



Reactor

S ã O P A U L O

Crie contratos inteligentes com o blockchain

Código do evento: #7755

Palestrante



Solange Gueiros

Blockchain developer

Speaker Bio: Especialista em Blockchain (arquitetura e desenvolvimento), com foco em Bitcoin, Ethereum e Smart Contracts, faz palestras, cursos e consultoria.

Trabalha com tecnologia há mais de 20 anos.

Palestrante em diversas conferências no Brasil e no mundo.

Ganhou prêmios nos hackathons de New York, Berlim e Denver.

Em 2020 está na lista top 50 do Cointelegraph Brazil.


<https://www.linkedin.com/in/solangegueiros/>

Blog: <https://solange.dev/>

Agenda

19:30	Blockchain fundamentals	Fundamentos de Blockchain
20:00	Blockopoly project: building the smart contracts	Projeto: Blockopoly: construindo os smart contracts
20:30	5-minute break	Intervalo de 5 minutos
20:35	Blockopoly project: deploy and interact with it	Projeto: Blockopoly: publicação e interação
21:00	Q&A	Perguntas e respostas
21:30	Event end	Fim do evento

Introdução

 meetup.com/Microsoft-Reactor-Sao-Paulo/



Motivação para a criação do Bitcoin

Criar um dinheiro digital

- descentralizado
- seguro
- aberto
- auto-regulado

O que foi inventado se tornou algo muito maior!



meetup.com/Microsoft-Reactor-Sao-Paulo/

Antes do Bitcoin

1991 - Stuart Haber e W. Scott Stornetta - Solução para documentos digitais

1993 - Cynthia Dwork e Moni Naor - proof of work (PoW)

1997 - Adam Back - hashcash

2004 - Hal Finney - RPoW – Reusable Proof of Work (Prova de Trabalho Reutilizável) - solucionou o problema do gasto duplo

Tentativas anteriores ao Bitcoin

1996 - e-gold

Gold & Silver Reserve Inc.

Era uma empresa

Emissão Centralizada

Lastro em Ouro

SEC não permitiu



 [meetup.com/Microsoft-Reactor-Sao-Paulo/](https://www.meetup.com/Microsoft-Reactor-Sao-Paulo/)

1998 – B-money

Wei Dai

Nunca foi lançado



1989 - digi.cash

David Chaum

Emissão centralizada

2005 – Bit gold

Nick Szabo

Proposta de uma moeda
digital descentralizada



<https://bitcoin.org/bitcoin.pdf>

Uma versão puramente ponto-a-ponto de dinheiro eletrônico permitiria que pagamentos on-line fossem enviados diretamente de uma parte para outra, sem passar por uma instituição financeira.

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

Bitcoin - origem

White paper em out/2008

No grupo de discussão The Cryptography Mailing

Pelo usuário de pseudônimo Satoshi Nakamoto

Será um programador ou um grupo de programadores?

Código fonte – rede no ar em jan/2009

Bitcoin - origem

Resolve os problemas de

- centralização
- gasto duplo
- identidade
- emissão de moedas
- consenso



Permite pagamento peer to peer (p2p) - de usuário a usuário – sem intermediários!

O que é Bitcoin?

- Um protocolo que define uma rede ponto a ponto descentralizada
- Um livro razão público com transações
- Um conjunto de regras para validar transações
- Um mecanismo para conseguir consenso
- Um sistema econômico que recompensa a participação



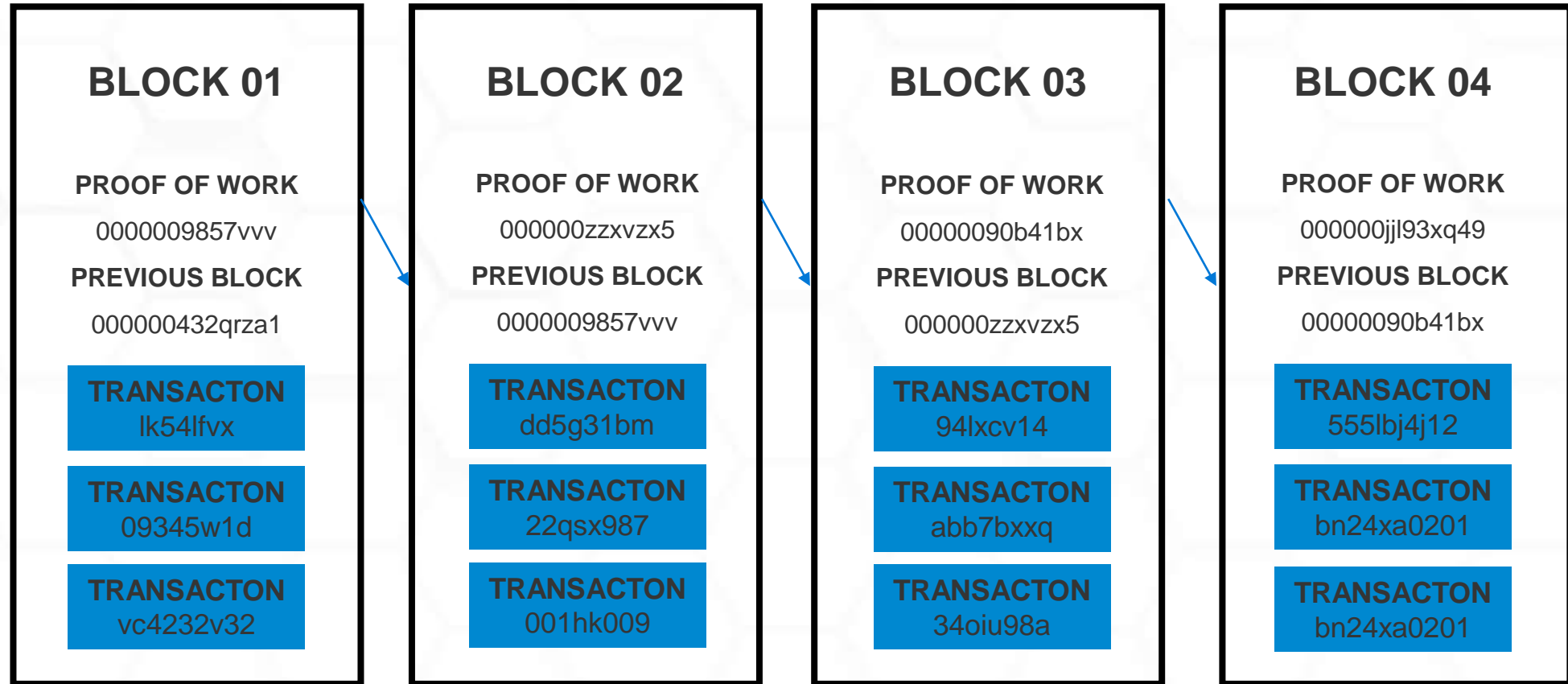
Bitcoin is not “unregulated”. It is regulated by algorithm instead of being regulated by government bureaucracies. Un-corrupted.

— *Andreas Antonopoulos* —

AZ QUOTES

Bitcoin não é “sem regras”. Ele é regulado por um algoritmo ao invés de burocracias de governo. “Incorruptível”.

Blockchain = cadeia de blocos



Transações e máquina de estado

Estado 1

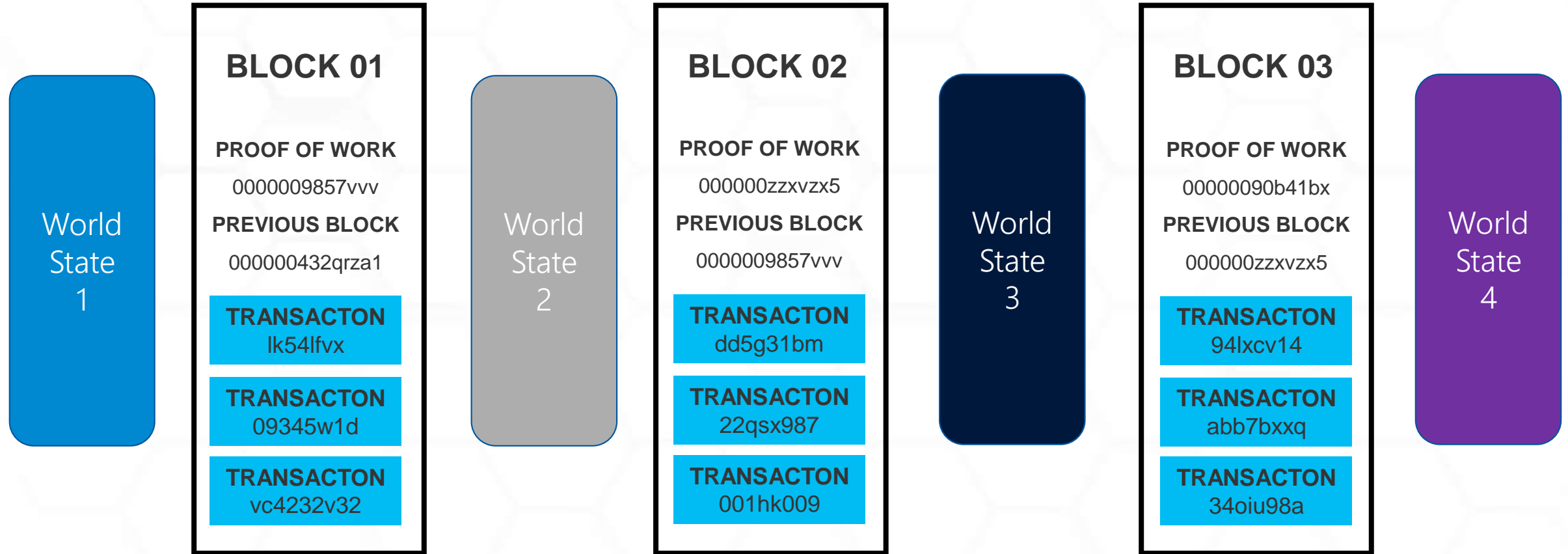
- Saldo 10
- Info A

Transação

Estado 2

- Saldo 5
- Info B

Estados alterados no Blockchain



O que é Blockchain?

Arquitetura - conjunto de tecnologias

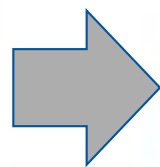
Sistemas distribuídos

Criptografia

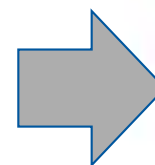
Teoria dos Jogos



Centralizado



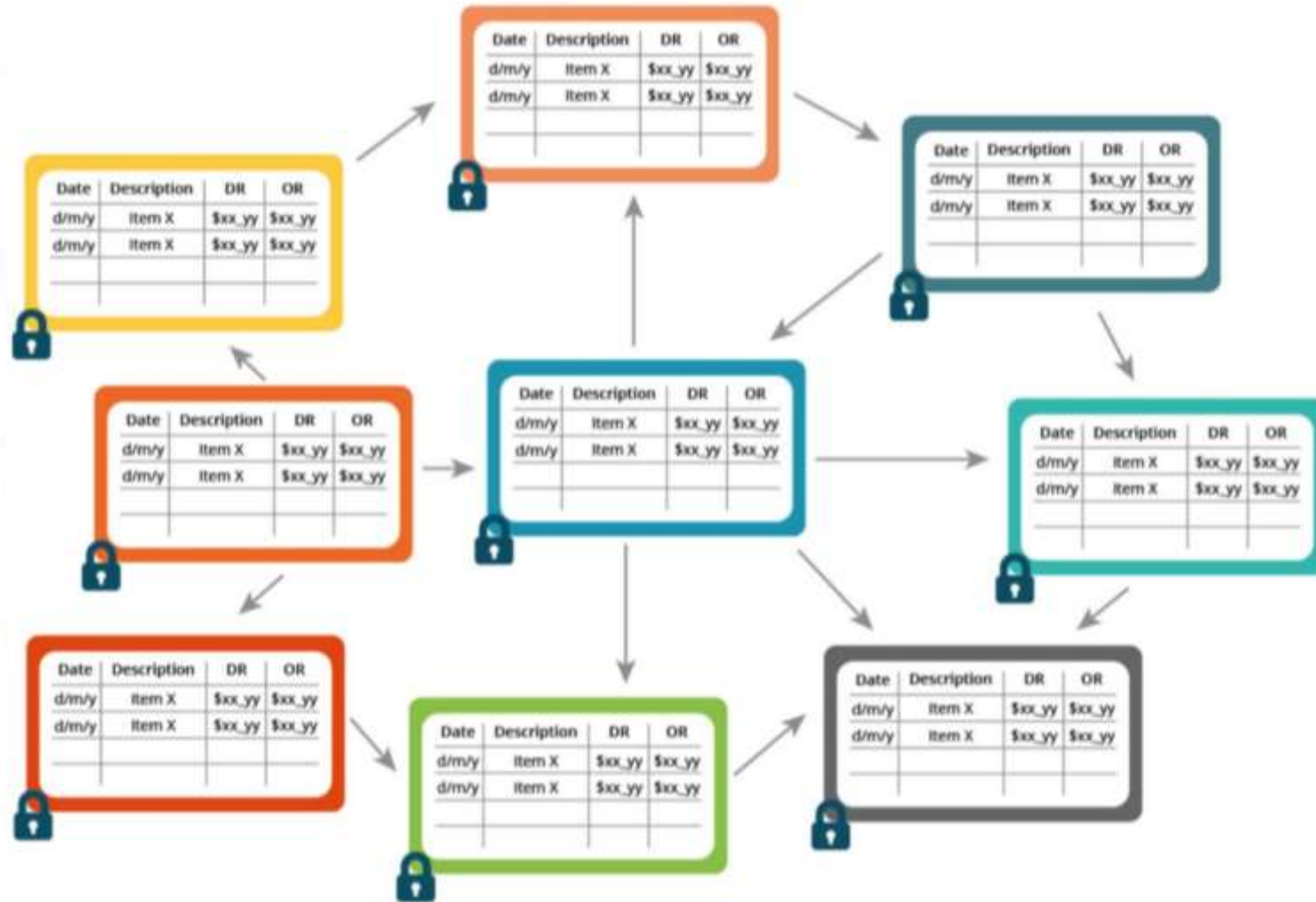
Descentralizado



Distribuído

BLOCKCHAIN

Blockchain = livro razão distribuído



Criptografia

Hash

Chave pública / privada

Merkle tree

Teoria dos Jogos

Algoritmos de consenso


Proof of Work (PoW) = Prova de trabalho

Proof of Stake (PoS) = Prova de participação

Mineração

- Disputa para encontrar um hash de um bloco menor que o nível de dificuldade atual antes dos outros
- Segurança e validação do blockchain
- Recompensa:
 - O bloco deve conter incluir uma transação Coinbase

Ethereum

 meetup.com/Microsoft-Reactor-Sao-Paulo/



Ethereum White Paper

A NEXT GENERATION SMART CONTRACT & DECENTRALIZED APPLICATION PLATFORM

By Vitalik Buterin

When Satoshi Nakamoto first set the Bitcoin blockchain into motion in January 2009, he was simultaneously introducing two radical and untested concepts. The first is the "bitcoin", a decentralized peer-to-peer online currency that maintains a value without any backing, intrinsic value or central issuer. So far, the "bitcoin" as a currency unit has taken up the bulk of the public attention, both in terms of the political aspects of a currency without a central bank and its extreme upward and downward volatility in price. However, there is also another, equally important, part to Satoshi's grand experiment: the concept of a proof of work-based blockchain to allow for public agreement on the order of transactions. Bitcoin as an application can be described as a first-to-file system: if one entity has 50 BTC and simultaneously sends the same 50 BTC to



Ethereum Launches

Posted by Stephan Tual on July 30, 2015

A few minutes ago, many of you generated and loaded the Ethereum Genesis block, marking the inception of Frontier, the first Live release of the Ethereum project.

Ethereum

- Blockchain público
- Totalmente descentralizado
- PoW / PoS
- Máquina virtual 'Turing-complete'
- Determinístico
- Execução de smart contracts
- Criação de aplicativos distribuídos



Ethereum

Blockchain
Infraestrutura
Tecnologia
Transações

Ether

Criptomoeda
Menor unidade: wei
 10^{-18}

1 ether =
1.000.000.000.000.000.000 wei



Gas

Gas (quantidade)

X

Gas price (Gwei)

Wei é ether, gas é ether

1 Gwei = 10000000000 (1 bilhão)

<https://ethgasstation.info/>

<https://eth-converter.com/>

Smart contracts

Nick Szabo

- Cientista da computação e criptógrafo Nick Szabo
- 1993 / 1994
- Publicação: 1996
- Na época não havia um ambiente apropriado para executar os smart contracts.

Imagem:
<https://bitconnect.co/bitcoin-news/800/nick-szabo-developed-a-method-of-sending-bitcoin-transactions-over-radio>

 meetup.com/Microsoft-Reactor-Sao-Paulo/



SMART CONTRACTS

Present

Smart Contracts: Building Blocks for Digital Markets
Copyright (c) 1996 by Nick Szabo

Past

permission to redistribute without alteration hereby granted

Subjects

Smart Contracts Glossary

Projects

Misc

(This is a partial rewrite of the article which appeared in Extropy #16)

Introduction

The contract, a set of promises agreed to in a "meeting of the minds", is the traditional way to formalize a relationship. While contracts are primarily used in business relationships (the focus of this article), they can also involve personal relationships such

Vending machine

Ancestral do Smart Contract

Imagem:
<http://www.intelligentvending.co.uk/combi-vending-machines-snacks-drinks/g-snack-smx-vending-machine-elevator-fridge.htm>

 [meetup.com/Microsoft-Reactor-Sao-Paulo/](https://www.meetup.com/Microsoft-Reactor-Sao-Paulo/)



Smart contracts no Ethereum


- Turing-complete
- Os contratos são compilados para a máquina virtual do Ethereum - EVM e em seguida gravados no blockchain
- Programa de computador auto executável



Imutabilidade

- Não pode corrigir o código!
- O smart contract pode ter funções para alterar dados.
- Não pode alterar o histórico:
 - A informação pode ser registrada em um bloco
 - E pode ser apagada em outro
 - Fica o histórico: auditoria!

Solidity

 meetup.com/Microsoft-Reactor-Sao-Paulo/

Fundamentos de Solidity

- Variáveis de estado
- Funções
- Modificadores
- Eventos
- Tipos de dados: Struct
- Tipos de dados: Enum
- Mappings



Reactors pelo mundo



 meetup.com/Microsoft-Reactor-Sao-Paulo/

Perfis e comportamentos

- Que tipo de perfis ou papéis precisamos?
- Quais ações existem no jogo?
- O que é transferido?
- Como identificamos jogadores?
- Em quem confiamos?

Pré-requisitos

- Node.js e NPM (Node Package Manager)
- <https://nodejs.org/en/>
- Visual Studio Code (VSCode) ou outro editor à sua escolha
- <https://code.visualstudio.com/>
- Truffle framework
- <https://www.trufflesuite.com/truffle>


Arquitetura - Smart contracts

- Bank
- AssetManager
- Blockopoly

Bank

- Uma variável pública `minter` para saber quem é o banqueiro, o emissor de dinheiro.
- Um mapping (variável do tipo chave -> valor) chamada `balances`, privada, que armazena o saldo de cada endereço.
- Um evento `Sent` que avisa cada vez que alguém envia dinheiro para outra pessoa.
- O `constructor` (construtor), executado apenas no momento da criação do banco, define quem é o `minter`.
- Uma função `mint` que faz a emissão de dinheiro.
- Uma função `sendMoney` para enviar dinheiro de um endereço para outro.
- Uma função `getBalance` que retorna o saldo de um determinado endereço.

Intervalo

 meetup.com/Microsoft-Reactor-Sao-Paulo/

AssetManager

- Uma variável pública ``manager`` para saber qual o endereço do administrador de ativos.
- Uma estrutura chamada ``Asset``, que armazena todas as informações de um ativo: proprietário, nome do ativo, preço e se ele existe.
- Um mapping chamado ``assets``, privado, que armazena um identificador associando-o a uma estrutura com as informações de um ativo.
- Um evento ``AssetAdded`` que avisa quando algum ativo for adicionado no jogo.
- Um evento ``AssetTransfer`` que avisa quando algum ativo for transferido de um endereço a outro no jogo.
- O ``constructor`` (construtor), executado apenas no momento da criação do smart contract, define quem é o ``manager``.
- Uma função ``addAsset`` que adiciona os ativos no jogo.
- Uma função ``getOwner`` que retorna quem é o proprietário de um ativo.
- Uma função ``transferAsset`` para transferir um ativo de uma pessoa para outra.

Blockopoly – variáveis

- Uma variável pública `banker` que define quem é o banqueiro do jogo;
- Uma variável pública `bank`, que é o smart contract banco, importado;
- Uma variável pública `assetManager`, que é o smart contract gerenciador de ativos, também importado;
- Uma estrutura chamada `Player`, que armazena todas as informações de um jogador: endereço (account) e nome;
- Um array `players` que é a lista pública de jogadores;
- Um mapping chamado `names`, público, que controla se um nome já está sendo utilizado no jogo;
- Um mapping chamado `addrPlayerMapping`, público, que faz a associação entre o endereço / conta de uma pessoa e o jogador que ela está utilizando;
- Uma variável pública `started` que define se a partida começou;
- Uma variável pública `endTime` que define quando a partida termina;

Blockopoly – eventos e construtor

Eventos

- Um evento `GameStarted` que avisa quando o jogo começou;
- Um evento `PlayerJoined` que avisa quando um jogador entrou no jogo;

Construtor


- define quem é o banqueiro
- cria o banco
- faz a emissão inicial de moedas da partida
- cria o gerenciador de ativos
- chama a função que cria as propriedades disponíveis na partida.

Blockopoly – funções

- A função `publishProperties` cria as propriedades disponíveis na partida. Esta função é privada, então só pode ser chamada por um smart contract. Ela é invocada pelo construtor;
- A função `startGame` inicia o jogo;
- A função `joinGame` possibilita que uma pessoa entre na partida;
- A função `buyProperty` é para alguém comprar uma propriedade.
- `getWinner` informa quem é o vencedor da partida.



Vamos jogar!

 meetup.com/Microsoft-Reactor-Sao-Paulo/

Regras do jogo

1. O banqueiro publica o jogo.
2. Os jogadores entram
3. Inicia a partida
4. Os jogadores alternam sua vez para jogar
5. Na sua vez, o jogador escolhe uma propriedade e compra
6. O final do jogo acontece depois de 15 minutos.
7. Quem tiver mais dinheiro no final do jogo ganha.
8. Se dois jogadores possuírem o mesmo saldo, o critério de desempate é quem se cadastrou primeiro na partida.

Perguntas e Respostas



Reactor

S Ã O P A U L O

Estamos constantemente nos esforçando para criar excelentes conteúdos e agradeceríamos se você pudesse responder esta rápida pesquisa.

Link pesquisa: <https://aka.ms/Reactor/Survey>

Use o código do evento 7755 no início da pesquisa.

Próximos passos

- Atualmente a mesma conta / endereço pode se cadastrar 2x
- Listar propriedades disponíveis
- Ordem dos jogadores
 - Implementar o rodízio no smart contract futuramente.
 - Ou
 - Sorteio do próximo jogador
- Final do jogo
- Fazer uma interface, frontend

Material

- <https://github.com/microsoft/ReactorSaoPaulo/tree/main/Workshops/Blockchain>

Junte-se a nós



[meetup.com/Microsoft-Reactor-Sao-Paulo/](https://www.meetup.com/Microsoft-Reactor-Sao-Paulo/)



@MSFTReactor



<http://www.youtube.com/c/MicrosoftReactor>



reactorsaopaulo@microsoft.com



[meetup.com/Microsoft-Reactor-Sao-Paulo/](https://www.meetup.com/Microsoft-Reactor-Sao-Paulo/)



Microsoft Reactor at Distrito AdTech Hub,
Rua Estados Unidos, 1570, Sao Paulo,
Sao Paulo 01412-100

Questions? reactorsaopaulo@microsoft.com



Reactor

S Ã O P A U L O

Muito obrigada!



meetup.com/Microsoft-Reactor-Sao-Paulo/



[@MSFTReactor](https://twitter.com/MSFTReactor)



<http://www.youtube.com/c/MicrosoftReactor>



aka.ms/ReactorEmailSignUp



meetup.com/Microsoft-Reactor-Sao-Paulo/