01

# SISTEMA DISTRIBUÍDO


NOMES:

Caroline Yumi Uehara
Lucas Kenji Uezu
Leonardo de Jesus Diz Conde
Victor Yuji Saito

# BLOCKCHAIN

## Agenda de Hoje

- História
- Algoritmos de consenso
- Smart contract e Dapp
- POC

# História

Linha do tempo



1992 — *How To Time-Stamp a Digital Document* (HABER e STORNETTA)

1992 — *Improving the Efficiency and Reliability of Digital Time-Stamping* (BAYER, HABER e STORNETTA)

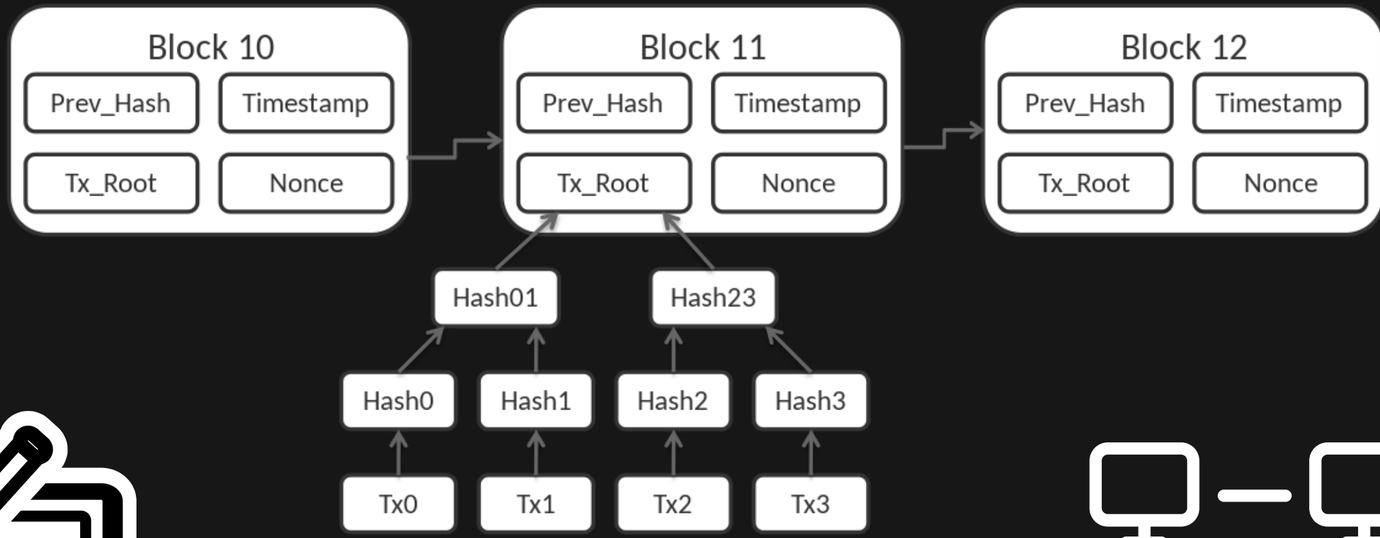
2002 — *Hashcash - A Denial of Service Counter-Measure* (BACK)

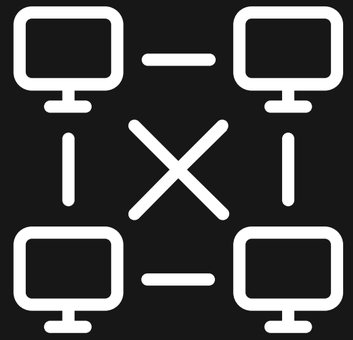
2008 — *Bitcoin: A Peer-to-Peer Electronic Cash System* (NAKAMOTO)

# BLOCKCHAIN

## Breve Introdução

- *Rede/Arquitetura P2P*
- *Ledger (livro-razão)*
- *Hashes*



# ALGORITMOS DE CONSENSO

Os mais utilizados são:

- Proof of Work
- Proof of Stake
- Practical Byzantine Fault Tolerance
- Delegate Proof of Stake

Utilizado na blockchain Neo:

- Delegated Byzantine Fault Tolerance

# PROOF OF WORK

BITCOIN

DOGECOIN

MONERO

- Conceito
  - Surgiu em 1993
  - Funções vinculados à memória
- Bitcoin
  - Rede da blockchain
    - Validação de 50%
    - Mineração
- Vantagens e Desvantagens
  - Fácil validação
  - Difícil de calcular
  - Alto custo de energia

# PROOF OF STAKE

ETHEREUM

CARDANO

BINANCE COIN

- Conceito
  - Surgiu em 2012
  - Sorteio de usuário para gerar o próximo hash
    - Proporcional
  - Validação do hash
- Consome menos recursos computacionais

# DELEGATE PROOF OF STAKE

SOLANA

TERRA

TEZOS

- Conceito
  - Surgiu em 2014
  - Evolução do Proof of Stake
  - Usuários da blockchain podem votar em quem deve fazer a validação do hash gerado
  - Usuários que tentarem forjar o hash serão punidos

ZILLIQA

# PRACTICAL BYZANTINE FAULT TOLERANCE

HYPERLEDGER FABRIC

- Byzantine Fault
  - 1978
  - Condição particular
    - Computação distribuída
  - Inconsistências
    - Apresenta sintomas diferentes para diferentes observadores
- Vantagens
  - Seguro em ambientes assíncronos
  - Defesa contra Byzantine Fault

NEO 2

# DELEGATED BYZANTINE FAULT TOLERANCE

NEO 3

- Conceito
  - Utilizado na Blockchain Neo
  - Baseado no Practical Byzantine Fault Tolerance
- Participação em larga escala
- Votação

# Smart Contracts e Dapp

- Atomicidade das operações
- Armazenados na blockchain
  - Transparência
  - Imutável
- Dapp

# ATIVIDADE PRÁTICA

NO LABORATÓRIO

**PROPOSTA:**

Criar um Smart Contract básico em Python para uma rede privada blockchain Neo.

**LINK GITHUB:**

# Referências:




- ALIAGA, Y. E. M.; HENRIQUES, M. A. A. Uma comparação de mecanismos de consenso em blockchains. University of Campinas. Campinas. 2017.
- HABER, S.; STORNETTA, W. S. How to time-stamp a digital document. Journal of Cryptology, p. 99-111, 1991. Disponivel em: <https://link.springer.com/content/pdf/ 10.1007/BF00196791.pdf>. Acesso em: 11 Junho 2022.
- BAYER, D.; HABER, S.; STORNETTA, W. S. Improving the Efficiency and Reliability of Digital, Março 1992. Disponivel em: <https://www.math.columbia.edu/ ~bayer/papers/Timestamp_BHS93.pdf>. Acesso em: 11 Junho 2022.
- BACK, A. Hashcash - A Denial of Service Counter-Measure. hashcash.org, 1 Agosto 2002. Disponivel em: <http://www.hashcash.org/papers/hashcash.pdf>. Acesso em: 12 Junho 2022.

# Referências:




- ALIAGA, Y. E. M.; HENRIQUES, M. A. A. Uma comparação de mecanismos de consenso em blockchains. University of Campinas. Campinas. 2017.
- IBM. IBM. What are smart contracts on blockchain?, 11 Junho 2022. Disponivel em: <https://www.ibm.com/topics/smart-contracts>.
- NAKAMOTO, S. bitcoin. Bitcoin: A Peer-to-Peer Electronic Cash System, 21 Outubro 2008. Disponivel em: <https://bitcoin.org/bitcoin.pdf>. Acesso em: 4 Junho 2022.
- https://docs.neo.org/v2/tutorials/en-us/7-consensus/3-PBFT_and_DBFT.html

# OBRIGADO.