Desenvolvimento de dApps com Ethereum e Smart Contracts

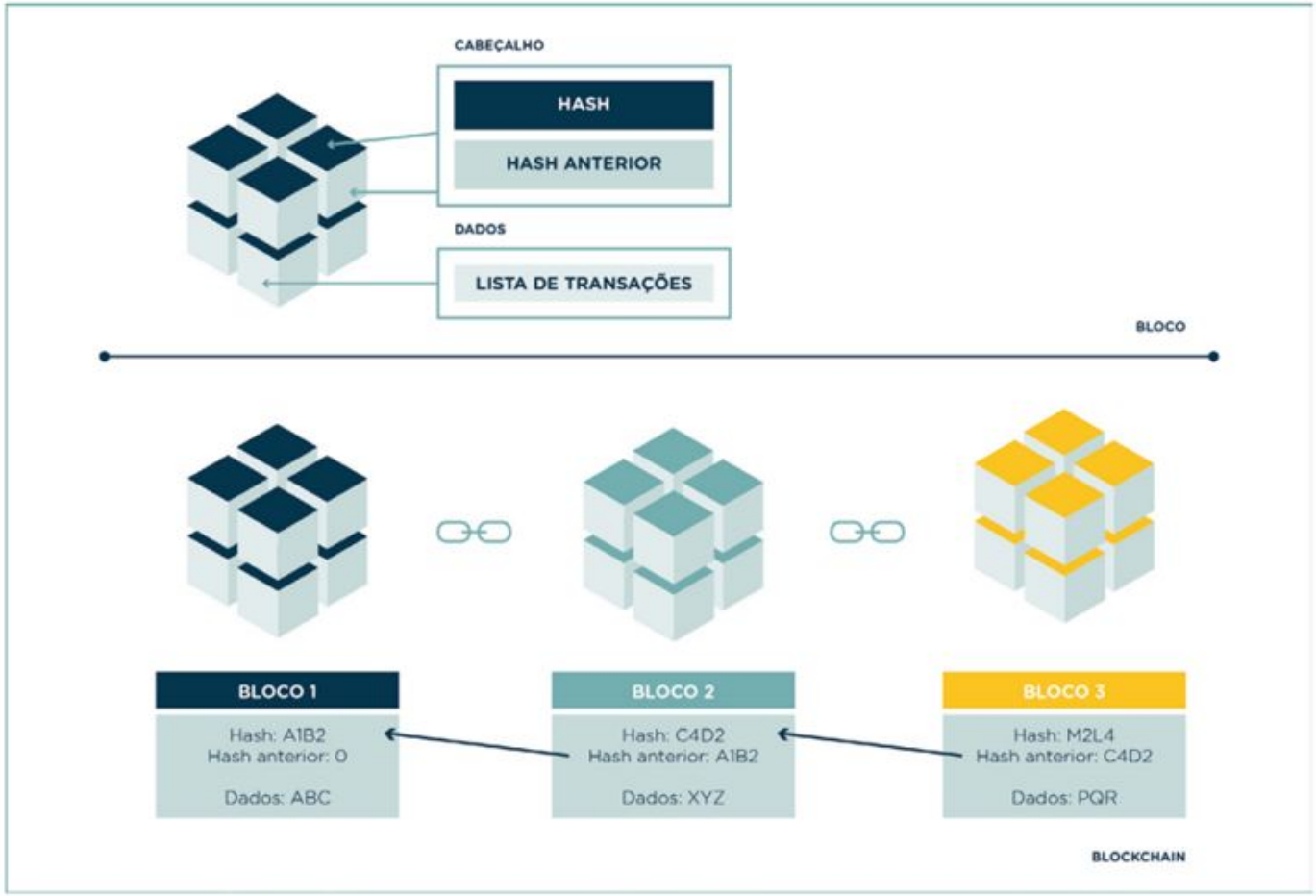
por Elvys Soares
elvys.soares@ifal.edu.br

Agenda

- Blockchain basics
- Ethereum basics
- Solidity basics
- Smart Contracts basics
- Criar um smart contract
- Testar um smart contract
- Fazer deploy de um smart contract

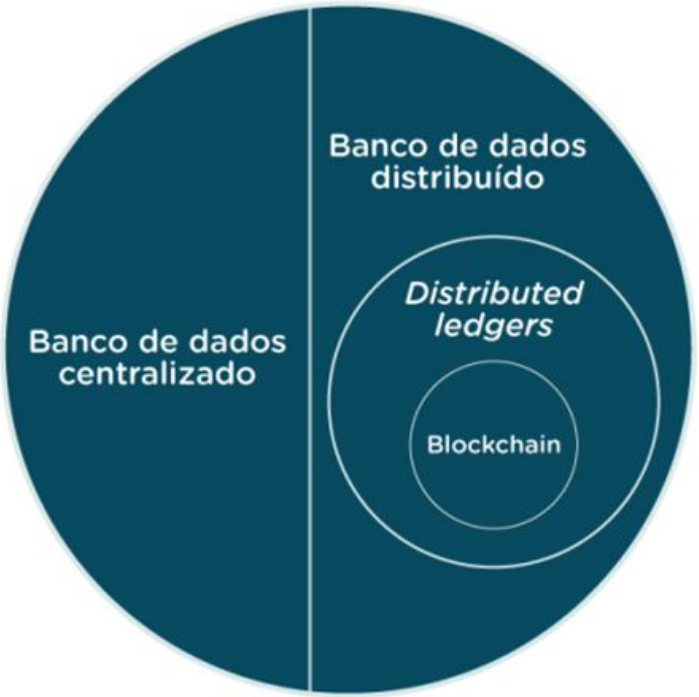
Blockchain basics

O que significa Blockchain?



Blockchain - Funcionamento genérico

Blockchain - Diferenças



Banco de dados distribuído

- não existe um "*master database*" central;
- provê um grau de tolerância a falhas caso alguns nós falhem;
- banco de dados tradicionais são, em geral, operados por uma entidade única que mantém um estrito controle de acesso para a rede;

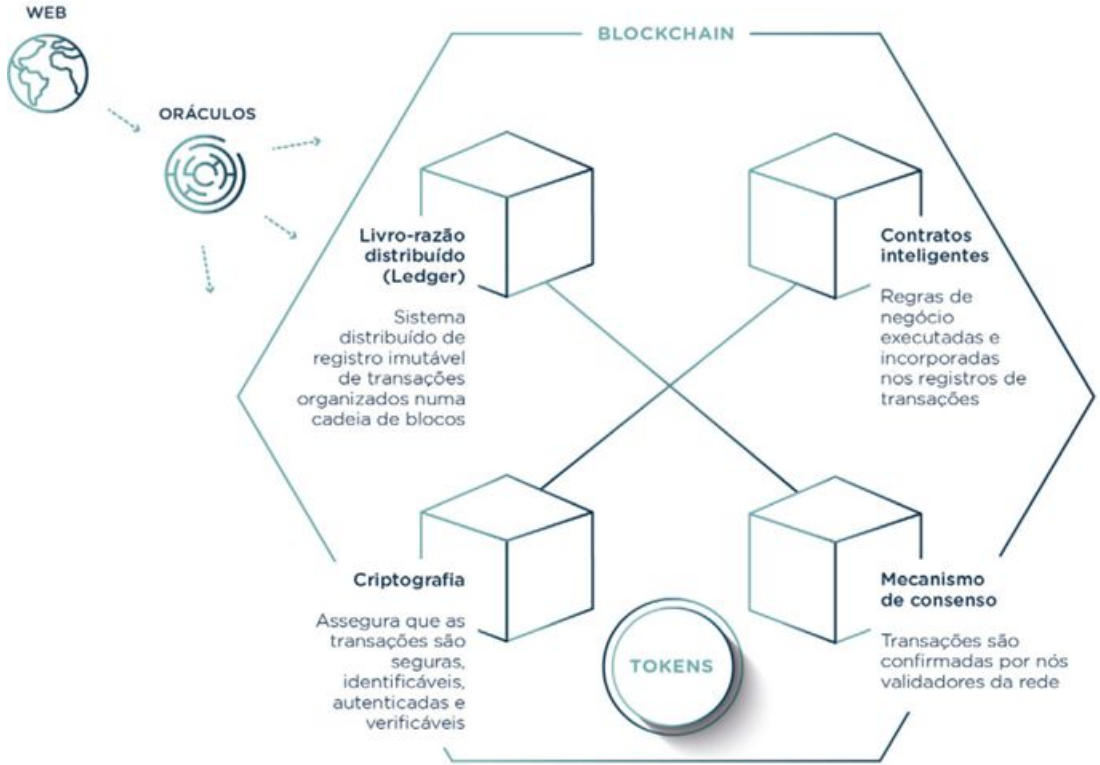
***Distributed Ledger Technology* (DLT)**

- mecanismo de consenso é baseado em um modelo de ameaças de adversários, assumindo que nem todos participantes são honestos;
- o banco de dados deve ser capaz de sincronizar e executar mesmo se um certo número de nós estão agindo de forma maliciosa;
- nós individuais precisam ser capazes de: **a)** verificar e validar, de forma independente, transações que alteram o estado do banco de dados e **b)** recriar todo o histórico de transações, de forma independente;

Blockchain

- usa uma estrutura de dados especial, *append-only*, que é composta por transações em lotes de blocos, os quais são ligados sequencialmente de forma inviolável, determinando a ordem das transações no sistema;

Blockchain - Componentes da Tecnologia



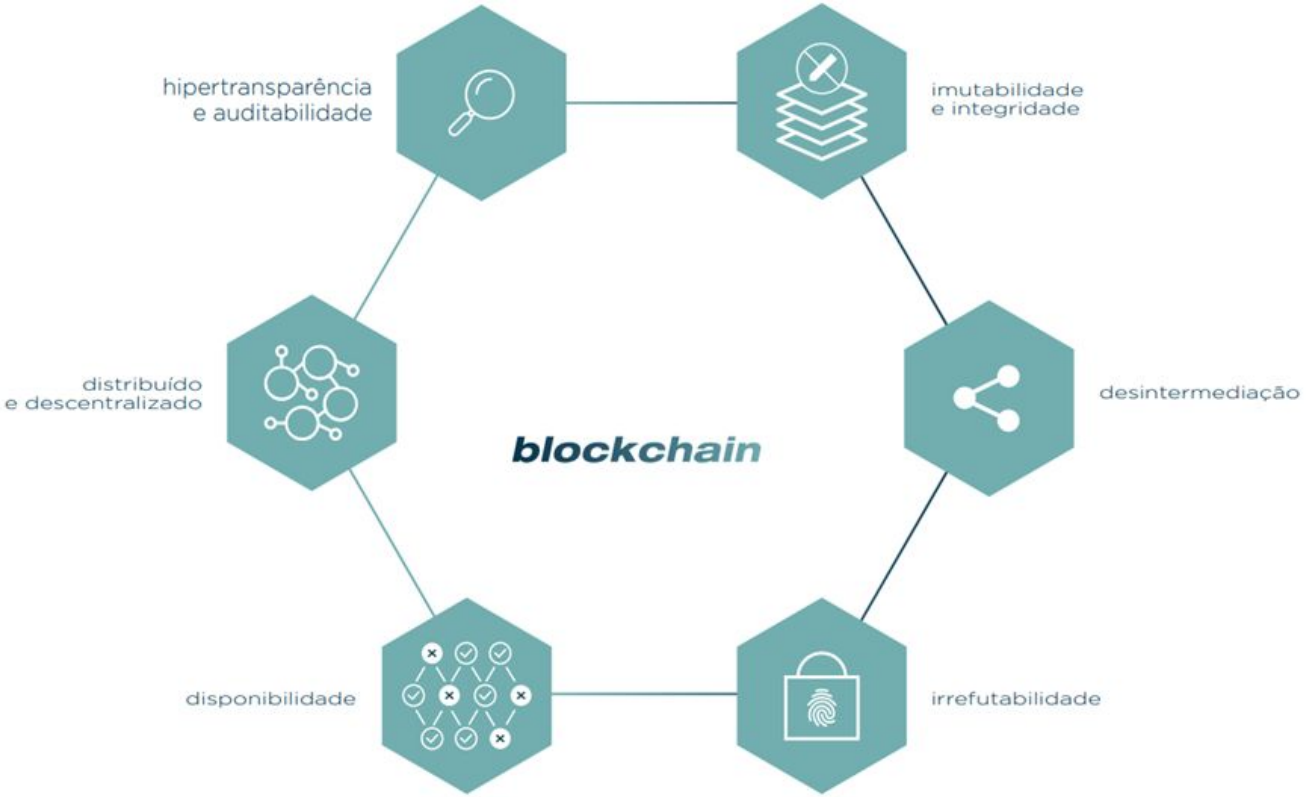
Blockchain - Contratos Inteligentes

- Caracterizados por quatro objetivos principais:
  - Observabilidade: todos cumpriram suas partes
  - Verificabilidade: o contrato foi cumprido e não violado;
  - Privacidade: conteúdo e execução privadas;
  - Obrigatoriedade: o contrato é executado de forma obrigatória, em sua completude, conforme programado em seu código-fonte, sem margem para interpretações diversas.

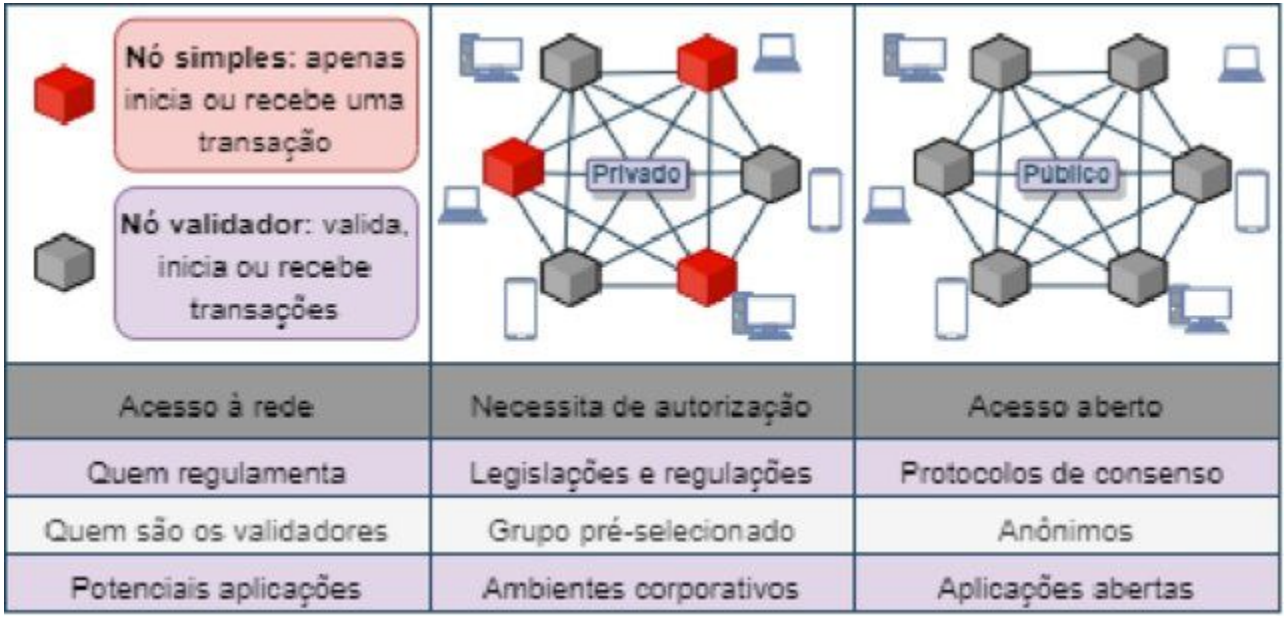
Blockchain - Contratos Inteligentes

- A utilização provê as seguintes vantagens:
  - Transparência;
  - Menor prazo para execução;
  - Precisão;
  - Segurança;
  - Rastreabilidade;
  - Menor custo;
  - Confiança.

Blockchain - Propriedades da Tecnologia



Blockchain - Tipos



Acesso à rede	Necessita de autorização	Acesso aberto
Quem regulamenta	Legislações e regulações	Protocolos de consenso
Quem são os validadores	Grupo pré-selecionado	Anônimos
Potenciais aplicações	Ambientes corporativos	Aplicações abertas

Ethereum Basics

O que é Ethereum?

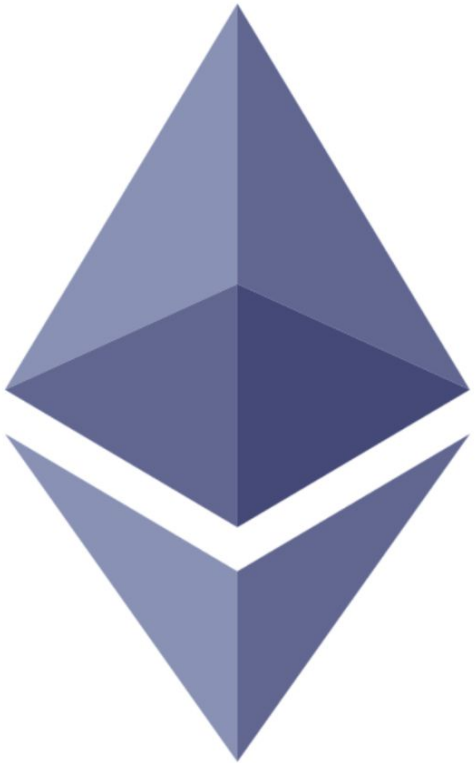
Ethereum é uma plataforma blockchain descentralizada que permite a criação de contratos inteligentes e aplicativos descentralizados (dApps) usando sua criptomoeda nativa, Ether (ETH).

Ethereum Basics

A definição visual:

- Imagine milhares de computadores em todo o mundo conectados pela Internet.
  - Cada um executa o mesmo programa de computador, sempre! Estes são os nós…
  - Este programa define regras (consenso!) sobre como os computadores devem trabalhar juntos
    - Como falar uns com os outros? Como armazenar dados? Como decidir o que é o quê...
- Em um sentido real, é um computador. **Este computador é Ethereum.**

Ethereum Basics



=

Propriedades do Computador Ethereum

1. Ethereum = Singleton Verdadeiro Global
2. Resistência à Censura
4. Ethereum = Nativamente Multi-Usuário
5. Ethereum é Verificável & Auditável

Hardhat

Hardhat é um ambiente de desenvolvimento para compilar, implantar, testar e depurar contratos inteligentes Ethereum.

Ele ajuda os desenvolvedores a gerenciar e automatizar as tarefas recorrentes inerentes ao processo de construção de contratos inteligentes e dApps, além de introduzir facilmente mais funcionalidades em torno desse fluxo de trabalho.

É a base para o nosso desenvolvimento em Ethereum!

Solidity

Here are some important properties of the Solidity language:

- **statically-typed** (fancy term meaning variables must be defined at compile time)
- **supports inheritance**: (specifically, smart contract inheritance chains)
- **libraries**
- **complex user-defined types**, among other features

Solidity is a programming language used to write smart contracts.

Solidity

- Linguagem usada para a escrita de smart contracts em Ethereum
- Similar ao Javascript, mas com algumas diferenças:
  - Estaticamente tipado (termo sofisticado que significa que as variáveis devem ser definidas em tempo de compilação)
  - Suporta herança: (especificamente, cadeias de herança de contratos inteligentes)
  - Bibliotecas
  - Tipos complexos definidos pelo usuário, entre outros recursos

Um contrato em Solidity

Solidity

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;

contract MyContract {
    constructor() {
        // called only ONCE on contract deployment
    }
}
```

Variáveis de estado

Solidity

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;

address owner;
bool isHappy;

contract MyContract {
    constructor(address _owner, bool _isHappy) {
        owner = _owner;
        isHappy = _isHappy;
    }
}
```

Tipos de dados em Solidity

- boolean: declarado como **bool**
- string: declarado como **string**
- integers: declarado como **uint** ou **int**
- bytes: declarado como **bytes**
- enums
- arrays
- mappings
- structs

Vamos à prática!

https://github.com/elvyssoares/minicurso

https://github.com/elvyssoares/hello-world-eth