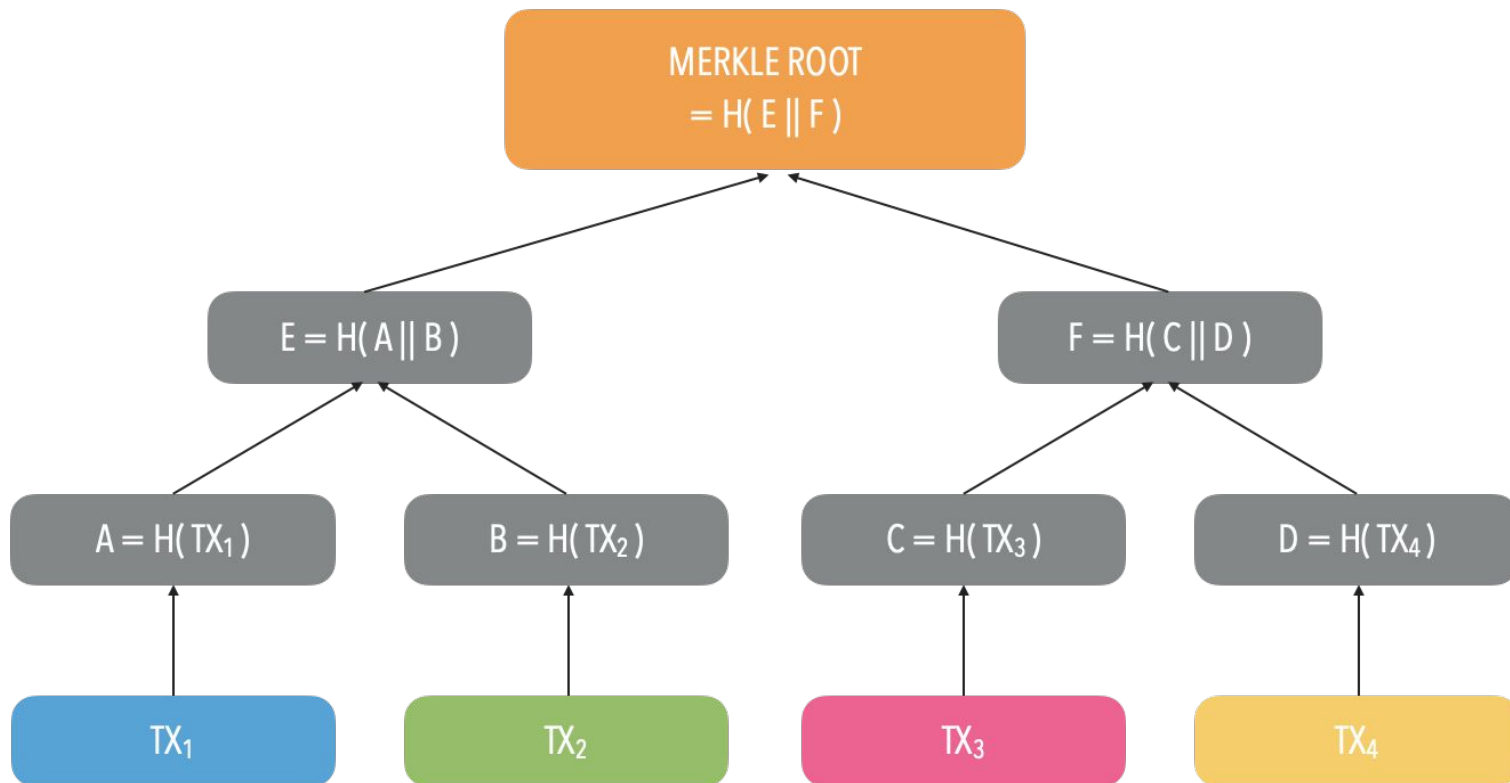
Cabeçalho (*header*) de um bloco



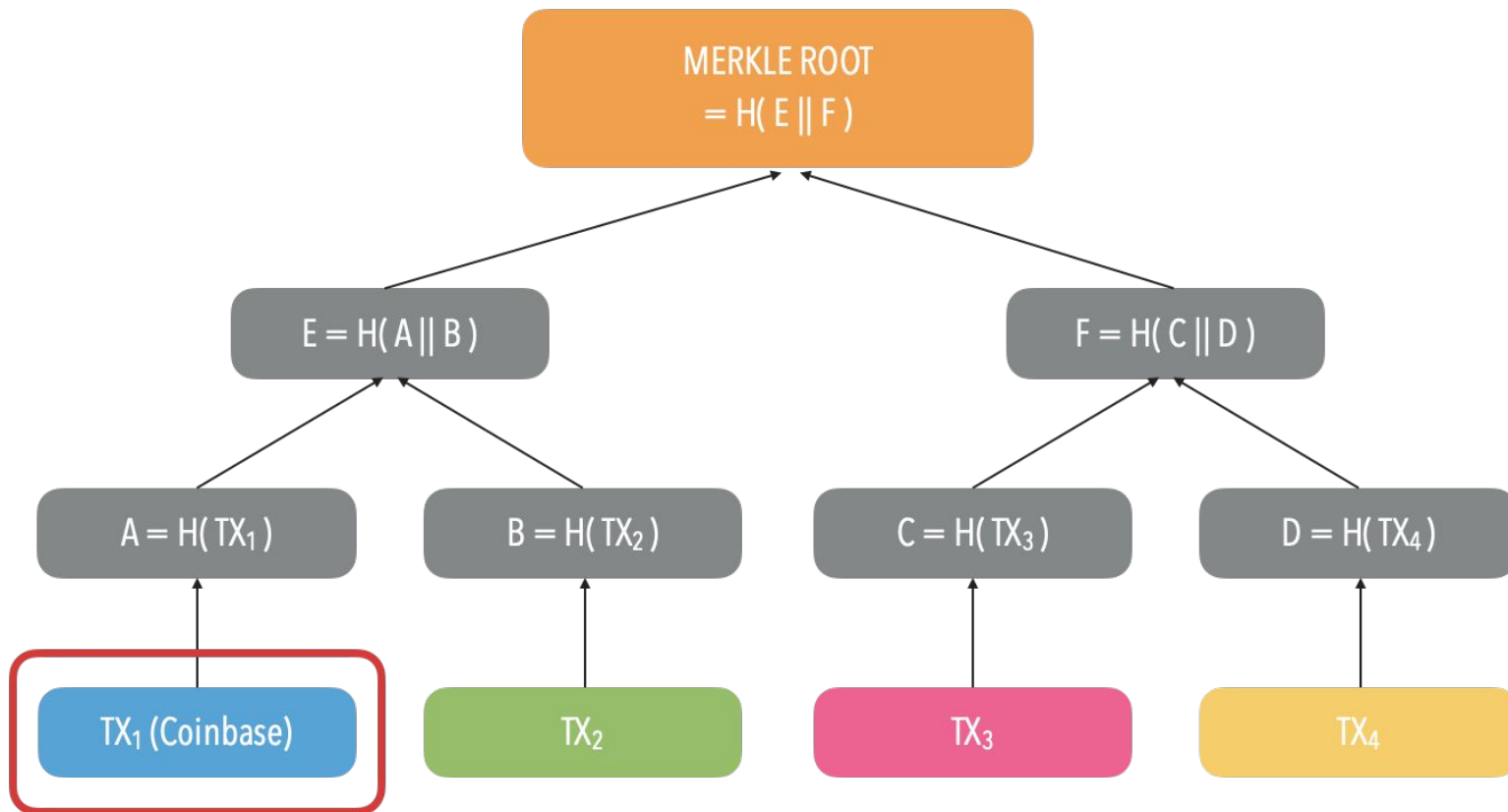
Cabeçalho (*header*) de um bloco



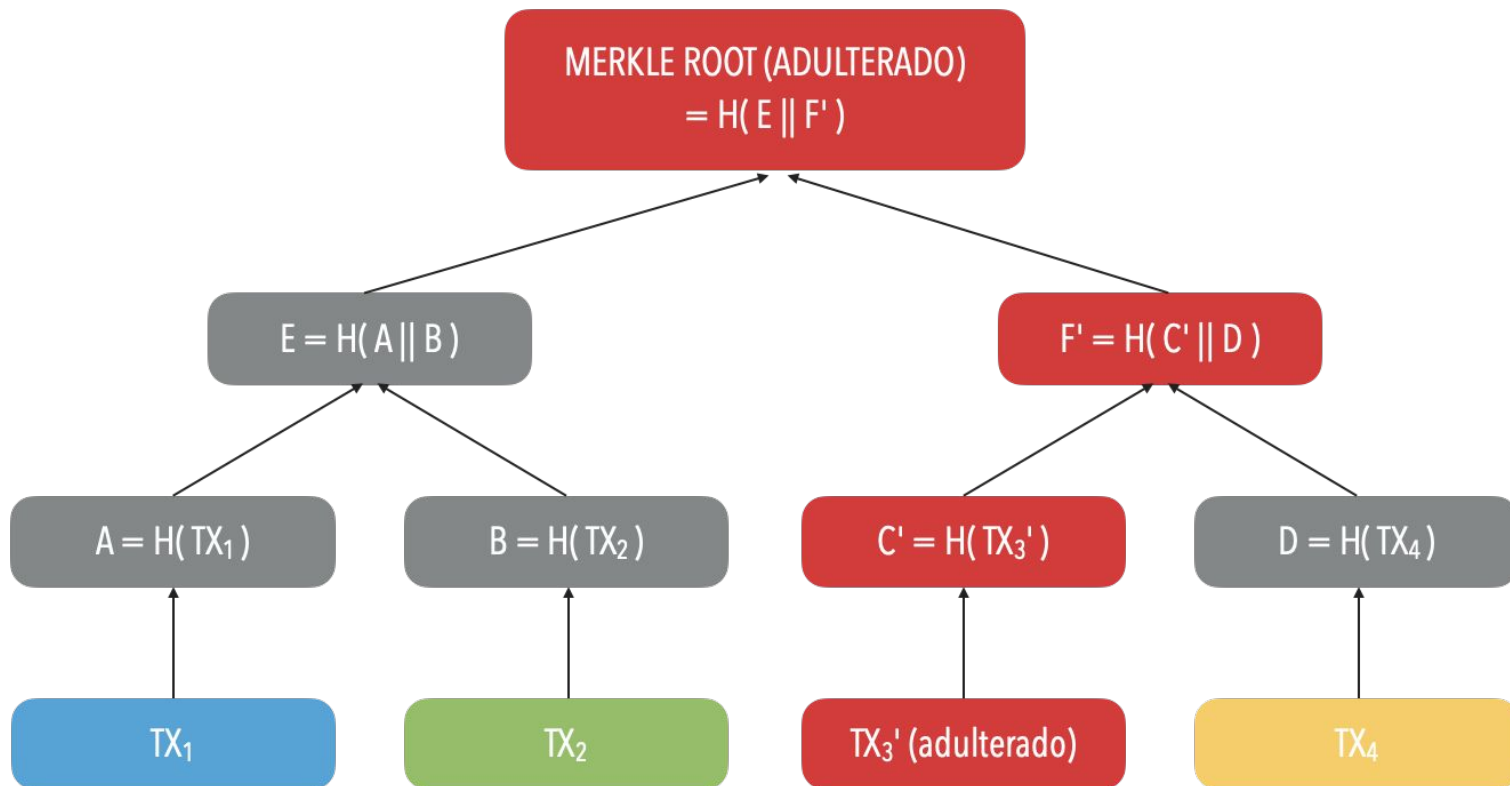
Merkle Root



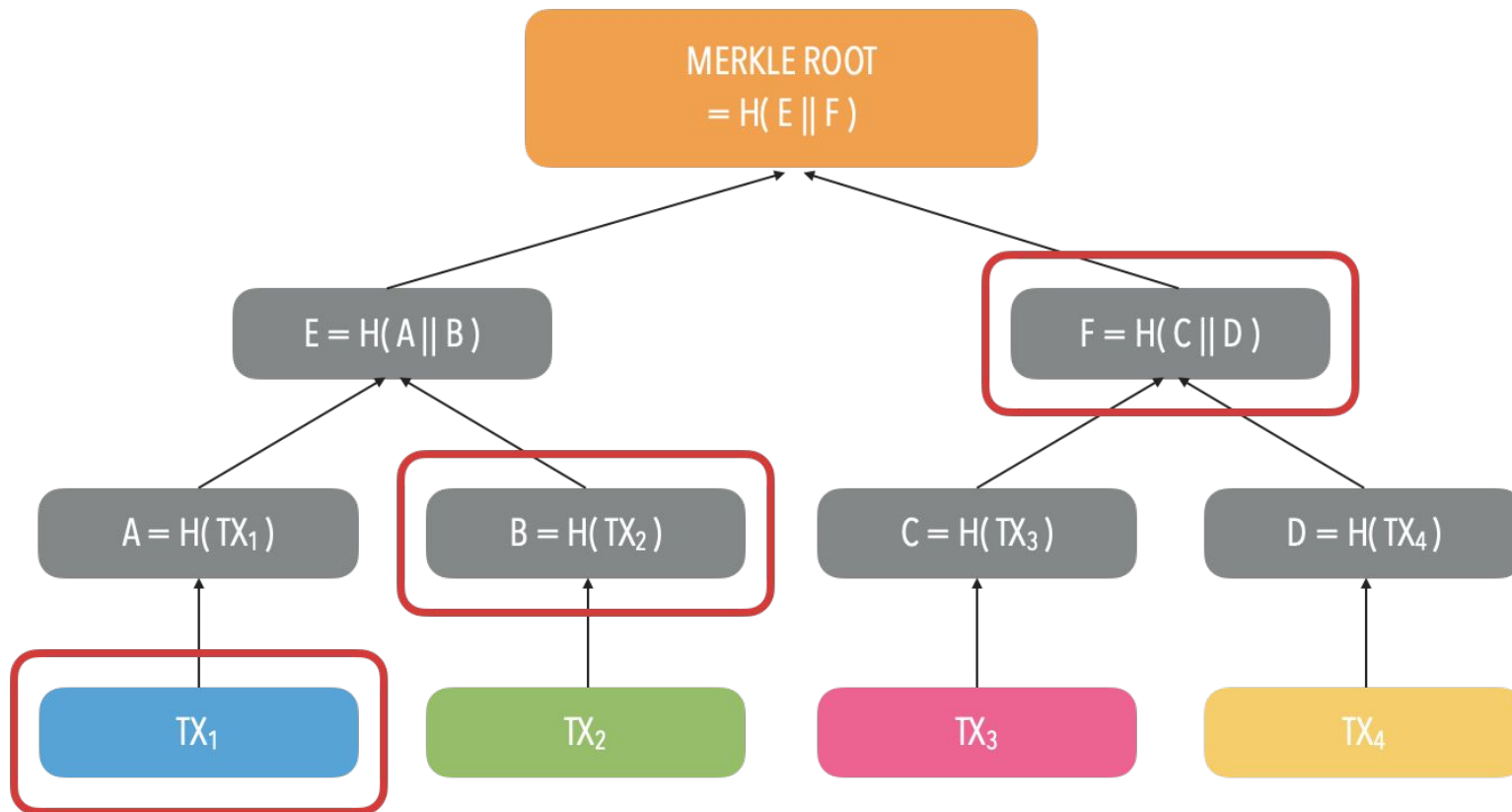
Merkle Root



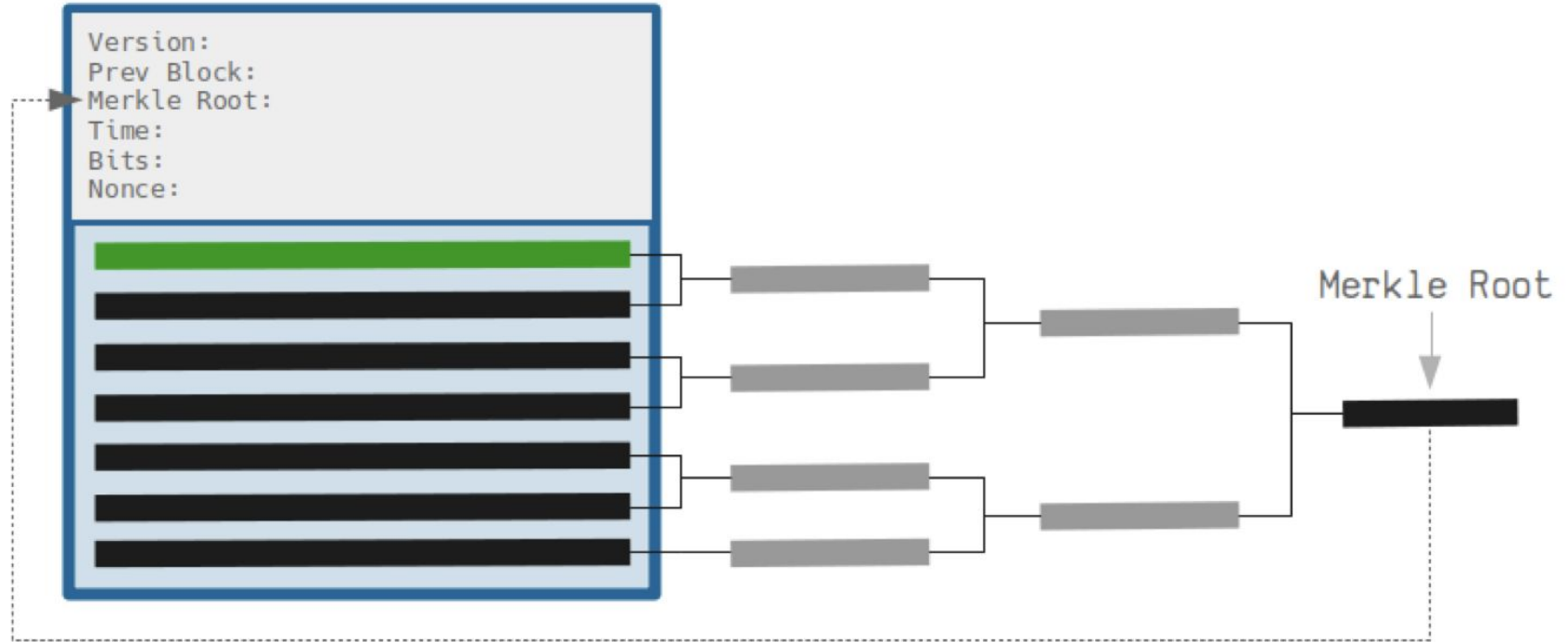
Merkle Root



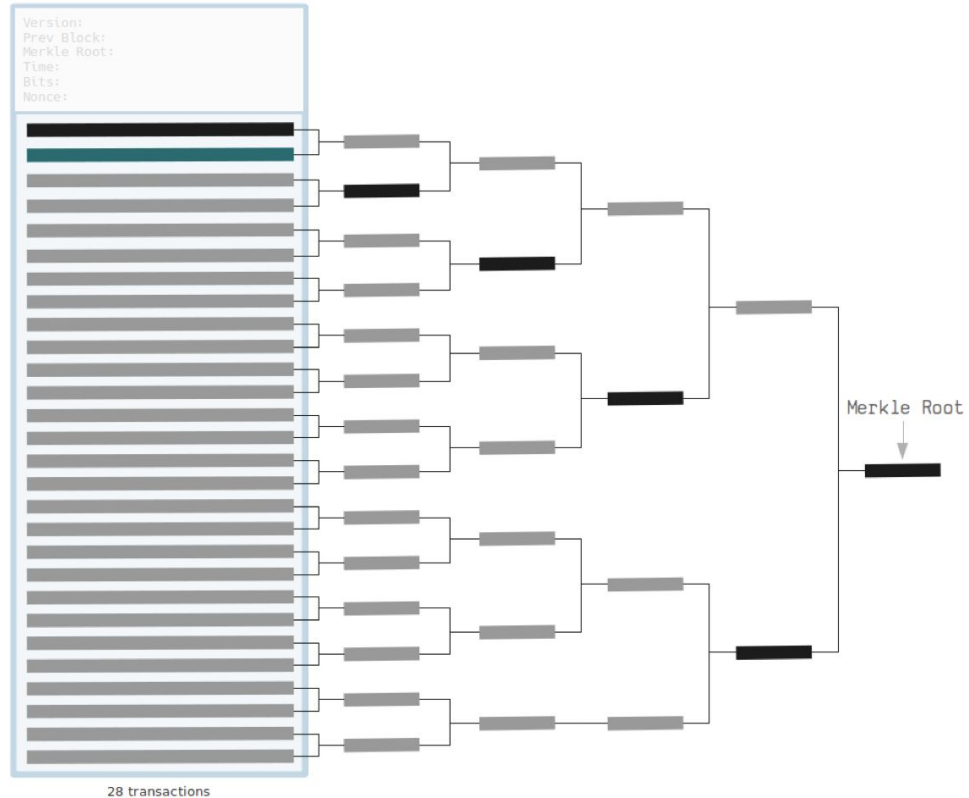
Merkle Root e *proof-of-inclusion*



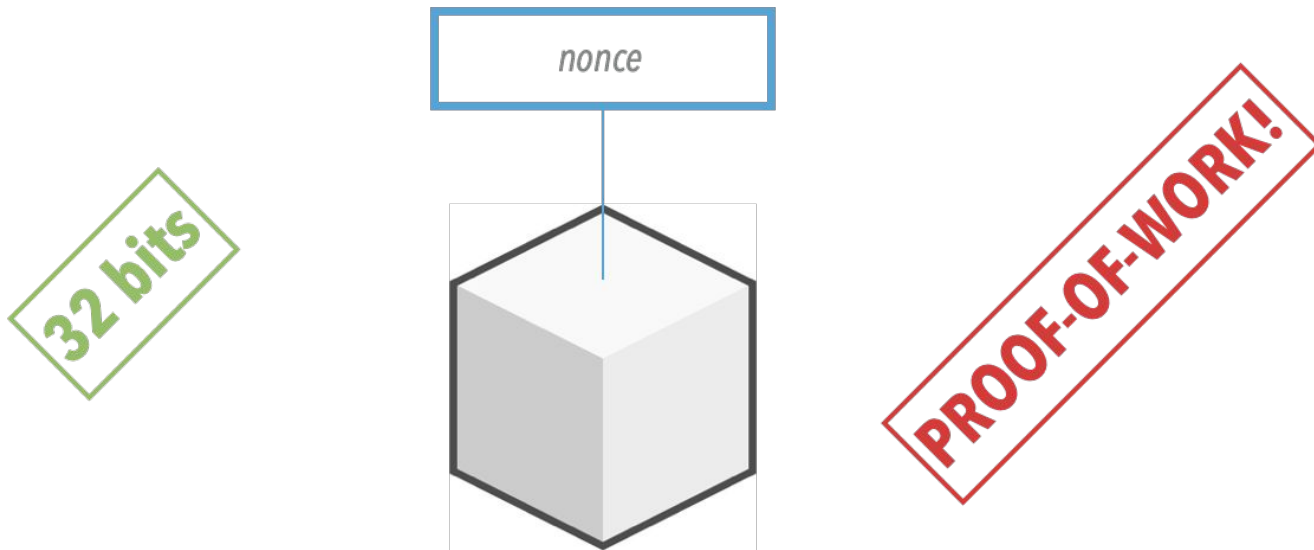
Merkle Root



Merkle Root



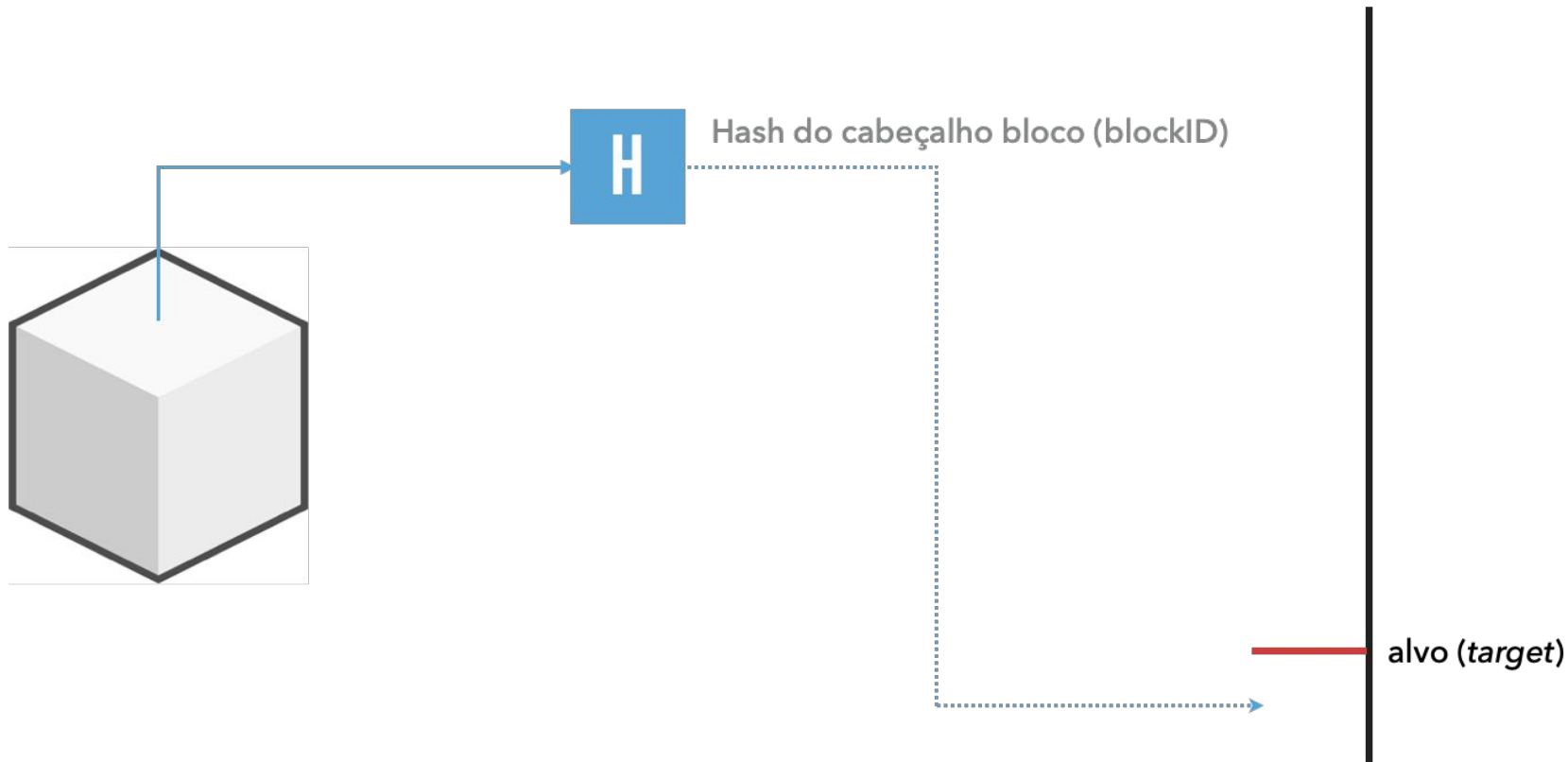
Cabeçalho (*header*) de um bloco



Enigma criptográfico *hash*: Encontrar um **nonce** que satisfaça a seguinte inequação:

$$H(\text{prevBlockHash} \ || \ \text{merkleRoot} \ || \ \text{time} \ || \ \text{nonce}) < \text{target}$$

Puzzle criptográfico baseado em *hash*



Puzzle criptográfico baseado em *hash*

`H(prevBlockHash || merkleRoot || time || nonce)`

`H("Hello, World!4250")`

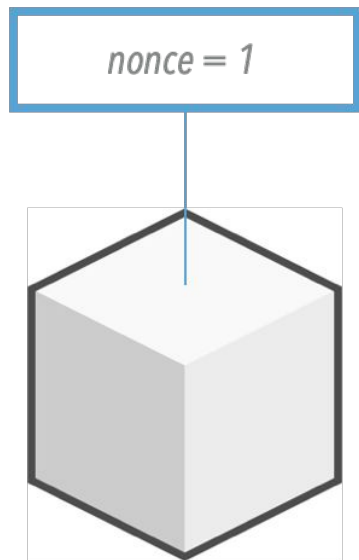
`0x0000c3af42fc31103f1fdc0151fa747ff87349a4714df7cc52ea464e12dcd4e9`

`<`

`0x0000ff`

resolvido!

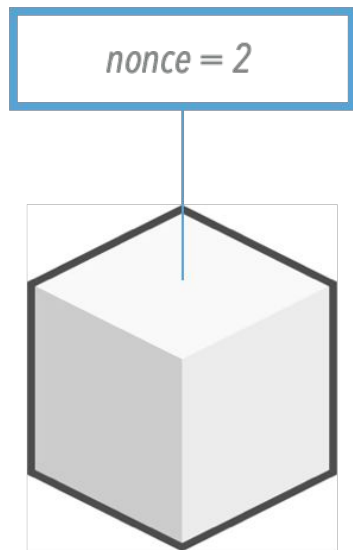
Cabeçalho (*header*) de um bloco



4c47c2d47712cc266c3b7ed7e9a0bcda2e6786f7455b9af3e9df3c5a2b26ddbfb



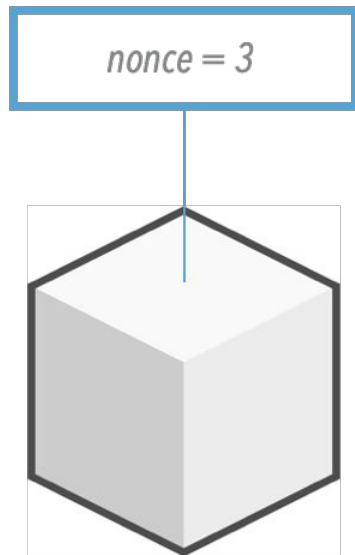
Cabeçalho (*header*) de um bloco



6bbe9136c059738eaaf237c995a78971788ee87119d82ef640a7288b43928017



Cabeçalho (*header*) de um bloco

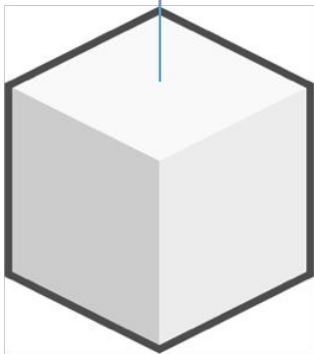


000004bb7c4d63435e1fa5595986fab643490560699bf35c43bdc6ecfd3ea721



Cabeçalho (*header*) de um bloco

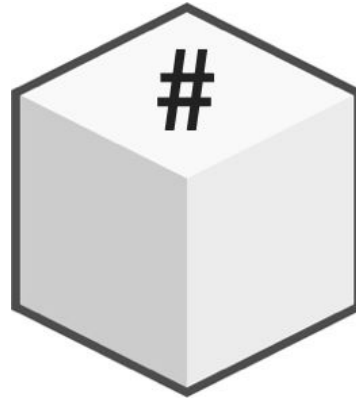
nonce = 1.619.820.810



00000000000000000274cb1a04c382475310f70cee3776af06414f22f8337044



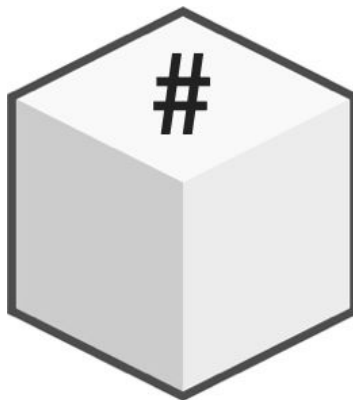
Dificuldade de um bloco



Dificuldade do bloco:

0000000HASHVALUE

Dificuldade de um bloco



2.016 blocos
≈
2 semanas

Dificuldade do bloco:

0000000HASHVALUE

```
H(prevBlockHash || merkleRoot || time || nonce) <
0x0000ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
0x0000000000000000ffffffffffffffffffffffffffffffffffffffffffffffffffffff
0x0000000000000000ffffffffffffffffffffffffffffffffffffffffffffffffffffff
```

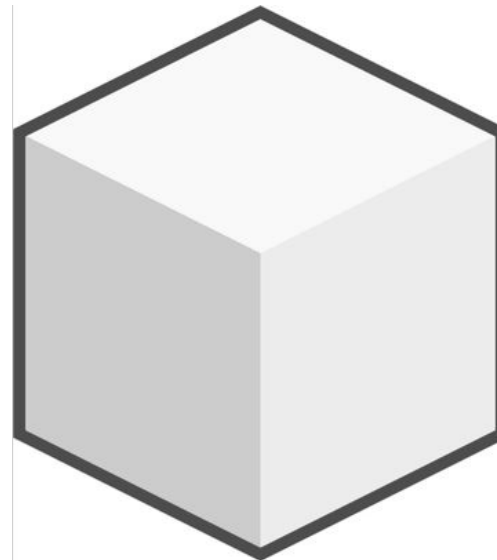
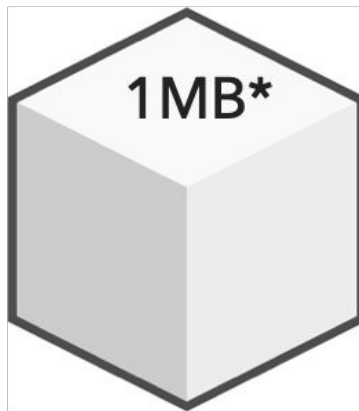
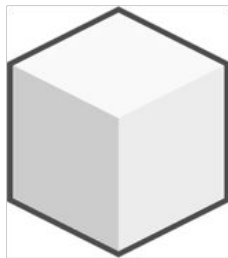
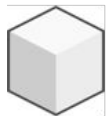
[illegible]

[illegible]

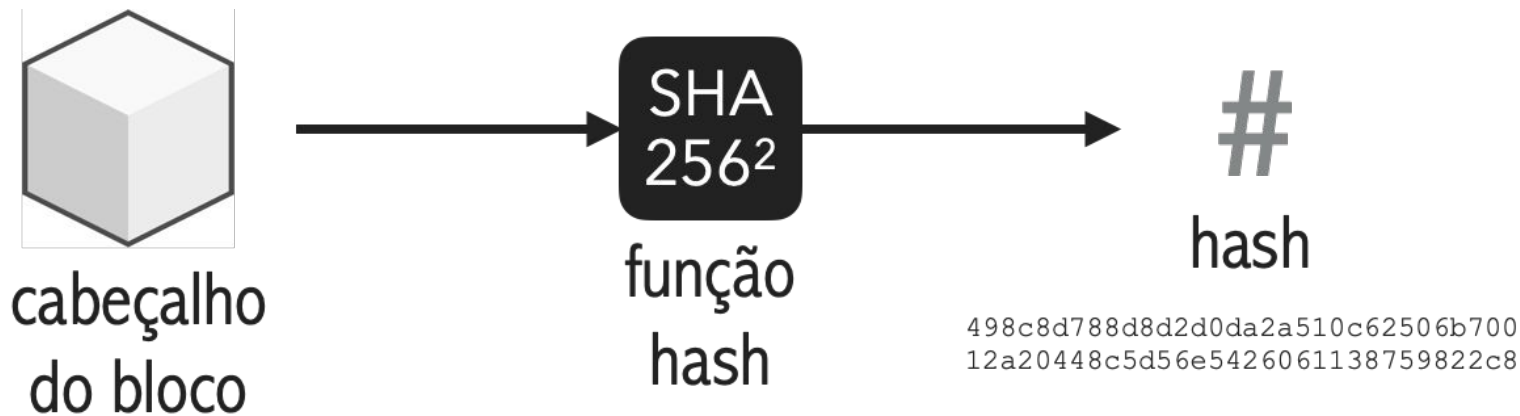
010000009500c43a25c624520b5100adf82cb9f9da72fd2447a496b
c600b00000000000006cd862370395dedf1da2841ccda0fc489e3039
de5f1ccddef0e834991a65600ea6c8cb4db3936a1ae3143991

Campo	Tamanho	Codificação
Version *	4 bytes	Little-Endian
Previous Block Hash	32 bytes	Little-Endian
Merkle Root	32 bytes	Little-Endian
Time	4 bytes	Little-Endian
Bits	4 bytes	Little-Endian
Nonce	4 bytes	Little-Endian

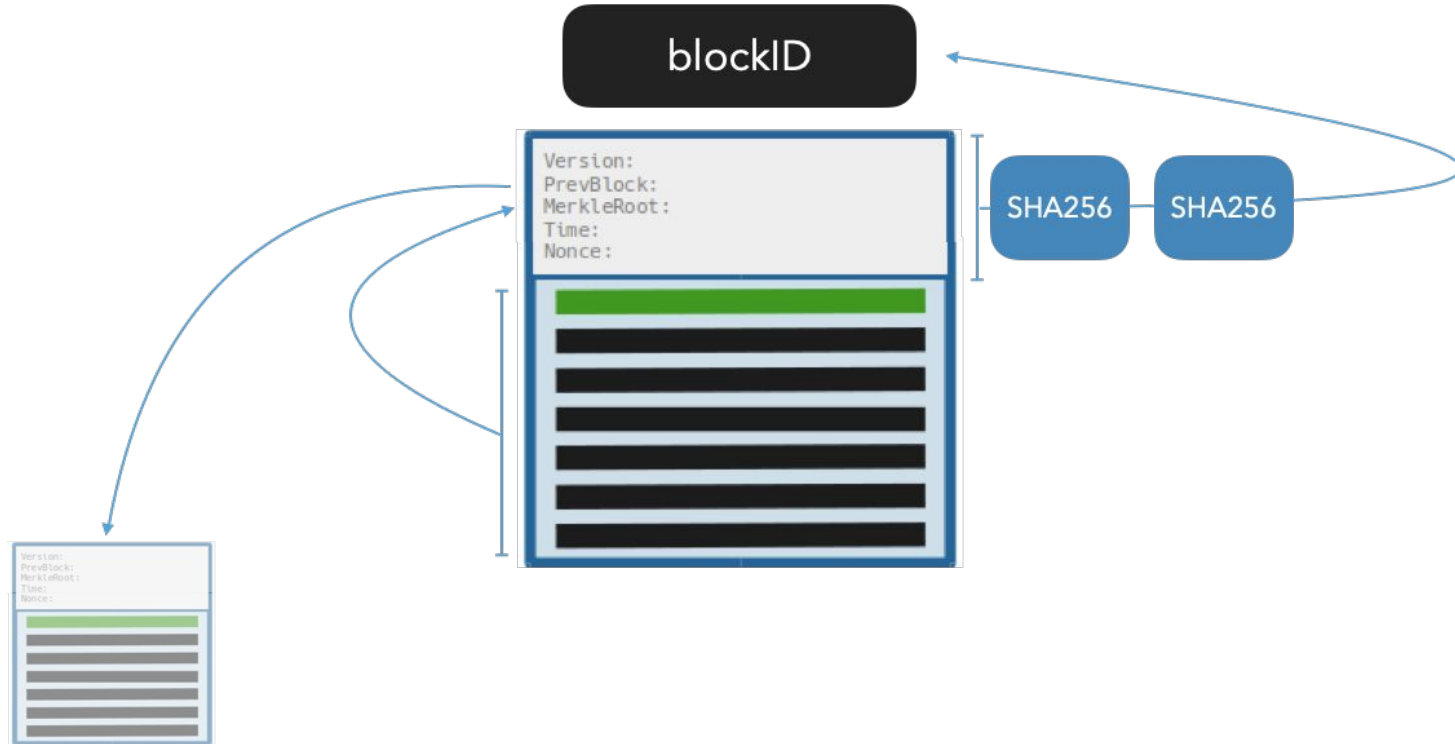
Tamanho de um bloco



Hash de um bloco



ID de um bloco



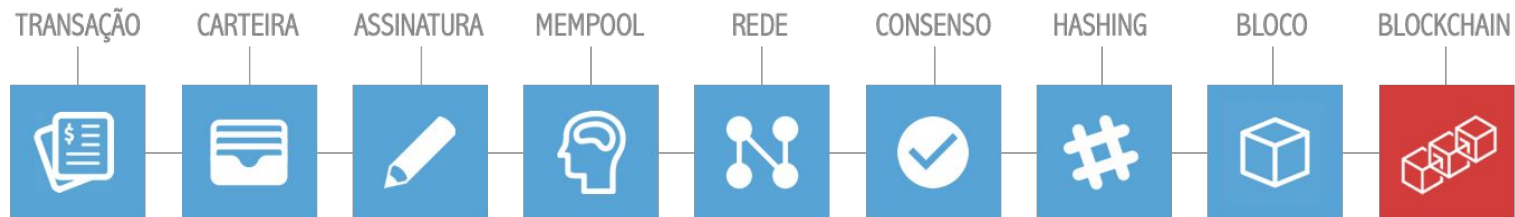
Exemplo: bloco #123456

<https://www.blockchain.com/explorer/blocks/btc/0000000000002917ed80650c6174aac8dfc46f5fe36480aaef682ff6cd83c3ca>

Bloco: demonstração

<https://andersbrownworth.com/blockchain/block>

ARQUITETURA DE UM **BLOCKCHAIN**



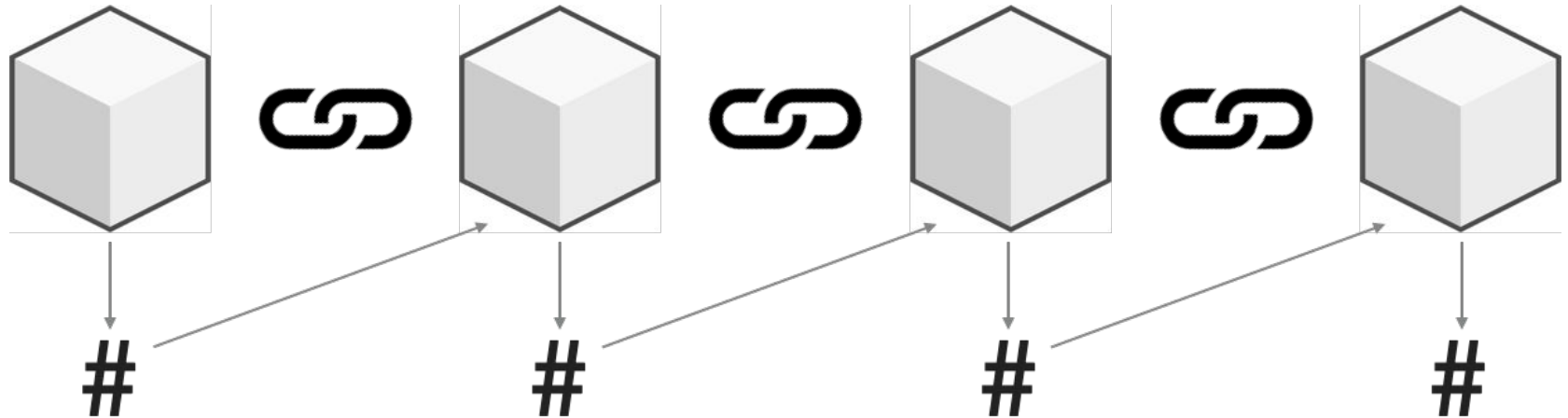
Blockchain

Um livro-razão digital e compartilhado que registra uma lista de transações no formato de uma sequência de blocos.

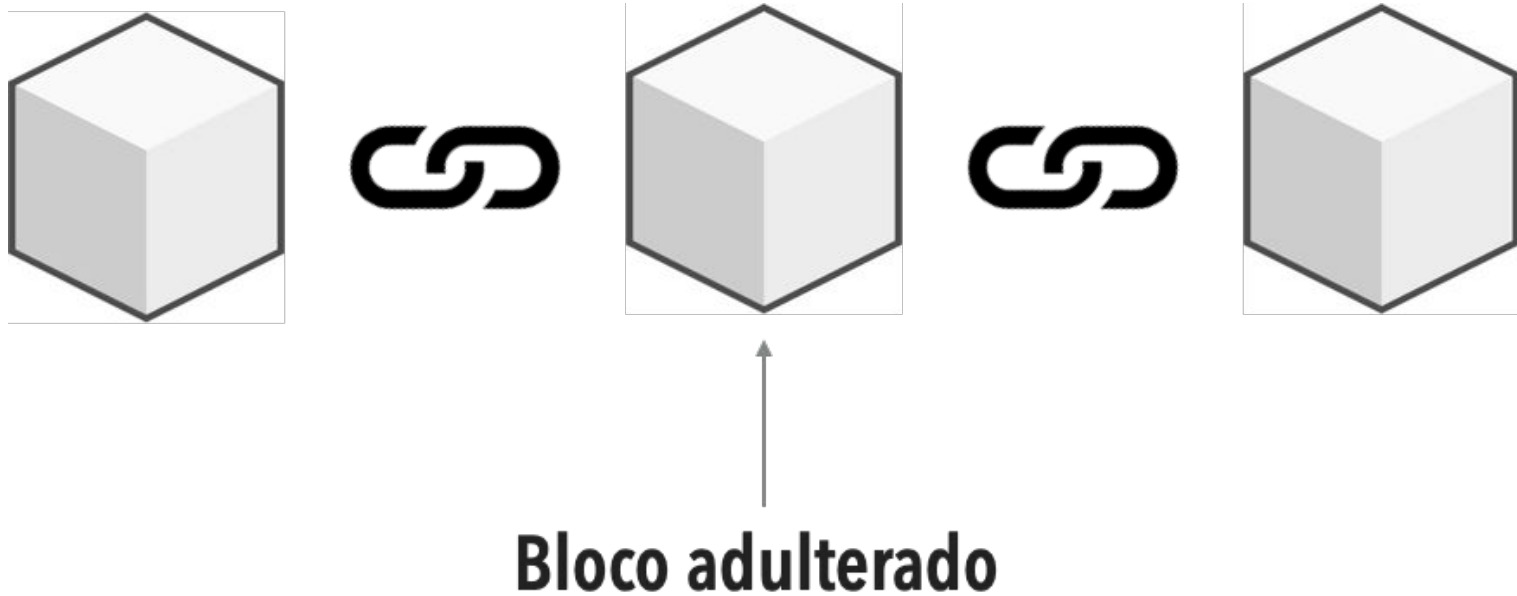
"Corrente" de blocos



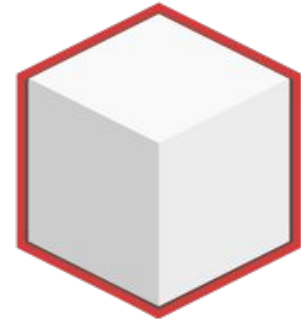
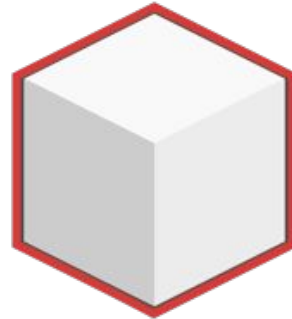
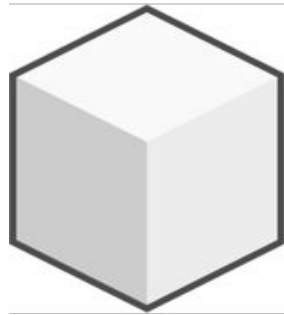
"Corrente" de blocos



"Corrente" de blocos



"Corrente" de blocos



Blocos inválidos

Bloco adulterado

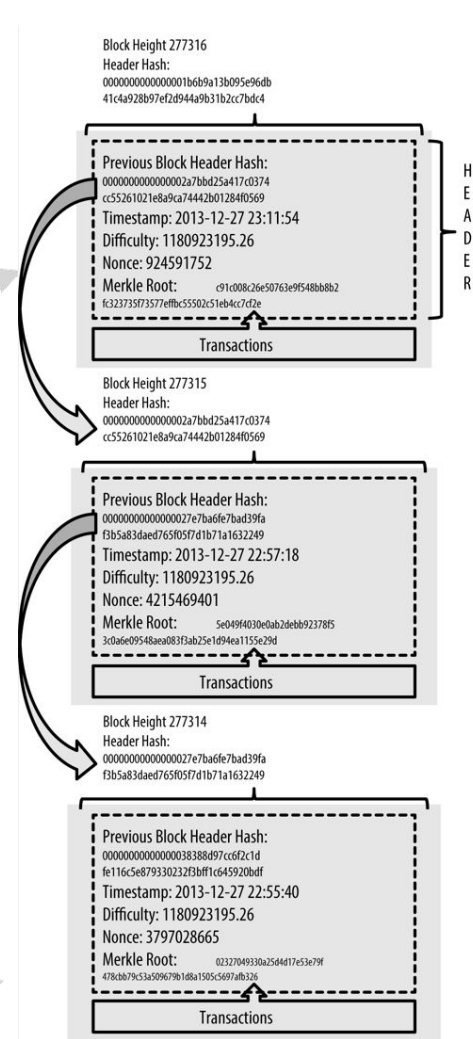
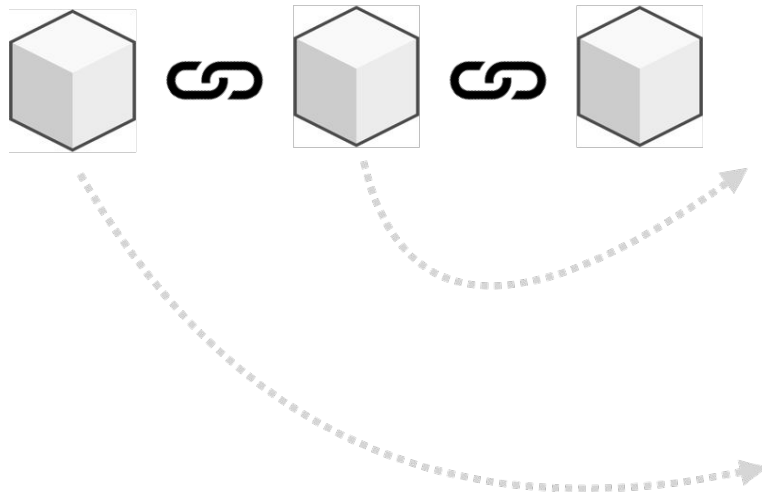
Número do bloco



Bloco "gênesis"



<https://www.blockchain.com/btc/block/000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f>



Blockchain: demonstração

<https://andersbrownworth.com/blockchain/blockchain>

[illegible]

Blocos em Python

<https://docs.python.org/3/tutorial/datastructures.html#dictionaries>

```
block = {
    'index': 2,
    'timestamp': 1506057125,
    'nonce': 324984,
    'merkle_root': "13c8bbf1dde38d5f86bfc48a5c027df0d8eb19c8a647de49976755e1b35b31ca",
    'previous_hash': "2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824",
    'transactions': [
        {
            'sender': "8527147fe1f5426f9dd545de4b27ee00",
            'recipient': "a77f5cdfa2934df3954a5c7c7da5df1f",
            'amount': 500000,
        }
    ]
}
```

```
block_header = {
    'index': 2,
    'timestamp': 1506057125,
    'nonce': 324984,
    'merkle_root': "13c8bbf1dde38d5f86bfc48a5c027df0d8eb19c8a647de49976755e1b35b31ca",
    'previous_hash': "2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824"
}
```

Atividade avaliativa #02

GitHub Classroom

`/02-blocks/`

