Fundamentos e desafios do blockchain 3.0

Erikson J. de Aguiar

erjulioaguiar@usp.br

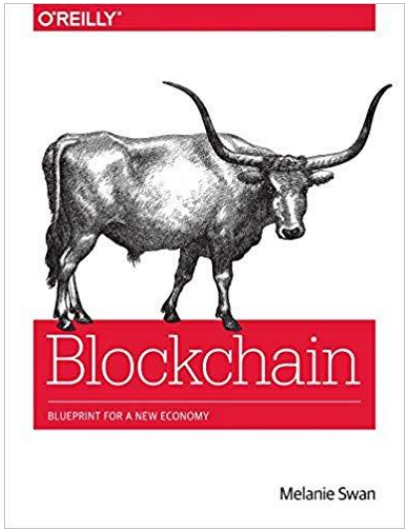
Quem sou eu ?

- Aluno de mestrado no **ICMC - USP**
- BSc. Ciência da Computação - **UENP**
- **Áreas de concentração:**
  - Sistemas distribuídos
  - Redes
  - IoT
  - Blockchain
  - Privacidade

https://eriksonjaguiar.github.io/

Introdução ao blockchain

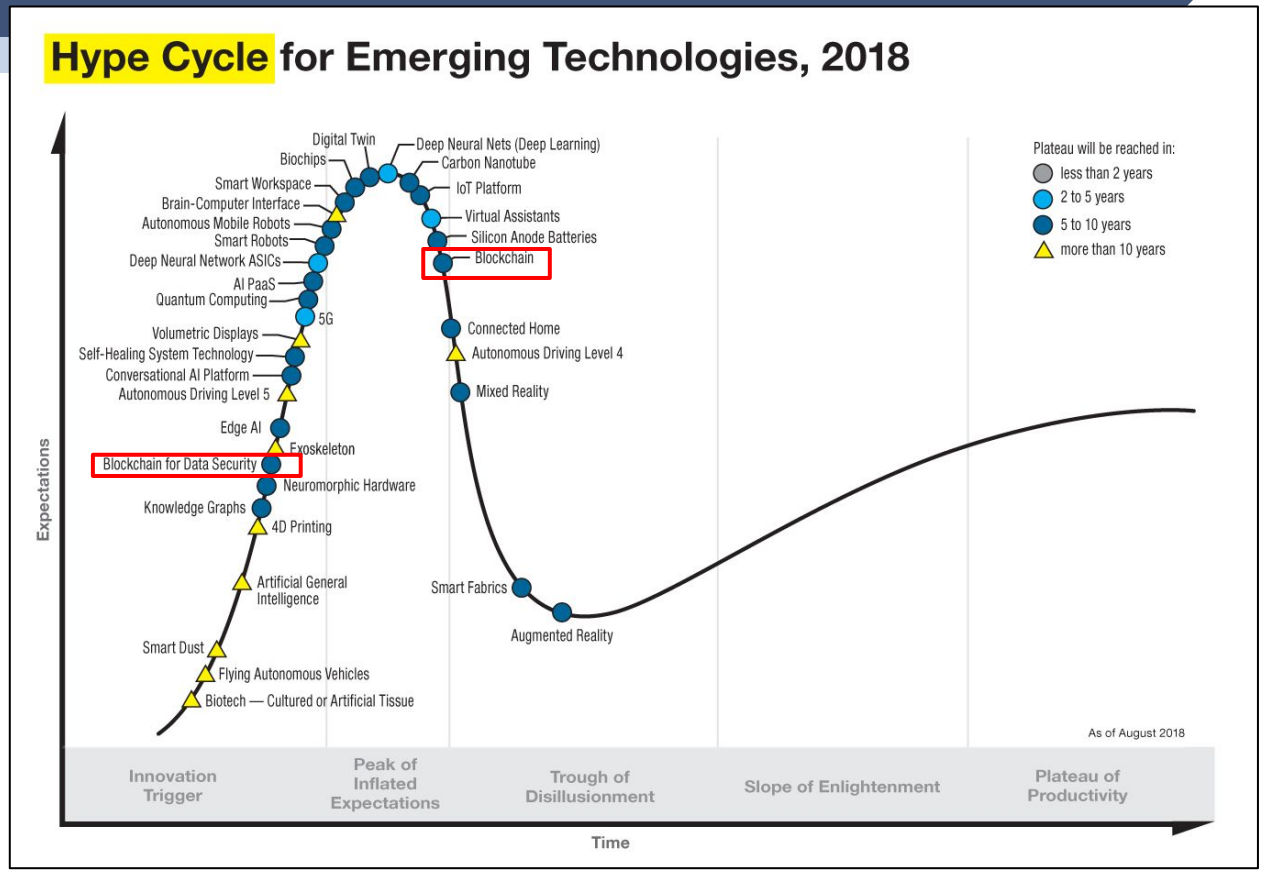
Por que blockchain 3.0 ?



(SWAN, 2015)

- Classificação de blockchain:
  - blockchain 1.0: Bitcoin
  - blockchain 2.0: Contratos inteligentes
  - blockchain 3.0: vários campos de aplicação (Atual)
- **Blockchain não é somente Bitcoin**

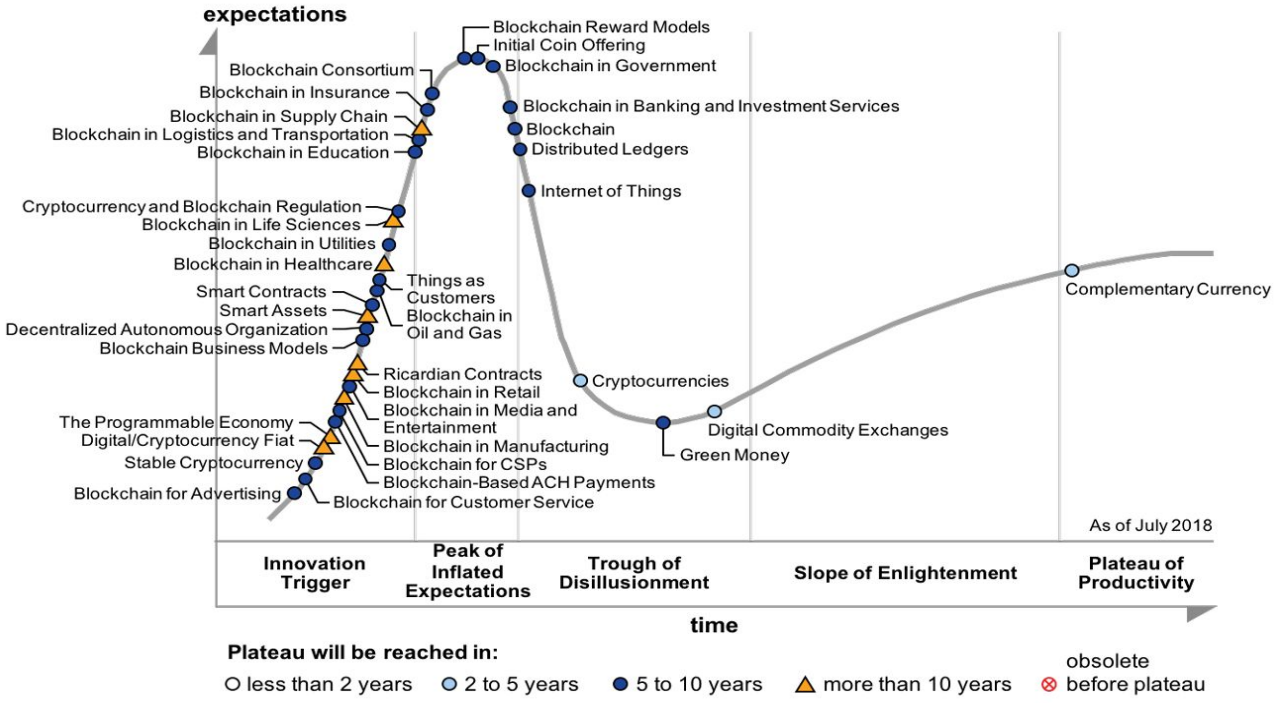
Potencial do blockchain



Fonte: (GARTNER, 2018)

Potencial do blockchain

Hype Cycle for Blockchain Business, 2018



Fonte: (GARTNER, 2018)

Potencial do blockchain

- **2009-2020:** Integração do blockchain nas empresas com as tecnologias existentes
- **2016-2020:** Trata de aplicações específicas que utiliza de tokens e descentralização
- **2021-2025:** Soluções completamente baseadas em blockchain
- **Depois de 2025:** Contratos autônomos, DAOs e microtransações

Fonte: (GARTNER, 2018)

Arquitetura do Bitcoin

- Apresenta a primeira arquitetura do blockchain (2008-2009)
- Criado por Satoshi Nakamoto
- Tem um livro-razão público
- Utiliza o protocolo de consenso PoW (Mineração)
- A moeda digital é um token que é trocado entre os participantes da rede

Fonte: (NAKAMOTO, 2008)

Cyberpunks

- Há uma teoria que eles criaram o blockchain
- Pessoas envolvidas no movimento Cyberpunk:
  - Adam Back - Inventor do Hashcash
  - Nick Szabo - Contratos inteligentes
  - Bram Cohem - Criador do BitTorrent
  - Julian Assange - Criador do Tor

Grupos que trabalham com blockchain

- **ICMC/USP - Prof. Dr. Jó Ueyama (Brasil)**
- Unifesp e ITA (Brasil)
- USC - Prof. Dr. Bhaskar Krishnamachari (EUA)
- ETH Zurich - Prof. Dr. David Basin (Suíça)
- Stanford - Prof. Dr. David Mazières (EUA)

Disciplina na pós-graduação

- **Nome:** SSC5964 - Plataformas de Criptomoedas: Abordagens e Aplicações
- **Horário:** Terça-feira 13:00 - 16:00

Características do blockchain

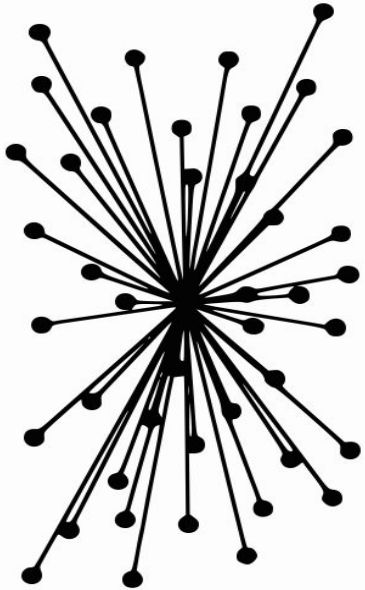
Definição do blockchain

- É uma tecnologia de livro-razão distribuída - DLT
- Tem como estrutura de dados blocos encadeados
- É uma tecnologia de livro-razão replicado e distribuído
- Fundamentada em um rede P2P

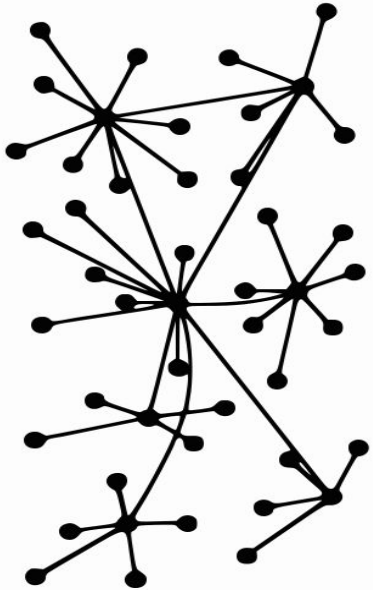
Fonte: (RIFI et al., 2017)

Definição do blockchain

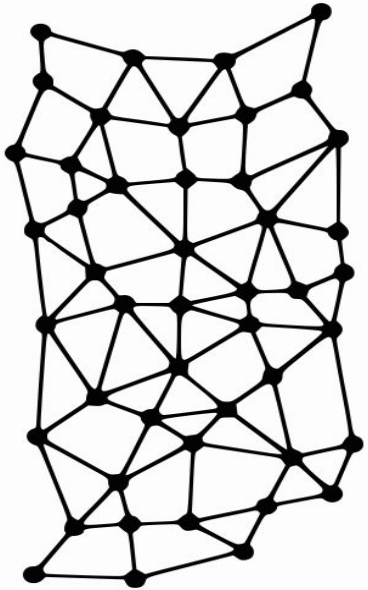
Tipos de rede



Centralized



Decentralized



Distributed

Fonte:
<https://diglife.com/decentralization/>

Conceitos básicos

- Hash
- Criptografia de chave pública
- Redes P2P
- Árvore de Merkle
- bloco

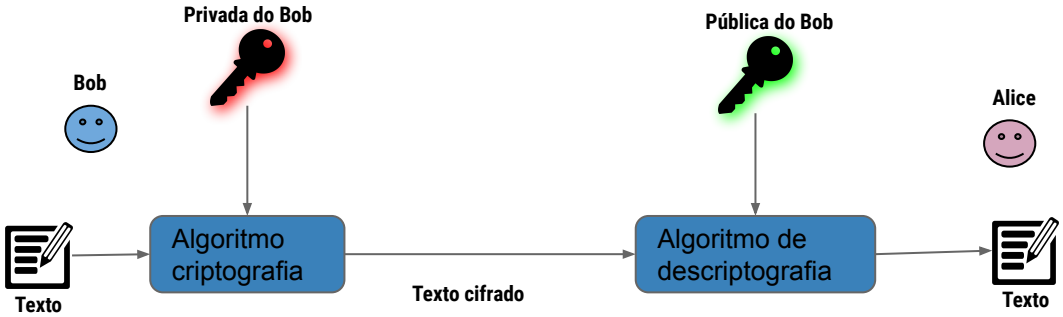
Conceitos básicos: Funções Hash

- Definição:
  - h = H(m)
  - m = mensagem de tamanho variável
  - Não tem função inversa
  - Algoritmo SHA-256 - Ferramenta

Fonte: (STALLINGS, 2017)

Conceitos básicos: Criptografia de chave pública

- Chave pública e privada
- Criptografia de ponta a ponta (WhatsApp)



Fonte: (STALLINGS, 2017)

Conceitos básicos: Redes P2P

- É um paradigma para a construção de sistemas distribuídos
- O peers da rede podem fazer o papel de cliente e servidor
- Utiliza do conceito de Overlays:
  - **Estruturado:** Tem um esquema de ID e nó
  - **Não-estruturado:** Conecta a nós arbitrários

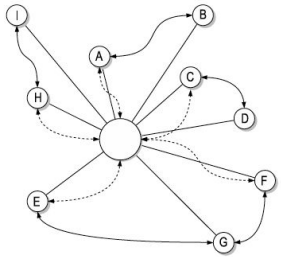
Fonte: (COULORIS, 2014) e (BARCELLOS; GASPARY, 2006)

Conceitos básicos: Redes P2P

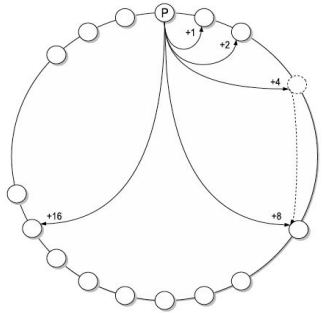
Busca: inundação ou aleatória.

- **Não estruturados:** Napster e BitTorrent, Gnutella e FastTrack
- **Estruturado:** Chord, CAN e Tapestry

Busca: Tabela Hash auxilia nesse processo.



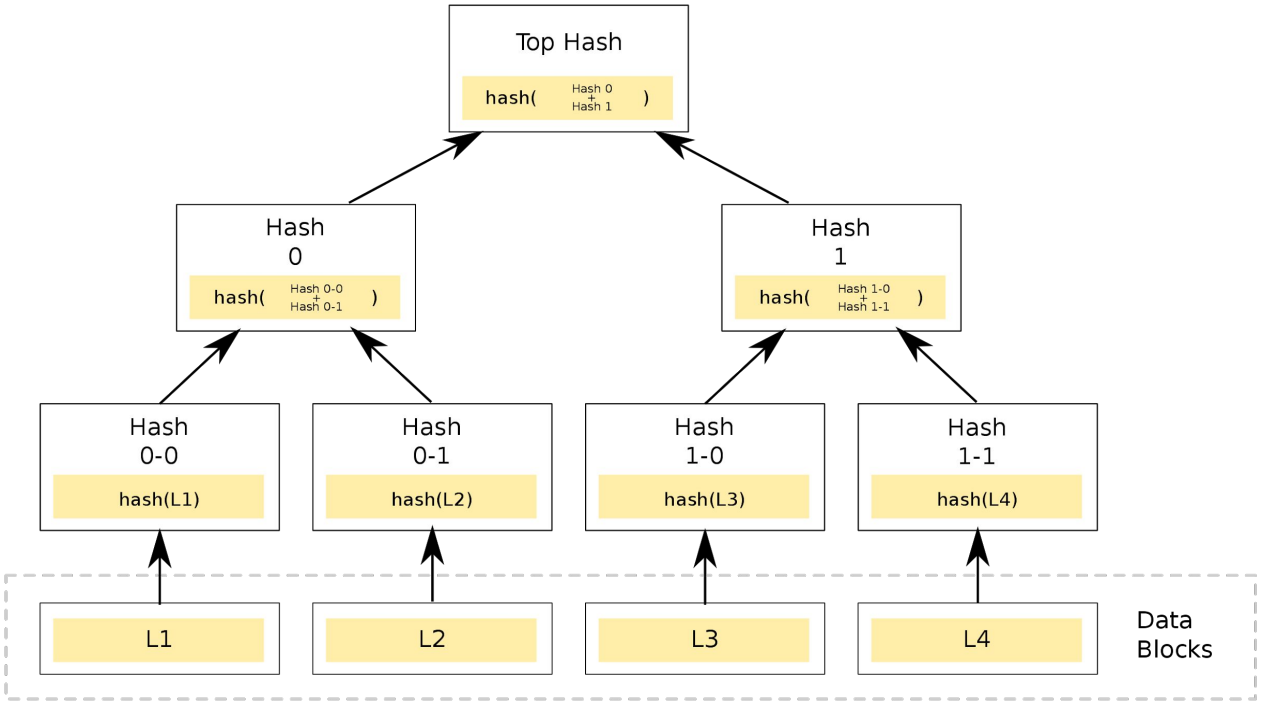
BitTorrent



Chord

Fonte: (BARCELLOS; GASPARY, 2006)

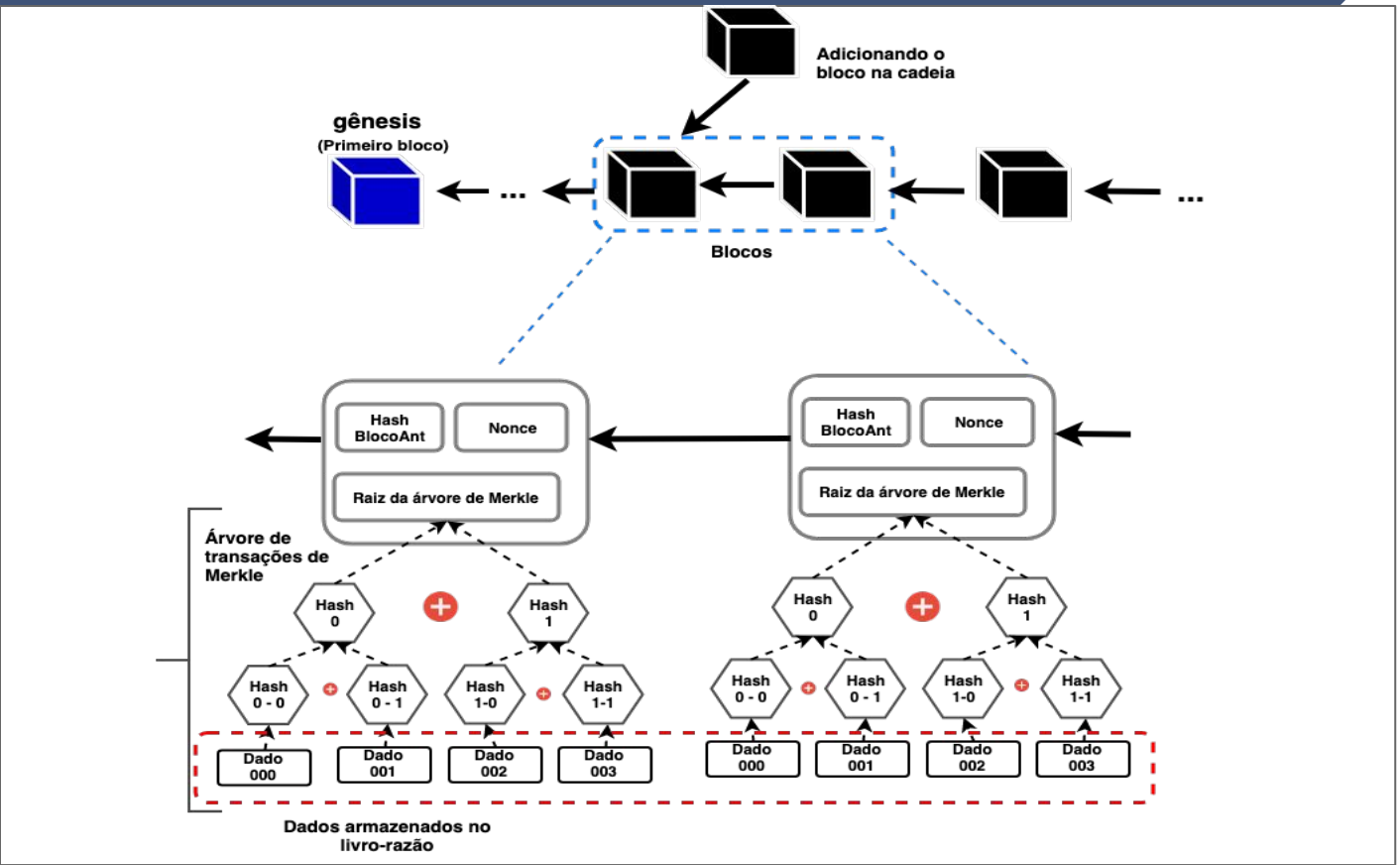
Conceitos básicos: Árvore de Merkle



Fonte:

<https://en.wikipedia.org/wiki/Merkle_tree>

Conceitos básicos: bloco



Fonte: (DINH et al., 2017)

Conceitos básicos: bloco

```python
import hashlib
import datetime
import json

class Bloco:

    block_json = {}

    def __init__(self, index, data, nonce, previous_hash):
        self.index = index
        self.timestamp = str(datetime.datetime.now())
        self.data = data
        self.nonce = nonce
        self.previous_hash = previous_hash
        self.hash_code = self.hash_block()

    def hash_block(self):
        sha = hashlib.sha256()
        sha.update( str(self.nonce).encode('utf-8') +
                    str(self.data).encode('utf-8') +
                    str(self.previous_hash).encode('utf-8') +
                    str(self.index).encode('utf-8') +
                    str(self.timestamp).encode('utf-8')
                  )

        return "00000"+sha.hexdigest()

```

Características fundamentais

- **Descentralização:** Os acordos são feitos de maneira distribuída entre as partes
- **Disponibilidade e integridade:** Livro-razão replicado
- **Transparência e Auditabilidade:** Livro-razão público
- **Imutabilidade dos dados:** Devido ao ponteiro Hash o livro-razão é imutável
- **Anonimidade:** Realizar transações sem que terceiros tenham acesso

Fonte: (GREVE et al., 2018)

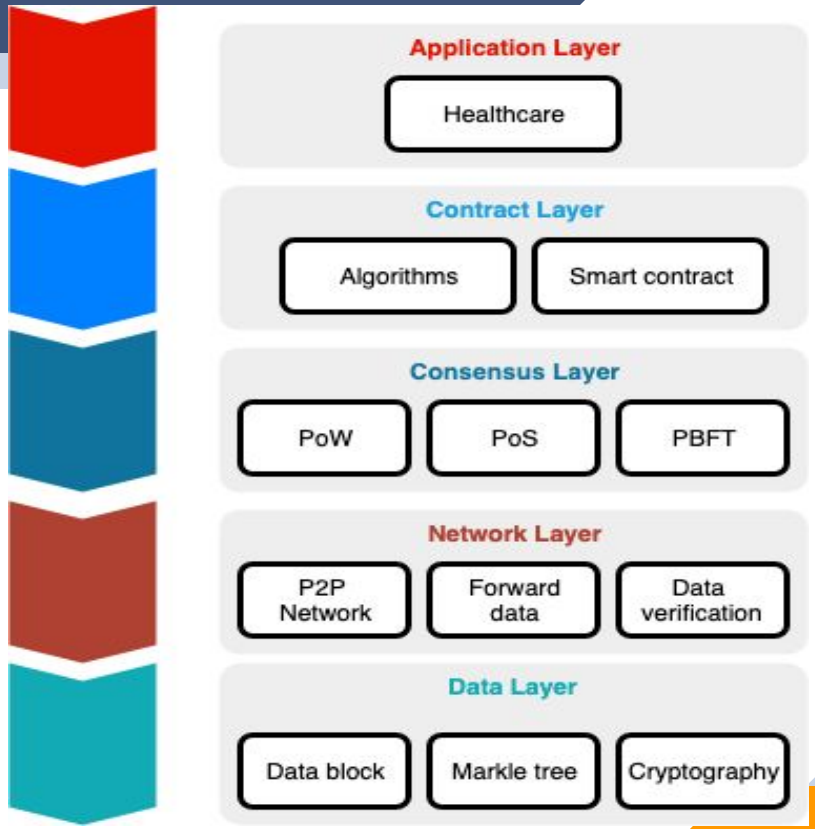
Características fundamentais

- **Desintermediação:** Sistemas realizando acordo de forma direta entre si, sem um terceiro confiável
- **Incentivo:** Modelo de negócios baseado no incentivo por validar uma transação

Fonte: (GREVE et al., 2018)

Camadas do blockchain

- Camada de aplicação
- Camada de contrato
- Camada de rede
- Camada de dados



Fonte: (YUAN; WANG, 2017)

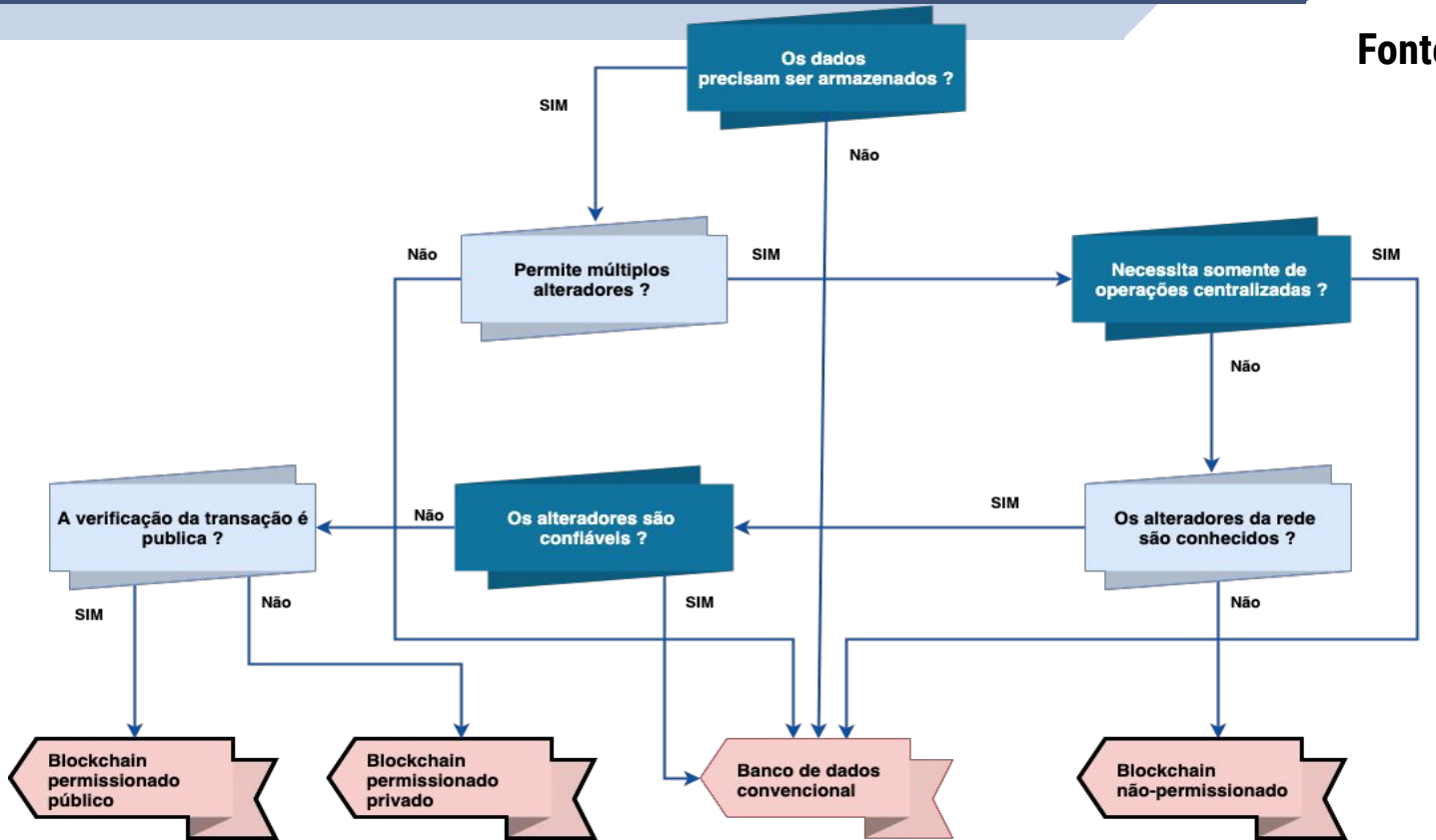
Tipos de redes blockchain

Descrição

Fonte: (WUST; GERVAIS, 2017)

- Podemos classificar as redes em três tipos:
  - **Permissionado público:** Nos específicos realizam o consenso
  - **Permissionado privado:** Um única organização controla a rede
  - **Não permissionado:** Qualquer nó pode fazer o consenso e ler o livro-razão

Fluxograma



Fonte: (WUST; GERVAIS, 2017)

Considerações finais

- Esse é um ponto que deve se considerar ao modelar uma rede blockchain
- Define a restrição de acesso aos dados no blockchain

Protocolos de consenso

Definição

- Estabelece um acordo entre as partes para determinar se um transação é válida
- Temos protocolos públicos e privados
- Bitcoin (público)
- Hyperledger Fabric (Privado)

Fonte: (RIFI et al., 2017)

Definição

- Para validar um transação deve-se:
  - Verificar as assinaturas
  - Verificar os valores da Hash do bloco (prevHash == Hash)
  - Confirmação se o valor não foi gasto por nenhuma outra transação

Fonte: (GREVE et. al., 2018)

Prova de trabalho

- *Proof of Work* (PoW)
- Proposto por Nakamoto em 2008/2009
- Base da criptomoeda Bitcoin
- Utiliza da concorrência entre nós para validar os blocos
- Resolve um enigma criptográfico
- Cada bloco necessita de 10 min para ser validado
- Requer grande custo computacional

Fonte: (NAKAMOTO, 2008)

Prova de trabalho

- Processo de validação:
  - O X envia um ativo para Y
  - A transação é propagada para a rede
  - Os mineradores competem para para validar o bloco
  - O primeiro a resolver a Hash cria um novo bloco com a transação de troca
  - O novo bloco é adicionado na rede
  - Os outros nós da rede compraram o novo bloco e o validam

Fonte: (NAKAMOTO, 2008)

Prova de participação

- *Proof of Stake* (PoS)
- Reduzir o custo de validação
- É baseado na posse da criptomoeda
- O participante que serão escolhidos como validadores mais confiáveis são:
  - Tem grandes quantidades de moeda
  - Passam mais tempo com as moedas

Fonte: (GREVE et. al., 2018)

Bizantino

- Baseado em tolerância a falhas Bizantinas práticas - PBFT
- Problemas dos generais Bizantinos
- Protocolo privado
- Baseado em votação para validar o bloco
- Os nós falhos < número total / 3

Fonte: (MINGXIAO et al., 2017)

Bizantino

- Validação:
  - É eleito um líder
  - Uma solicitação de entrada do cliente é enviada
  - A mensagem do cliente é pré-preparada e enviada para os nós validarem
  - Se o líder aceitar a mensagem é replicada para outros nós validarem
  - O líder aguarda a confirmação de 2*(f + 1) nós
  - A mensagem é validada e o líder adiciona no *ledger*

Fonte: (MINGXIAO et al., 2017)

Contratos inteligentes

Definição

- O termo contrato inteligente for definido por Nick Szabo
- Determina que uma transação eletrônica siga termos de um contrato
- Os contratos inteligentes tem o objetivo de satisfazer tarefas comuns
  - Comprar algo online
  - Trocar um ativo
- Reduzir o número de fraudes

Fonte: (GREVE et. al., 2018)

Ferramentas de implementação

- **Hyperledger Fabric:**
  - Privado
  - Provê escalabilidade
  - Utiliza o PBFT
  - Gerenciado pela IBM
  - Linguagem CTO, GO e JavaScript

https://composer-playground.mybluemix.net/editor

- **Ethereum:**
  - Público
  - Utilizado o PoS
  - Token GAS
  - Linguagem GO e Solidity

https://remix.ethereum.org

Hyperledger Fabric

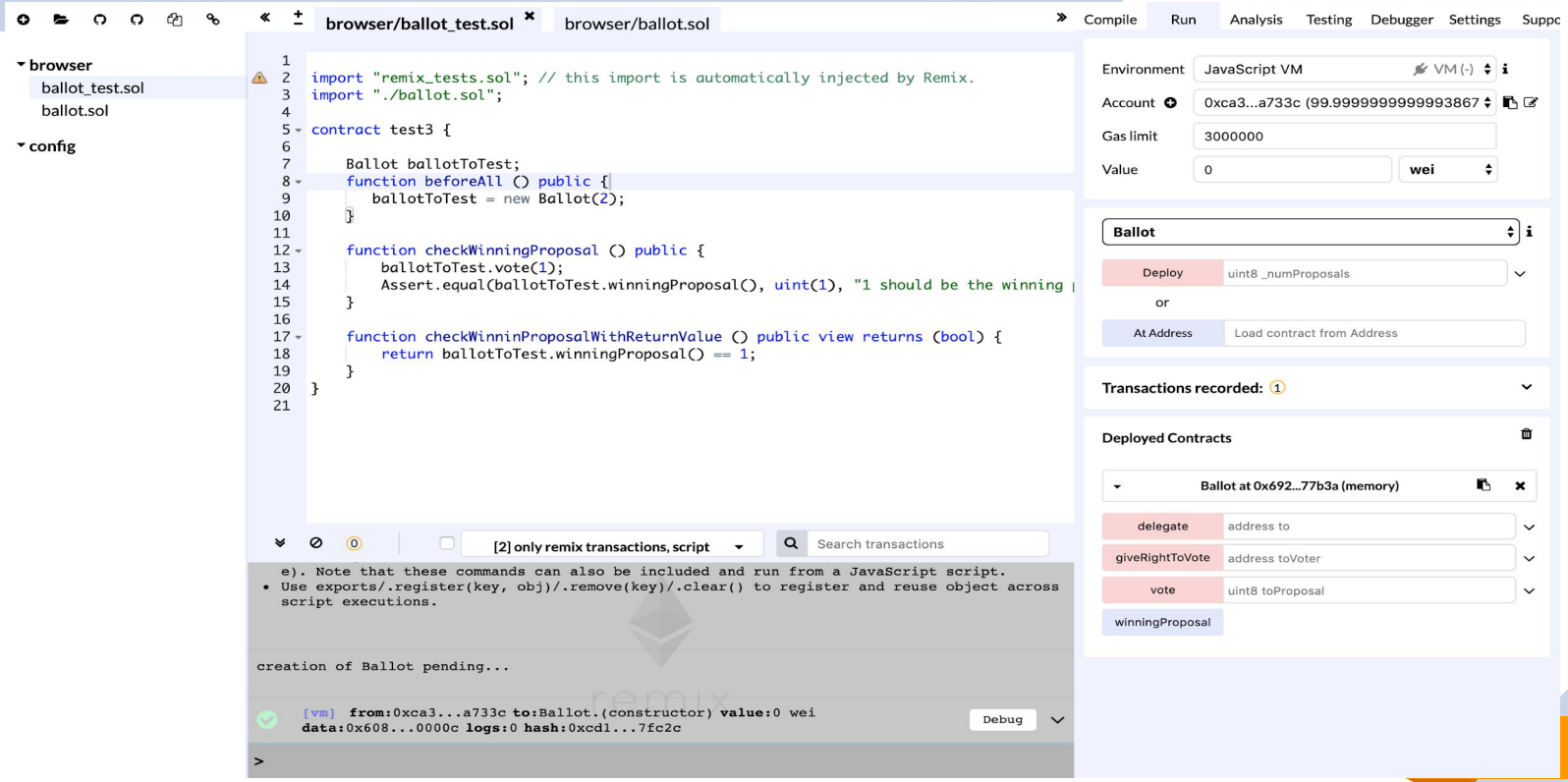
Script File lib/sample.js

```
async function sampleTransaction(tx) {  // eslint-disable-line no-unused-vars

    // Save the old value of the asset.
    const oldValue = tx.asset.value;

    // Update the asset with the new value.
    tx.asset.value = tx.newValue;

    // Get the asset registry for the asset.
    const assetRegistry = await getAssetRegistry('org.example.basic.SampleAsset');
    // Update the asset in the asset registry.
    await assetRegistry.update(tx.asset);

    // Emit an event for the modified asset.
    let event = getFactory().newEvent('org.example.basic', 'SampleEvent');
    event.asset = tx.asset;
    event.oldValue = oldValue;
    event.newValue = tx.newValue;
    emit(event);
}

```

Model File models/sample.cto

```
namespace org.example.basic

asset SampleAsset identified by assetId {
  o String assetId
  --> SampleParticipant owner
  o String value
}

participant SampleParticipant identified by participantId {
  o String participantId
  o String firstName
  o String lastName
}

transaction SampleTransaction {
  --> SampleAsset asset
  o String newValue
}

event SampleEvent {
  --> SampleAsset asset
```

Remix Solidity



Desafios do blockchain

Desafios

- Latência
- Throughput
- Largura de banda
- **Segurança**
- **Privacidade**
- Usabilidade
- Desperdício de recursos (Energia)

Fonte: (SWAN, 2015)

Segurança

- Gasto duplo
  - Saldo A = $ 100
  - TA -> A ⟶ B - Timestamp:2019-03/08-10:30:00 ($ 100)
  - TB -> A ⟶ C - Timestamp:2019-03/08 - 10:30:00 ($ 100)
  - Cadeia mais longa
- Perda da chave para acesso da carteira
- Falsificação das assinaturas

Fonte: (SWAN, 2015)

Privacidade

- Trata de dados pessoais do usuário
- Duas classes:
  - Privacidade de transação
  - Privacidade de identidade

Fonte: (FENG et al., 2019)

Técnicas de privacidade

- Criptografia homomórfica
- Prova de conhecimento zero
- Criptografia baseada em atributos
- Multi Multi-Party computation - SGX Intel
- Mixing

Fonte: (FENG et al., 2019)

Estudos de caso

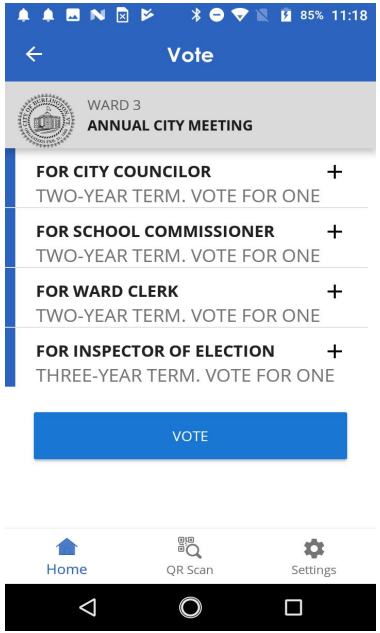
Financeiro

- BNDES Token
- Blockchain público
- Para registrar a distribuição de recursos
- Como empréstimos para instituições públicas
- API Web para acesso

Fonte: (ARANTES JR. et al., 2018)

Governamental

- Voted
- Aplicação baseada em blockchain para voto eletrônico
- Foi testado no Oeste da Virgínia - EUA
- Blockchain privado
- Eleitor tem um Id que referencia sua carteira
- Aplicativo é o Voatz



Saúde e IoT

- Monitoramento de pacientes utilizando IoT
- O dados são enviados para um Gateway
- O Gateway transfere as informações para servidor público em um rede blockchain
- Controle no acesso dos dados do paciente

Oportunidades de pesquisa

Oportunidades

- Melhorias na autonomia dos protocolos de consenso
- Garantir a privacidade dos usuários
- Tornar os contratos inteligentes autônomos (IA)
- Metodologias para modelagem de contratos

Oportunidades

- Melhoria na usabilidade das aplicações descentralizadas
- Prover escalabilidade as aplicações descentralizadas
- Seguir políticas de proteção a dados como a LGPD
- Melhor gerenciamento na distribuição das chaves

Considerações finais

Conclusões

- Blockchain é muito mais que só BTC
- É aplicado a qualquer área
- Está no início e ainda apresentando várias limitações
- Podemos construir Dapps

Perguntas ??

Obrigado!

Fundamentos e desafios do blockchain 3.0

Erikson J. de Aguiar

erjulioaguiar@usp.br

Intermídia