Go Blockchain

BLOCKCHAINBH

ethereum

Em 2019: Uma Análise

Daniel Martins
Henrique Leite
Yakko Majuri
Alex Braz
Solange Gueiros

Temas

Novidades e notícias

Novas versões frameworks

Solidity 0.5

Eventos de 2018

Demanda Brasil e mundo profissionais Blockchain

Análise arquitetura Decentralized Organization

Desenvolvimento de uma aplicação simples

Novidades e notícias

Ethereum Classic sofre ataque de 51% - Links: 1, 2, 3, 4

Vulnerabilidade adia fork do Ethereum

Um olhar sobre 2018: o ano do Enterprise Ethereum - 1, 2, 3

ConsenSys news

Ethereum Upgrade

0	Olympic	2015	
1	Frontier	2015	
2	Homestead	2016	primeira ESTÁVEL
3	Metropolis	2017	Byzantium
3.5	Metropolis	2019	Constantinople
4	Serenity	?	Proof-Of-Stake finalmente

Otimizações!

EIP 145	Bitwise shifting instructions on the EVM
EIP 1014	State Channels: transactions offchain
EIP 1052	Verification of smart contracts
EIP 1283	Reduce Gas for some storage operations
EIP 1234	3 ETH to 2 ETH reward. Preparation for Casper

Novas versões Frameworks

Consensys: Developer Tools List

Solidity 0.5

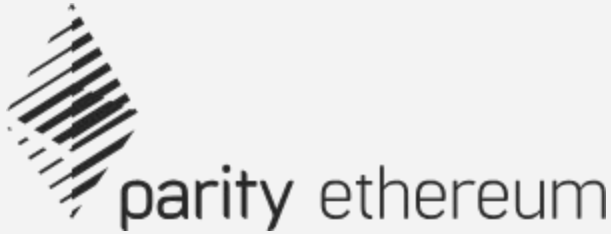
Truffle 5.0.2

MetaMask 5.3.0

Web3.js 1.0.0

Geth 1.8.21

Parity Ethereum



Solidity 0.5

https://solidity.readthedocs.io/

Novidades

Mais segurança

Restrição de formas na qual a linguagem pode ser usada

Remoção de ambiguidades, comportamento estranho

Requer ao programador ser mais específico

Visibilidade, construtor e eventos

```
// 0.4.0
contract MyContract {

    address owner;

    event Withdrawn();

    function MyContract() {
        initialize();
    }

    function initialize() {
        owner = msg.sender;
    }

    function withdraw() {
        require(msg.sender == owner);
        msg.sender.transfer(address(this).balance);
        Withdrawn();
    }
}
```

Visibilidade, construtor e eventos

```
// 0.5.0 (halfway)
contract MyContract {

    address owner;

    event Withdrawn();

    function MyContract() public {
        initialize();
    }

    function initialize() internal {
        owner = msg.sender;
    }

    function withdraw() public {
        require(msg.sender == owner);
        msg.sender.transfer(address(this).balance);
        Withdrawn();
    }
}
```

Visibilidade, construtor e eventos

```
// 0.5.0 (almost)
contract MyContract {

    address owner;

    event Withdrawn();

    constructor() public {
        initialize();
    }

    function initialize() internal {
        owner = msg.sender;
    }

    function withdraw() public {
        require(msg.sender == owner);
        msg.sender.transfer(address(this).balance);
        Withdrawn();
    }
}
```

Visibilidade, construtor e eventos

```
// 0.5.0
contract MyContract {

    address owner;

    event Withdrawn();

    constructor() public {
        initialize();
    }

    function initialize() internal {
        owner = msg.sender;
    }

    function withdraw() public {
        require(msg.sender == owner);
        msg.sender.transfer(address(this).balance);
        emit Withdrawn();
    }
}
```

Tipos explícitos

```
// 0.4.0
contract MyContract {

    function sum(uint[] a) public pure returns(uint s) {
        for(var i = 0; i < a.length; i++) {
            s += a[i];
        }
    }

}
```

Tipos explícitos

```
// 0.5.0 (halfway)
contract MyContract {

    function sum(uint[] a) public pure returns(uint s) {
        for(uint i = 0; i < a.length; i++) {
            s += a[i];
        }
    }

}
```

Tipos explícitos

```
// 0.5.0
contract MyContract {

    function sum(uint[] memory a) public pure returns(uint s) {
        for(uint i = 0; i < a.length; i++) {
            s += a[i];
        }
    }

}
```

Ponteiros storage devem ser inicializados

```
// 0.4.0
contract MyContract {

    struct Data { string name; uint amount; }
    Data[] participants;

    function participate(string _name) public {
        Data p;
        p.name = _name;
        p.amount = msg.value;
        participants.push(p);
    }

}
```

Ponteiros storage devem ser inicializados

```
// 0.5.0
contract MyContract {

    struct Data { string name; uint amount; }
    Data[] participants;

    function participate(string memory _name) public {
        participants.length += 1;
        Data storage p = participants[participants.length - 1];
        p.name = _name;
        p.amount = msg.value;
    }

}
```

Referências

https://solidity.readthedocs.io/

Apresentação na Dappcon

Eventos de 2018

DevCon 4

TruffleCon

LaBITConf

BitConf

Blockchain Festival

Demanda por profissionais Blockchain

Análise: Decentralized Organization

Go Blockchain

https://goblockchain.io

– *СПАСИБО!*