IMD0913

SMART CONTRACTS COM SOLIDITY

Solidity

Smart Contracts são normalmente escritos em uma linguagem de alto nível, como o **Solidity**

Depois de compilada, é realizado o *deploy* na plataforma Ethereum usando uma transação especial de **criação de contrato**

> E não podem ser mais modificado (imutável), mas pode ser deletado (se usar opcode SELFDESTRUCT)

Cada contrato é identificado por um endereço, mas não tem chaves privadas associadas (ou seja, não tem dono)

Compilador Solidity

Exemplo

Faucet.sol

```
pragma solidity ^0.5.10;
// Nosso primeiro contrato é um Faucet!
contract Faucet {

    // Envia ether para qualquer um que solicitar
    function withdraw(uint withdraw_amount) public {

        // Limita a quantidade de transferência
        require(withdraw_amount <= 100000000000000000);

        // Envia a quantidade para o endereço que requisitou
        msg.sender.transfer(withdraw_amount);
    }

    // Aceita qualquer quantidade enviada para o Faucet
    function () external payable {}

}
```

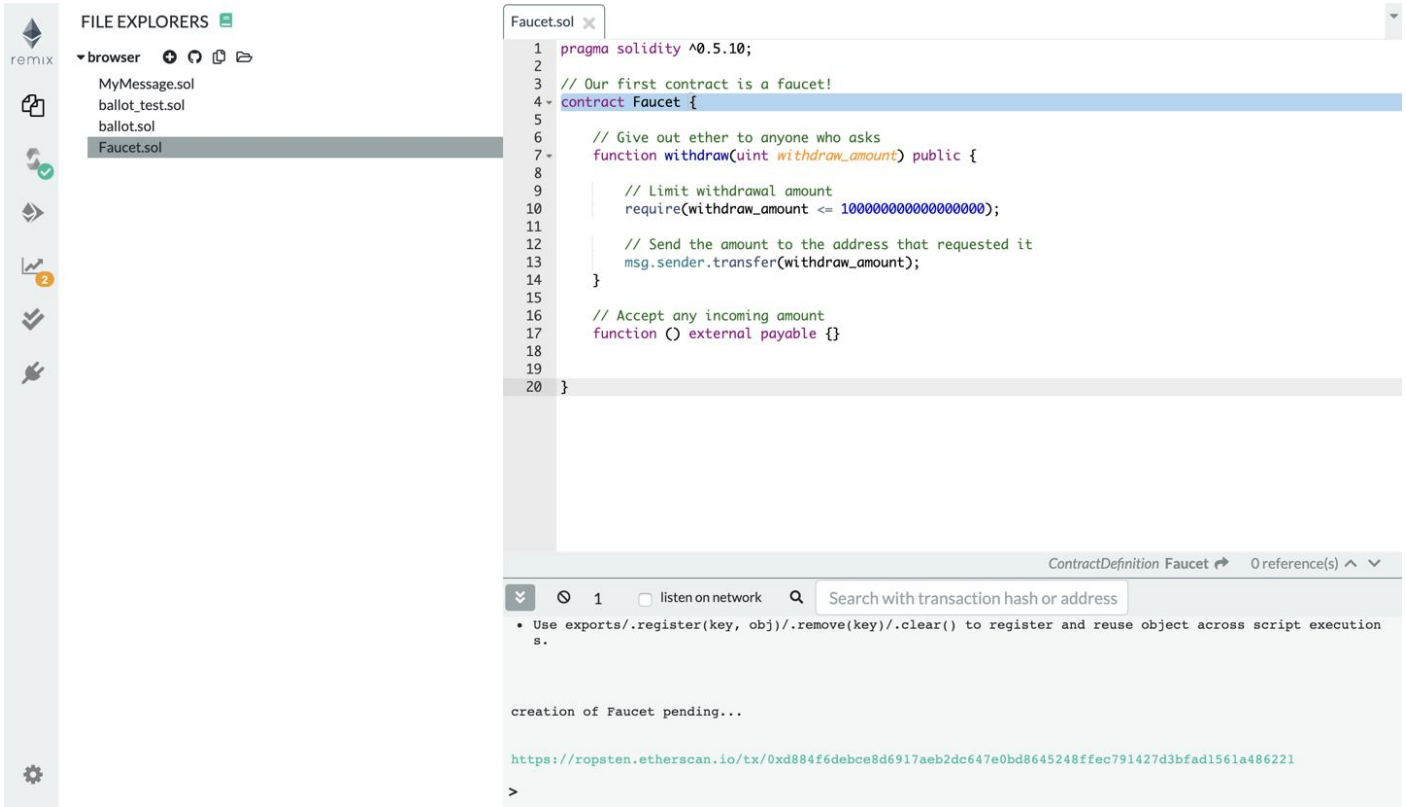
Exemplo

```
PUSH1 0x80 PUSH1 0x40 MSTORE CALLVALUE DUP1 ISZERO PUSH2 0x10 JUMPI PUSH1 0x0 DUP1
REVERT JUMPDEST POP PUSH1 0xE7 DUP1 PUSH2 0x1F PUSH1 0x0 CODECOPY PUSH1 0x0 RETURN
INVALID PUSH1 0x80 PUSH1 0x40 MSTORE PUSH1 0x4 CALLDATASIZE LT PUSH1 0x1C JUMPI PUSH1
0x0 CALLDATALOAD PUSH1 0xE0 SHR DUP1 PUSH4 0x2E1A7D4D EQ PUSH1 0x1E JUMPI JUMPDEST
STOP JUMPDEST CALLVALUE DUP1 ISZERO PUSH1 0x29 JUMPI PUSH1 0x0 DUP1 REVERT JUMPDEST
POP PUSH1 0x53 PUSH1 0x4 DUP1 CALLDATASIZE SUB PUSH1 0x20 DUP2 LT ISZERO PUSH1 0x3E
JUMPI PUSH1 0x0 DUP1 REVERT JUMPDEST DUP2 ADD SWAP1 DUP1 DUP1 CALLDATALOAD SWAP1
PUSH1 0x20 ADD SWAP1 SWAP3 SWAP2 SWAP1 POP POP POP PUSH1 0x55 JUMP JUMPDEST STOP
JUMPDEST PUSH8 0x16345785D8A0000 DUP2 GT ISZERO PUSH1 0x69 JUMPI PUSH1 0x0 DUP1
REVERT JUMPDEST CALLER PUSH20 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF AND PUSH2
0x8FC DUP3 SWAP1 DUP2 ISZERO MUL SWAP1 PUSH1 0x40 MLOAD PUSH1 0x0 PUSH1 0x40 MLOAD
DUP1 DUP4 SUB DUP2 DUP6 DUP9 DUP9 CALL SWAP4 POP POP POP POP ISZERO DUP1 ISZERO PUSH1
0xAE JUMPI RETURNDATASIZE PUSH1 0x0 DUP1 RETURNDATACOPY RETURNDATASIZE PUSH1 0x0
REVERT JUMPDEST POP POP JUMP INVALID LOG2 PUSH6 0x627A7A723058 KECCAK256 DUP12 SDIV
DUP4 0x2d 0xc5 ADDMOD MOD BYTE 0xa7 PUSH13 0xF3B7C3499097383D5B9E9C83BD PUSH24
0x7CE0352811C9D3AC64736F6C634300050A00320000000000
```

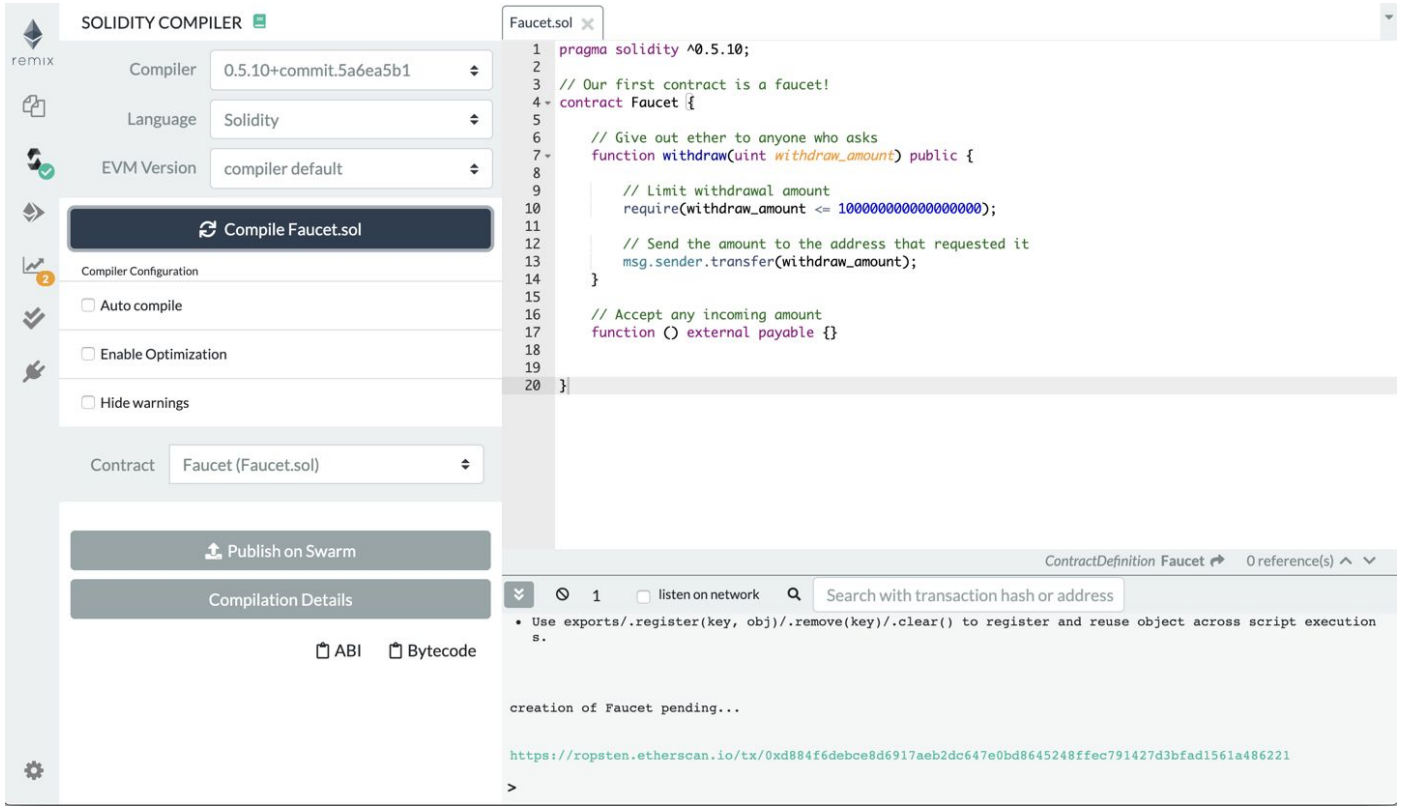
Remix IDE

https://remix.ethereum.org

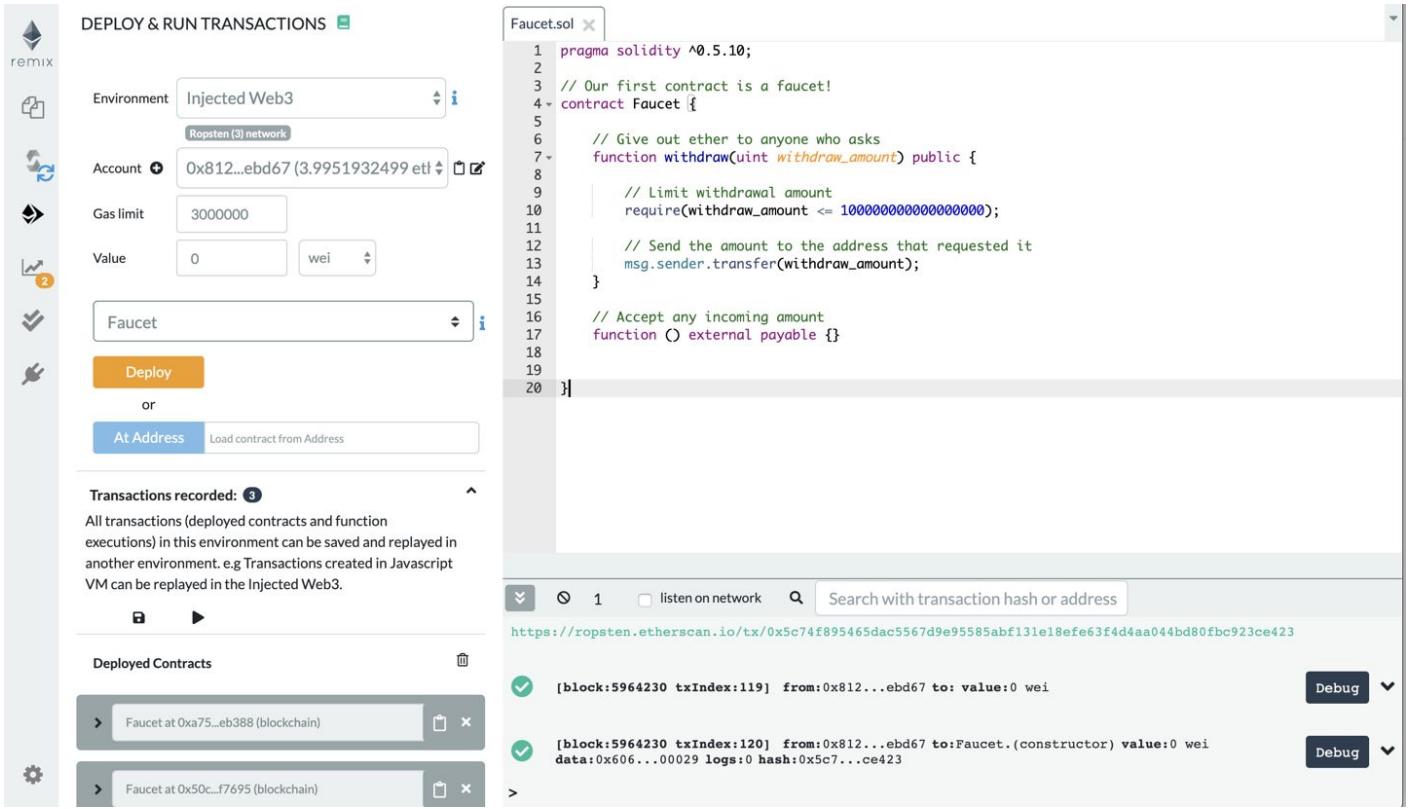
Remix IDE


FILE EXPLORERS
browser
MyMessage.sol
ballot_test.sol
ballot.sol
Faucet.sol
Faucet.sol
pragma solidity ^0.5.10;
// Our first contract is a faucet!
contract Faucet {
// Give out ether to anyone who asks
function withdraw(uint withdraw_amount) public {
// Limit withdrawal amount
require(withdraw_amount <= 10000000000000000);
// Send the amount to the address that requested it
msg.sender.transfer(withdraw_amount);
}
// Accept any incoming amount
function () external payable {}
}
ContractDefinition Faucet
0 reference(s)
listen on network
Search with transaction hash or address
Use exports/.register(key, obj)/.remove(key)/.clear() to register and reuse object across script executions.
creation of Faucet pending...
https://ropsten.etherscan.io/tx/0xd884f6debce8d6917aeb2dc647e0bd8645248ffec791427d3bfad1561a486221
>

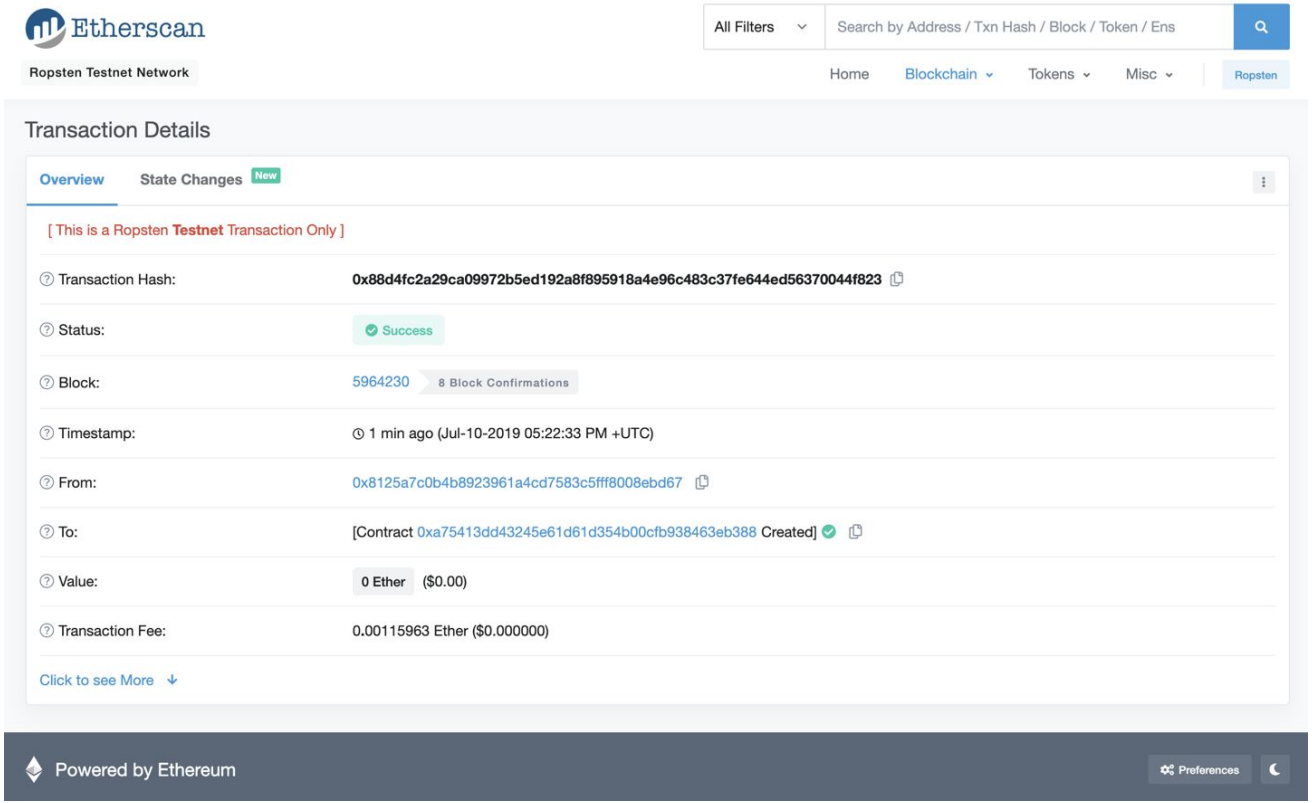
Remix IDE


SOLIDITY COMPILER
Compiler
0.5.10+commit.5a6ea5b1
Language
Solidity
EVM Version
compiler default
Compile Faucet.sol
Compiler Configuration
Auto compile
Enable Optimization
Hide warnings
Contract
Faucet (Faucet.sol)
Publish on Swarm
Compilation Details
ABI
Bytecode
Faucet.sol
pragma solidity ^0.5.10;
// Our first contract is a faucet!
contract Faucet {
// Give out ether to anyone who asks
function withdraw(uint withdraw_amount) public {
// Limit withdrawal amount
require(withdraw_amount <= 10000000000000000);
// Send the amount to the address that requested it
msg.sender.transfer(withdraw_amount);
}
// Accept any incoming amount
function () external payable {}
}
ContractDefinition Faucet
0 reference(s)
listen on network
Search with transaction hash or address
Use exports/.register(key, obj)/.remove(key)/.clear() to register and reuse object across script executions.
creation of Faucet pending...
https://ropsten.etherscan.io/tx/0xd884f6debce8d6917aeb2dc647e0bd8645248ffec791427d3bfad1561a486221

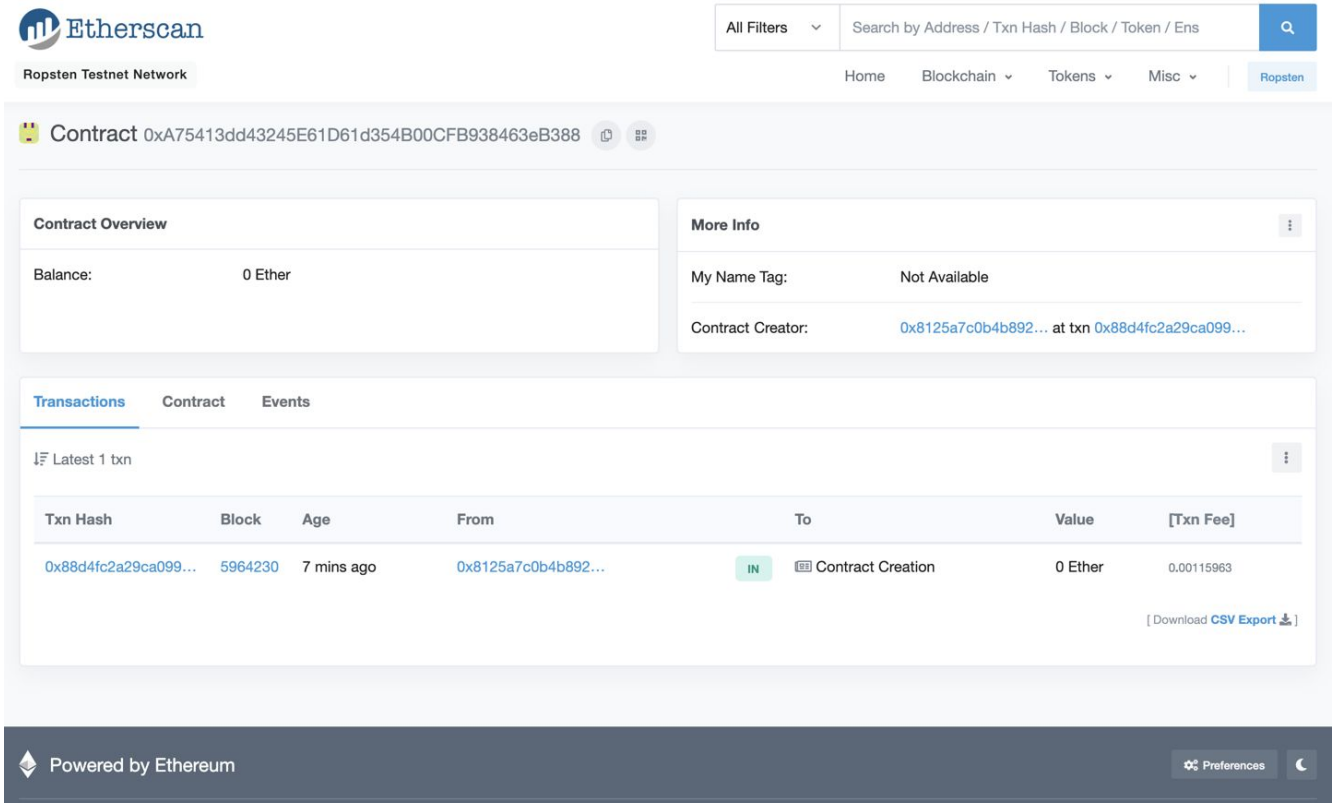
Remix IDE


DEPLOY & RUN TRANSACTIONS
Environment
Injected Web3
Ropsten (3) network
Account
0x812...ebd67 (3.9951932499 etl
Gas limit
3000000
Value
0
wei
Faucet
Deploy
or
At Address
Load contract from Address
Transactions recorded: 3
All transactions (deployed contracts and function executions) in this environment can be saved and replayed in another environment. e.g Transactions created in Javascript VM can be replayed in the Injected Web3.
Deployed Contracts
Faucet at 0xa75...eb388 (blockchain)
Faucet at 0x50c...f7695 (blockchain)
Faucet.sol
pragma solidity ^0.5.10;
// Our first contract is a faucet!
contract Faucet {
// Give out ether to anyone who asks
function withdraw(uint withdraw_amount) public {
// Limit withdrawal amount
require(withdraw_amount <= 10000000000000000);
// Send the amount to the address that requested it
msg.sender.transfer(withdraw_amount);
}
// Accept any incoming amount
function () external payable {}
}
listen on network
Search with transaction hash or address
https://ropsten.etherscan.io/tx/0x5c74f895465dac5567d9e95585abf131e18efe63f4d4aa044bd80fbc923ce423
[block:5964230 txIndex:119] from:0x812...ebd67 to: value:0 wei
Debug
[block:5964230 txIndex:120] from:0x812...ebd67 to:Faucet.(constructor) value:0 wei
data:0x606...00029 logs:0 hash:0x5c7...ce423
Debug

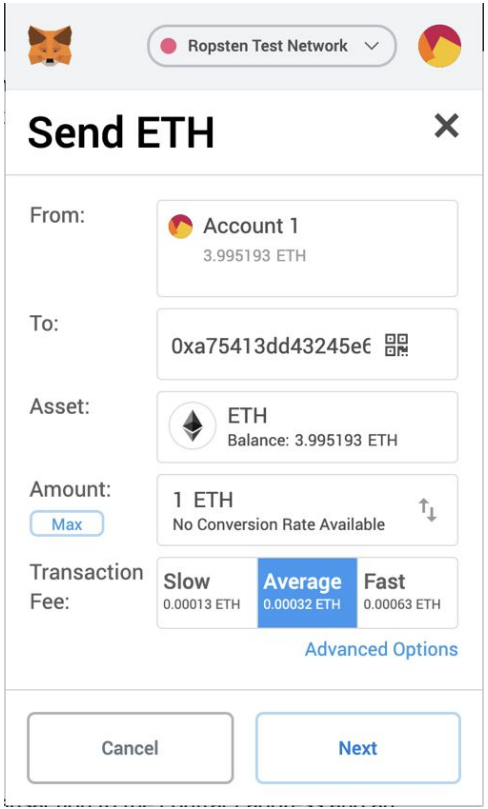
Remix IDE


Etherscan
Ropsten Testnet Network
All Filters
Search by Address / Txn Hash / Block / Token / Ens
Home
Blockchain
Tokens
Misc
Ropsten
Transaction Details
Overview
State Changes
New
[This is a Ropsten Testnet Transaction Only]
Transaction Hash:
0x88d4fc2a29ca09972b5ed192a8f895918a4e96c483c37fe644ed56370044f823
Status:
Success
Block:
5964230
8 Block Confirmations
Timestamp:
1 min ago (Jul-10-2019 05:22:33 PM +UTC)
From:
0x8125a7c0b4b8923961a4cd7583c5fff8008ebd67
To:
[Contract 0xa75413dd43245e61d61d354b00cfb938463eb388 Created]
Value:
0 Ether
($0.00)
Transaction Fee:
0.00115963 Ether ($0.000000)
Click to see More
Powered by Ethereum
Preferences

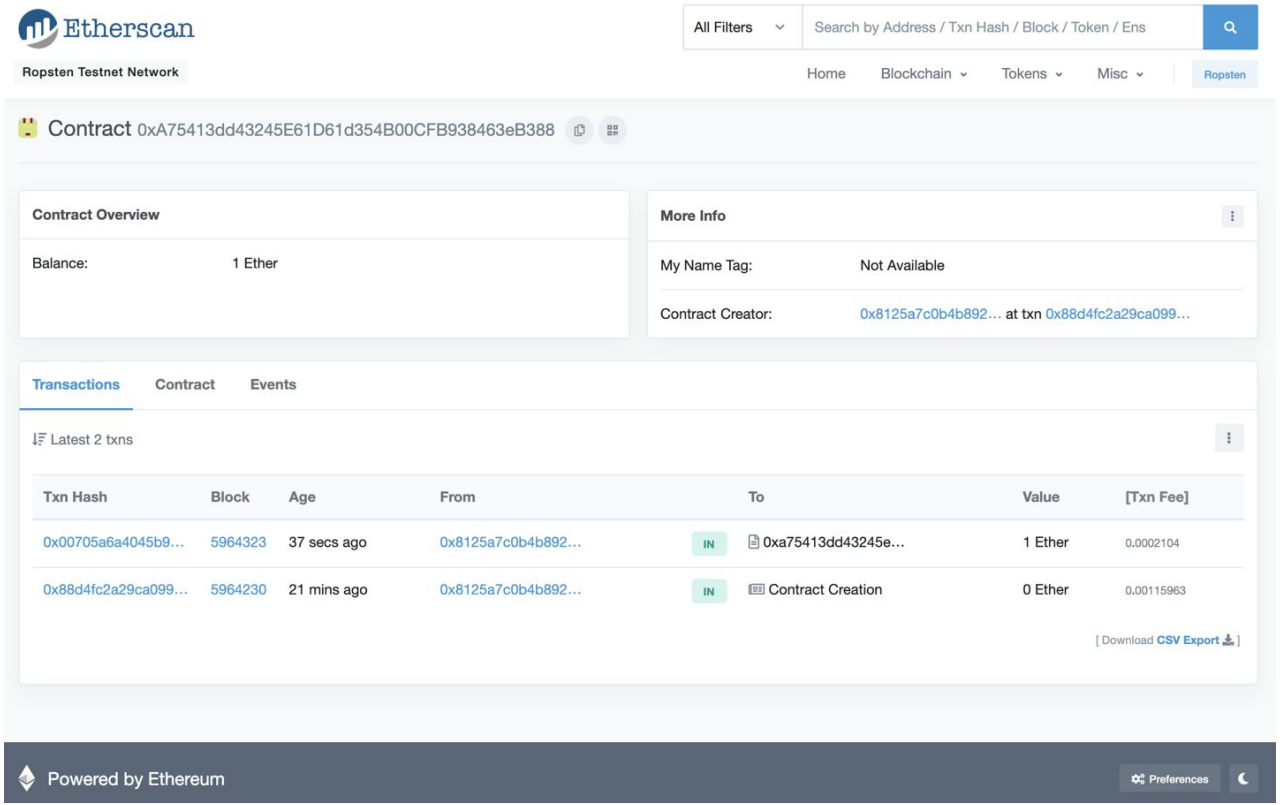
Remix IDE


Etherscan
Ropsten Testnet Network
All Filters
Search by Address / Txn Hash / Block / Token / Ens
Home
Blockchain
Tokens
Misc
Ropsten
Contract 0xA75413dd43245E61D61d354B00CFB938463eB388
Contract Overview
Balance:
0 Ether
More Info
My Name Tag:
Not Available
Contract Creator:
0x8125a7c0b4b892... at txn 0x88d4fc2a29ca099...
Transactions
Contract
Events
Latest 1 txn
Txn Hash
Block
Age
From
To
Value
[Txn Fee]
0x88d4fc2a29ca099...
5964230
7 mins ago
0x8125a7c0b4b892...
IN
Contract Creation
0 Ether
0.00115963
[Download CSV Export]
Powered by Ethereum
Preferences

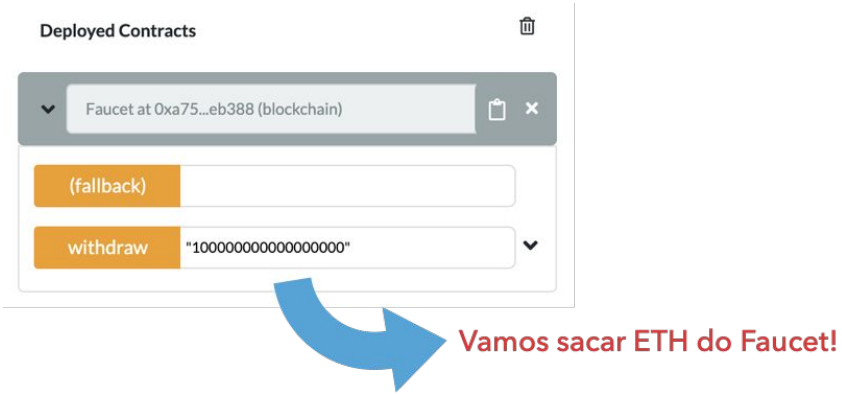
Remix IDE


Ropsten Test Network
Send ETH
From:
Account 1
3.995193 ETH
To:
0xa75413dd43245e6
Asset:
ETH
Balance: 3.995193 ETH
Amount:
Max
1 ETH
No Conversion Rate Available
Transaction Fee:
Slow
0.00013 ETH
Average
0.00032 ETH
Fast
0.00063 ETH
Advanced Options
Cancel
Next

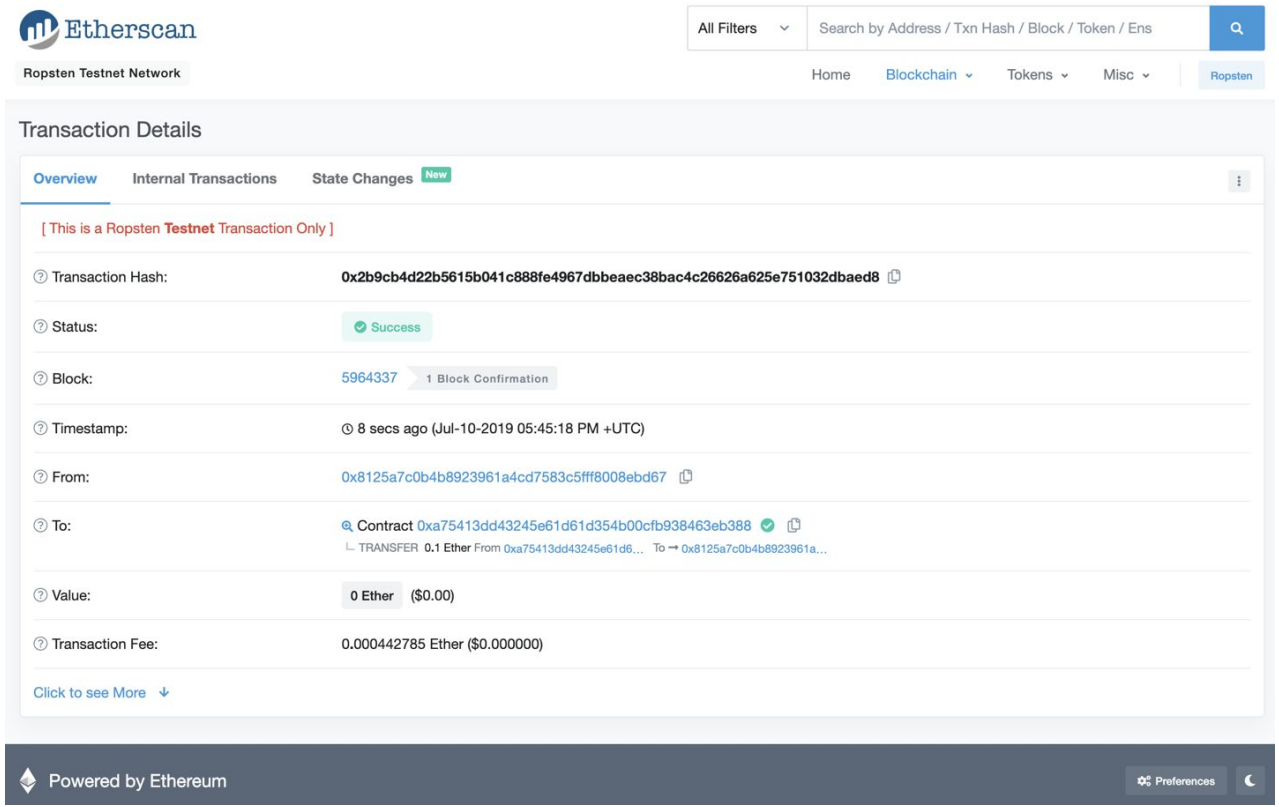
Remix IDE


Etherscan
Ropsten Testnet Network
All Filters
Search by Address / Txn Hash / Block / Token / Ens
Home
Blockchain
Tokens
Misc
Ropsten
Contract 0xA75413dd43245E61D61d354B00CFB938463eB388
Contract Overview
Balance:
1 Ether
More Info
My Name Tag:
Not Available
Contract Creator:
0x8125a7c0b4b892... at txn 0x88d4fc2a29ca099...
Transactions
Contract
Events
Latest 2 txns
Txn Hash
Block
Age
From
To
Value
[Txn Fee]
0x00705a6a4045b9...
5964323
37 secs ago
0x8125a7c0b4b892...
IN
0xa75413dd43245e...
1 Ether
0.0002104
0x88d4fc2a29ca099...
5964230
21 mins ago
0x8125a7c0b4b892...
IN
Contract Creation
0 Ether
0.00115963
[Download CSV Export]
Powered by Ethereum
Preferences

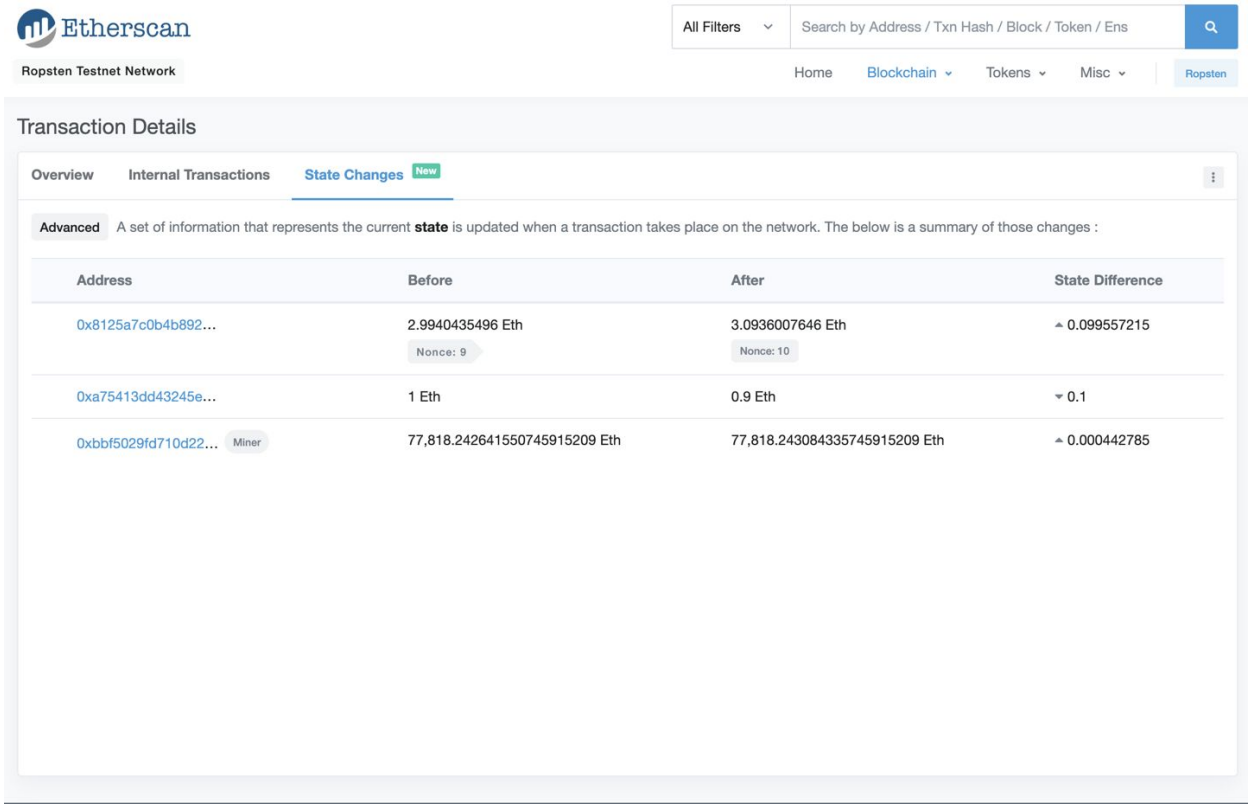
Remix IDE


Deployed Contracts
Faucet at 0xa75...eb388 (blockchain)
(fallback)
withdraw
"100000000000000000"
Vamos sacar ETH do Faucet!

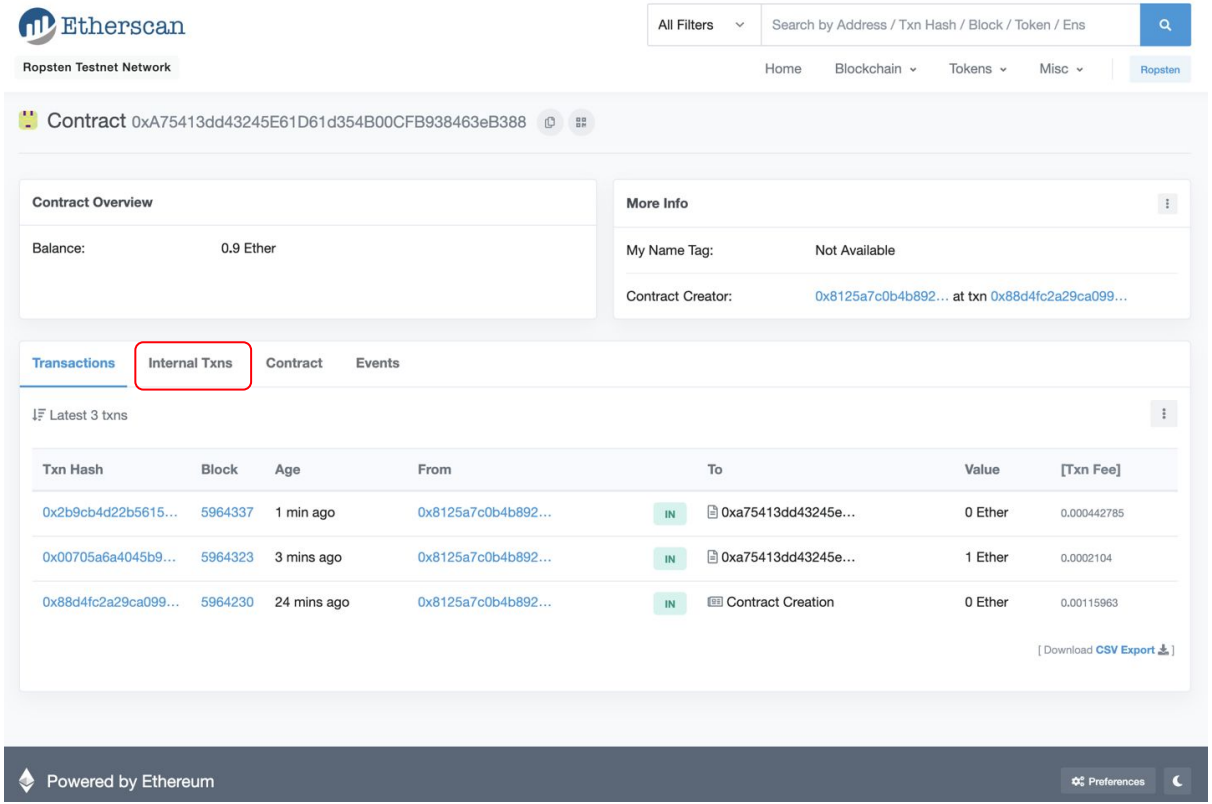
Remix IDE


Etherscan
Ropsten Testnet Network
All Filters
Search by Address / Txn Hash / Block / Token / Ens
Home
Blockchain
Tokens
Misc
Ropsten
Transaction Details
Overview
Internal Transactions
State Changes
New
[This is a Ropsten Testnet Transaction Only]
Transaction Hash:
0x2b9cb4d22b5615b041c888fe4967dbbeaec38bac4c26626a625e751032dbaed8
Status:
Success
Block:
5964337
1 Block Confirmation
Timestamp:
8 secs ago (Jul-10-2019 05:45:18 PM +UTC)
From:
0x8125a7c0b4b8923961a4cd7583c5fff8008ebd67
To:
Contract 0xa75413dd43245e61d61d354b00cfb938463eb388
TRANSFER 0.1 Ether From 0xa75413dd43245e61d6... To 0x8125a7c0b4b8923961a...
Value:
0 Ether ($0.00)
Transaction Fee:
0.000442785 Ether ($0.000000)
Click to see More
Powered by Ethereum
Preferences

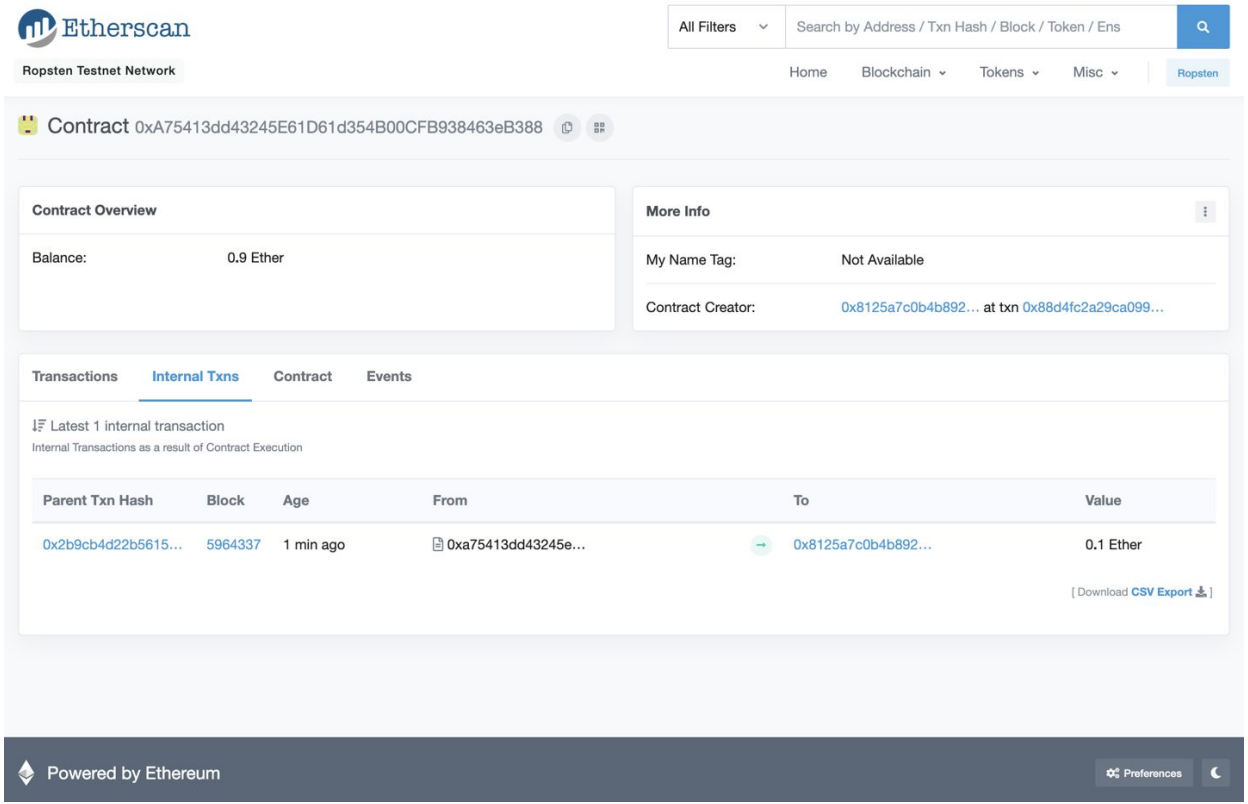
Remix IDE


Etherscan
Ropsten Testnet Network
All Filters
Search by Address / Txn Hash / Block / Token / Ens
Home
Blockchain
Tokens
Misc
Ropsten
Transaction Details
Overview
Internal Transactions
State Changes
New
Advanced
A set of information that represents the current state is updated when a transaction takes place on the network. The below is a summary of those changes :
Address
Before
After
State Difference
0x8125a7c0b4b892...
2.9940435496 Eth
Nonce: 9
3.0936007646 Eth
Nonce: 10
0.099557215
0xa75413dd43245e...
1 Eth
0.9 Eth
0.1
0xbbf5029fd710d22...
Miner
77,818.242641550745915209 Eth
77,818.243084335745915209 Eth
0.000442785

Remix IDE


Etherscan
Ropsten Testnet Network
All Filters
Search by Address / Txn Hash / Block / Token / Ens
Home
Blockchain
Tokens
Misc
Ropsten
Contract 0xA75413dd43245E61D61d354B00CFB938463eB388
Contract Overview
Balance: 0.9 Ether
More Info
My Name Tag: Not Available
Contract Creator: 0x8125a7c0b4b892... at txn 0x88d4fc2a29ca099...
Transactions
Internal Txns
Contract
Events
Latest 3 txns
Txn Hash | Block | Age | From | To | Value | [Txn Fee]
0x2b9cb4d22b5615... | 5964337 | 1 min ago | 0x8125a7c0b4b892... | IN | 0xa75413dd43245e... | 0 Ether | 0.000442785
0x00705a6a4045b9... | 5964323 | 3 mins ago | 0x8125a7c0b4b892... | IN | 0xa75413dd43245e... | 1 Ether | 0.0002104
0x88d4fc2a29ca099... | 5964230 | 24 mins ago | 0x8125a7c0b4b892... | IN | Contract Creation | 0 Ether | 0.00115963
[Download CSV Export]
Powered by Ethereum
Preferences

Remix IDE


Etherscan
Ropsten Testnet Network
All Filters
Search by Address / Txn Hash / Block / Token / Ens
Home
Blockchain
Tokens
Misc
Ropsten
Contract 0xA75413dd43245E61D61d354B00CFB938463eB388
Contract Overview
Balance:
0.9 Ether
More Info
My Name Tag:
Not Available
Contract Creator:
0x8125a7c0b4b892… at txn 0x88d4fc2a29ca099…
Transactions
Internal Txns
Contract
Events
Latest 1 internal transaction
Internal Transactions as a result of Contract Execution
Parent Txn Hash
Block
Age
From
To
Value
0x2b9cb4d22b5615…
5964337
1 min ago
0xa75413dd43245e…
0x8125a7c0b4b892…
0.1 Ether
[Download CSV Export]
Powered by Ethereum
Preferences

ABI - Application Binary Interface

Interface entre dois módulos

Similar a API, porém em linguagem de máquina

No Ethereum, o ABI define as funções em um contrato que podem ser invocadas e descreve como cada função irá receber argumentos e seus retornos

Especificado em *JSON*

ABI - Application Binary Interface

Contrato em Solidity

```
pragma solidity ^0.5.10;

contract Faucet {

    function withdraw(uint withdraw_amount) public {

        require(withdraw_amount <= 100000000000000000);

        msg.sender.transfer(withdraw_amount);
    }

    function () external payable {}

}
```

ABI - Application Binary Interface

Contrato em ByteCode

```
PUSH1 0x80 PUSH1 0x40 MSTORE CALLVALUE DUP1 ISZERO PUSH2 0x10 JUMPI PUSH1 0x0 DUP1
REVERT JUMPDEST POP PUSH1 0xE7 DUP1 PUSH2 0x1F PUSH1 0x0 CODECOPY PUSH1 0x0 RETURN
INVALID PUSH1 0x80 PUSH1 0x40 MSTORE PUSH1 0x4 CALLDATASIZE LT PUSH1 0x1C JUMPI PUSH1
0x0 CALLDATALOAD PUSH1 0xE0 SHR DUP1 PUSH4 0x2E1A7D4D EQ PUSH1 0x1E JUMPI JUMPDEST
STOP JUMPDEST CALLVALUE DUP1 ISZERO PUSH1 0x29 JUMPI PUSH1 0x0 DUP1 REVERT JUMPDEST
POP PUSH1 0x53 PUSH1 0x4 DUP1 CALLDATASIZE SUB PUSH1 0x20 DUP2 LT ISZERO PUSH1 0x3E
JUMPI PUSH1 0x0 DUP1 REVERT JUMPDEST DUP2 ADD SWAP1 DUP1 DUP1 CALLDATALOAD SWAP1
PUSH1 0x20 ADD SWAP1 SWAP3 SWAP2 SWAP1 POP POP POP PUSH1 0x55 JUMP JUMPDEST STOP
JUMPDEST PUSH8 0x16345785D8A0000 DUP2 GT ISZERO PUSH1 0x69 JUMPI PUSH1 0x0 DUP1
REVERT JUMPDEST CALLER PUSH20 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF AND PUSH2
0x8FC DUP3 SWAP1 DUP2 ISZERO MUL SWAP1 PUSH1 0x40 MLOAD PUSH1 0x0 PUSH1 0x40 MLOAD
DUP1 DUP4 SUB DUP2 DUP6 DUP9 DUP9 CALL SWAP4 POP POP POP POP ISZERO DUP1 ISZERO PUSH1
0xAE JUMPI RETURNDATASIZE PUSH1 0x0 DUP1 RETURNDATACOPY RETURNDATASIZE PUSH1 0x0
REVERT JUMPDEST POP POP JUMP INVALID LOG2 PUSH6 0x627A7A723058 KECCAK256 DUP12 SDIV
DUP4 0x2d 0xc5 ADDMOD MOD BYTE 0xa7 PUSH13 0xF3B7C3499097383D5B9E9C83BD PUSH24
0x7CE0352811C9D3AC64736F6C634300050A00320000000000
```

ABI - Application Binary Interface

ABI do contrato

```
[
	{
		"constant": false,
		"inputs": [
			{
				"name": "withdraw_amount",
				"type": "uint256"
			}
		],
		"name": "withdraw",
		"outputs": [],
		"payable": false,
		"stateMutability": "nonpayable",
		"type": "function"
	},
	{
		"payable": true,
		"stateMutability": "payable",
		"type": "fallback"
	}
]
```

cryptozombies.io

```
1  pragma solidity >=0.5.0 <0.6.0;
2
3  contract HelloWorld {
4
5  }
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
```

```solidity
contract Example {
  // Isso será armazenado permanentemente no blockchain (variável de estado).
  uint myUnsignedInteger = 100;
}
```

```solidity
// Adição: x + y
// Subtração: x - y,
// Multiplicação: x * y
// Divisão: x / y
// Módulo: x % y

uint x = 5 ** 2; // igual a 5^2 = 25
```

```
1  struct Person {
2    uint age;
3    string name;
4  }
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
```

```solidity
// array com comprimento fixo de 2 elementos.
uint[2] fixedArray;
// outro array fixo com 5 elementos.
string[5] stringArray;
// array dinâmico - não tem tamanho fixo, pode crescer.
uint[] dynamicArray;

Person[] people; // array de structs dinâmico.

// você pode declarar seu array com 'public' e o Solidity cria automaticamente o getter.
// outros contratos podem ler desse array, mas não podem escrever.
Person[] public people;
```

```
// visibilidade: pública
// argumentos de tipos dados por referência (string, arrays, structs,
// mappings) é preciso explicitar ‘memory'

function eatHamburgers(string memory _name, uint _amount) public {

}

eatHamburgers(“daniel", 100);
```

```solidity
struct Person {
  uint age;
  string name;
}

Person[] public people;

// cria uma nova 'Person':
Person satoshi = Person(172, "Satoshi");

// adiciona essa 'Person' ao final do array:
people.push(satoshi);

// adiciona um novo struct direto ao final do array.
people.push(Person(16, "Vitalik"));

uint[] numbers;
numbers.push(5);
numbers.push(10);
numbers.push(15);
// 'numbers' agora é igual a [5, 10, 15]
```

```sol
// em solidity, funções são públicas por padrão.
// se definido como 'private' somente outras funções dentro do mesmo contrato poderão chamar a função
// por padrão, normalmente iniciamos o nome da função com _ quando ela é privada

uint[] numbers;

function _addToArray(uint _number) private {
  numbers.push(_number);
}

// Para definir um retorno usamos a palavra-chave returns:
// modificador ‘view': quando somente lemos dados do blockchain, e não escrevemos.

string greeting = "Hello!";

function sayHello() public view returns (string memory) {
  return greeting;
}

// modificador ‘pure’: quando não precisamos nem ler do blockchain.
function _multiply(uint a, uint b) private pure returns (uint) {
  return a * b;
}
```

```
1  uint8 a = 5;
2  uint b = 6;
3  // gera um erro porque a * b retorna um uint, e não uint8.
4  uint8 c = a * b;
5  // typecasting 'b' para uint8
6  uint8 c = a * uint8(b);
7
8  // o Ethereum tem uma função keccak256 (SHA3). Gera uma hash de 256 bits.
9  // keccak256 espera um dado do tipo 'bytes'.
10
11 keccak256(abi.encodePacked("aaaab")); //6e91ec6b618bb462a4a6ee5aa2cb0e9cf30f7a052bb467b0ba58b8748c00d2e5
12 keccak256(abi.encodePacked("aaaac")); //b1f078126895a1424524de5321b339ab00408010b7cf0e6ed451514981e58aa9
13
14 // Gerando números aleatórios entre 1 e 100:
15 uint random = uint(keccak256(abi.encodePacked(now, msg.sender))) % 100;
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
```

```solidity
// declara o evento
event IntegersAdded(uint x, uint y, uint result);

function add(uint _x, uint _y) public returns (uint) {
  uint result = _x + _y;
  // dispara um evento que deixa o app saber que a função foi chamada.
  emit IntegersAdded(_x, _y, result);
  return result;
}
```

```solidity
// se por exemplo, quisermos armazenar o saldo de um usuário no nosso app.
mapping (address => uint) public accountBalance;
// ou se precisamos armazenar usuarios baseados em seus ids.
mapping (uint => string) userIdToName;

// utilizando msg.sender

mapping (address => uint) favoriteNumber;

function setMyNumber(uint _myNumber) public {
  // atualiza o mapping `favoriteNumber` para armazenar `_myNumber` na chave `msg.sender`
  favoriteNumber[msg.sender] = _myNumber;
  // ^ A sintaxe de atribuir valores em um mapping é similar a arrays.
}

function whatIsMyNumber() public view returns (uint) {
  // retorna o valor armazenado na chave 'endereço do remetente'
  // será `0` se o remetente ainda não chamou`setMyNumber`
  return favoriteNumber[msg.sender];
}
```

```solidity
function sayHi(string memory _name) public returns (string memory) {
  // Compara se _name é igual a "Ethereum". Lança um erro e a execução para se não for verdade.
  // Obs: Solidity não tem comparação de strings nativo, então comparamos a hash de cada string
  // utilizando o keccak256.
  require(keccak256(abi.encodePacked(_name)) == keccak256(abi.encodePacked("Ethereum")));
  // se verdadeiro, continua a execução
  return "Hi!";
}
```

```
contract Doge {
  function catchphrase() public returns (string memory) {
    return "So Wow CryptoDoge";
  }
}

contract BabyDoge is Doge {
  function anotherCatchphrase() public returns (string memory) {
    return "Such Moon BabyDoge";
  }
}
```

```solidity
import "./someothercontract.sol";

contract newContract is SomeOtherContract {

}
```

```solidity
contract SandwichFactory {
  struct Sandwich {
    string name;
    string status;
  }

  Sandwich[] sandwiches;

  function eatSandwich(uint _index) public {
    // Sandwich mySandwich = sandwiches[_index];
    // ^ parece bem direto, mas o Solidity vai ter um warning informando que
    // você deve explicitamente declarar `storage` ou `memory`

    // portanto você deve declarar com a palavra-chave `storage`:
    Sandwich storage mySandwich = sandwiches[_index];
    // ...onde no caso `mySandwich` é um ponteiro para `sandwiches[_index]`
    // em storage, e...
    mySandwich.status = "Eaten!";
    // ...isso irá permanentemente modificar `sandwiches[_index]` no blockchain.

    // se você só quer uma cópia, você pode usar `memory`:
    Sandwich memory anotherSandwich = sandwiches[_index + 1];
    // ...que no caso `anotherSandwich` irá simplemente ser uma cópia do valor
    // em memória, e...
    anotherSandwich.status = "Eaten!";
    // ...irá modificar somente a variável temporária e não terá efeito em
    // `sandwiches[_index + 1]`. Mas você pode fazer isso:
    sandwiches[_index + 1] = anotherSandwich;
    // ...se você quer copiar as mudanças de volta ao blockchain (storage).
  }
}
```

```solidity
contract Sandwich {
  uint private sandwichesEaten = 0;

  function eat() internal {
    sandwichesEaten++;
  }
}

contract BLT is Sandwich {
  uint private baconSandwichesEaten = 0;

  function eatWithBacon() public returns (string memory) {
    baconSandwichesEaten++;
    // Podemos chamar aqui porque é internal
    eat();
  }
}
```

```solidity
// contrato implantado no blockchain no endereço de contrato
// 0x06012c8cf97BEaD5deAe237070F9587f8E7A266d

contract LuckyNumber {
  mapping(address => uint) numbers;

  function setNum(uint _num) public {
    numbers[msg.sender] = _num;
  }

  function getNum(address _myAddress) public view returns (uint) {
    return numbers[_myAddress];
  }
}
```

```solidity
contract NumberInterface {
  function getNum(address _myAddress) public view returns (uint);
}

contract MyContract {
  address NumberInterfaceAddress = 0x06012c8cf...;
  // ^ o endereço do contrato LuckyNumber no Ethereum
  NumberInterface numberContract =
NumberInterface(NumberInterfaceAddress);
  // Agora `numberContract` está apontando para o outro contrato

  function someFunction() public {
    // Agora podemos chamar `getNum` do outro contrato:
    uint num = numberContract.getNum(msg.sender);
    // ...e fazer alguma coisa com `num`
  }
}
```

```sol
contract Ownable {

  address private _owner;

  event OwnershipTransferred(
    address indexed previousOwner,
    address indexed newOwner
  );

  constructor() internal {
    _owner = msg.sender;
    emit OwnershipTransferred(address(0), _owner);
  }

  function owner() public view returns(address) {
    return _owner;
  }

  modifier onlyOwner() {
    require(isOwner());
    _;
  }

  function isOwner() public view returns(bool) {
    return msg.sender == _owner;
  }

  function renounceOwnership() public onlyOwner {
    emit OwnershipTransferred(_owner, address(0));
    _owner = address(0);
  }

  function transferOwnership(address newOwner) public onlyOwner {
    _transferOwnership(newOwner);
  }

  function _transferOwnership(address newOwner) internal {
    require(newOwner != address(0));
    emit OwnershipTransferred(_owner, newOwner);
    _owner = newOwner;
  }
}
```

https://openzeppelin.com/

```
pragma solidity >=0.5.0 <0.6.0;

import "./ownable.sol";

contract ZombieFactory is Ownable {
    Zombie[] public zombies;
}

contract KittyInterface {
  function getKitty(uint256 _id) external view returns (
    bool isGestating,
    bool isReady,
    uint256 cooldownIndex,
    uint256 nextActionAt,
    uint256 siringWithId,
    uint256 birthTime,
    uint256 matronId,
    uint256 sireId,
    uint256 generation,
    uint256 genes
  );
}

contract ZombieFeeding is ZombieFactory {

  KittyInterface kittyContract;

  function setKittyContractAddress(address _address) external onlyOwner {
    kittyContract = KittyInterface(_address);
  }

}
```

```solidity
// um mapping para armazenar a idade do usuário:
mapping (uint => uint) public age;

// modificador que requer que o usuário seja mais velho que uma determinada idade:
modifier olderThan(uint _age, uint _userId) {
  require(age[_userId] >= _age);
  _;
}

// deve ter mais que 18 anos para dirigir um carro.
// podemos chamar o modificador `olderThan` com argumentos:
function driveCar(uint _userId) public olderThan(18, _userId) {
  // a lógica da função
}


// podemos acumular vários modificadores
function test() external view onlyOwner anotherModifier { /* ... */ }
```

```solidity
uint lastUpdated;

// Seta `lastUpdated` para `now`
function updateTimestamp() public {
  lastUpdated = now;
}

// irá retornar `true` se 5min se passaram desde que `updateTimestamp` foi
// chamada, `false` se os 5min não passaram
function fiveMinutesHavePassed() public view returns (bool) {
  return (now >= (lastUpdated + 5 minutes));
}
```

```
1  // if, else, while, do, for, break, continue, return são similares ao C/Javascript.
2
3  // https://docs.soliditylang.org/
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
```

```
contract OnlineStore {
  function buySomething() external payable {
    // checa se 0.001 ether foi enviado para a chamada da função:
    require(msg.value == 0.001 ether);
    // se sim, alguma lógica de transferência pode ser feita.
    transferThing(msg.sender);
  }
}
```

```solidity
contract GetPaid is Ownable {

  function withdraw() external onlyOwner {
    address payable _owner = address(uint160(owner()));
    _owner.transfer(address(this).balance);
  }

}
```

Variáveis globais

O objeto **msg** é a chamada da transação (se proveniente de EAO) ou chamada de mensagem (se proveniente de contrato) que lançou a execução do presente contrato

msg.sender representa o endereço que iniciou a chamada ao contrato

msg.value o valor de ether (em wei) enviado por essa chamada

msg.data a carga útil de dados dessa chamada para nosso contrato

msg.sig os primeiros 4 *bytes* da carga útil de dados, que é o identificador de função

Variáveis globais

O objeto **tx** provê meios para acessar informações relacionadas a transação

tx.gasprice o gasPrice da transação de chamada

tx.origin o endereço da EOA que originou a transação

Variáveis globais

O objeto **block** contém informações sobre o bloco atual:

block.coinbase o endereço do destinatário das taxas e recompensa deste bloco

block.number O número do bloco (altura)

block.timestamp O timestamp do bloco

Funções globais

blockhash(uint blockNumber) returns (bytes32)

gasleft() returns (uint256)

Funções para o tipo *address*

Objetos do tipo address tem os seguintes membros:

<address>.balance saldo do endereço em wei

<address payable>.transfer(amount) envia *amount* em wei para o endereço, lança exceção se falhar

<address payable>.send(amount) envia *amount* em wei para o endereço e retorna false se falhar

Funções relacionadas ao contrato

Outras funções embutidas notáveis:

this o endereço do contrato em execução

selfdestruct(recipientAddress) Destrói o contrato em questão, enviando os fundos restantes para o endereço passado

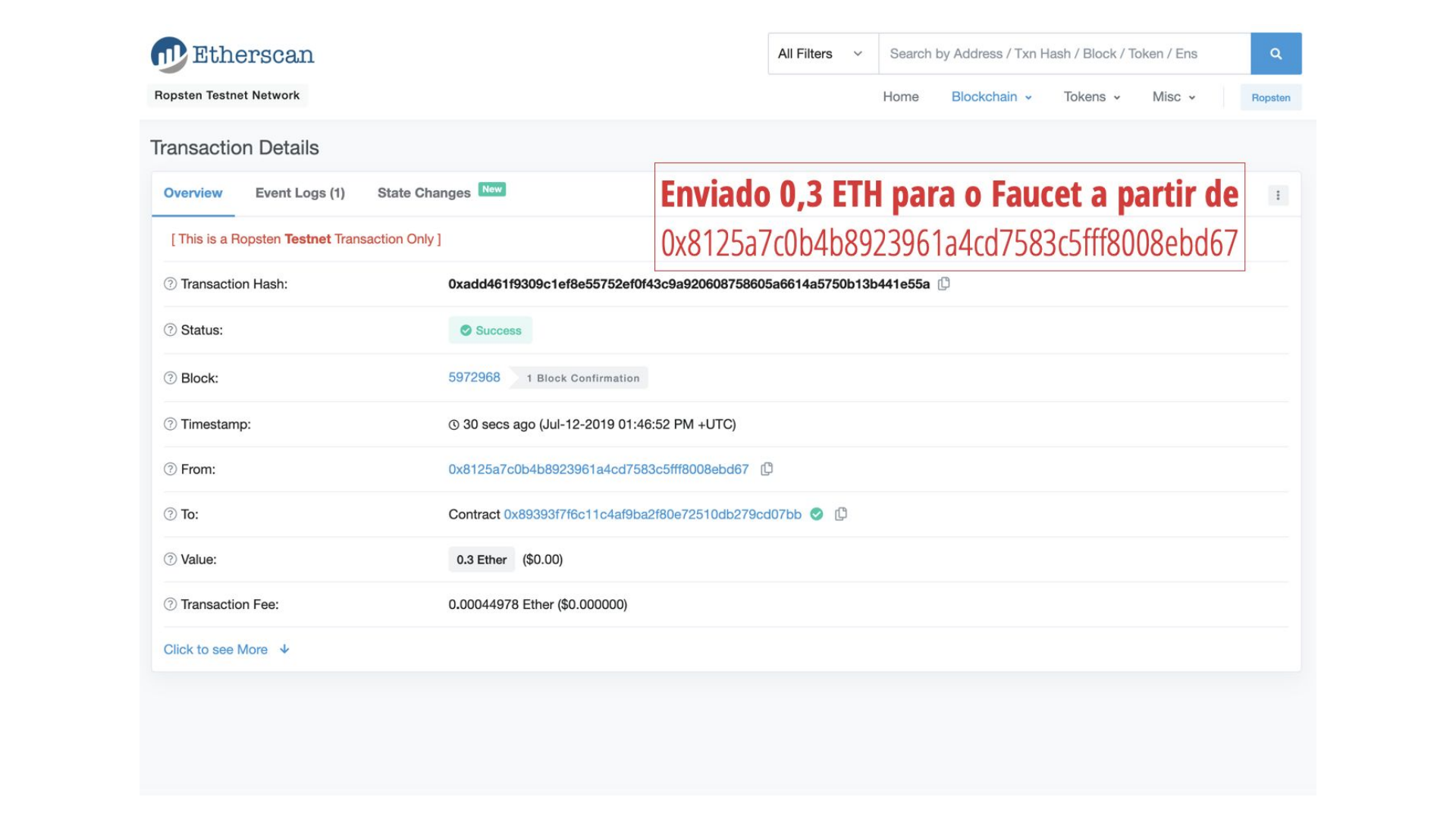
```
1  pragma solidity ^0.5.10;
2
3  contract Faucet {
4
5      address payable owner;
6
7      constructor() public {
8          owner = msg.sender;
9      }
10
11     function destroy() public {
12         require(msg.sender == owner);
13         selfdestruct(owner);
14     }
15
16     function withdraw(uint withdraw_amount) public {
17         require(withdraw_amount <= 10000000000000000);
18         msg.sender.transfer(withdraw_amount);
19     }
20
21     function () external payable {}
22
23 }
24
25
26
27
28
29
30
31
32
33
34
35
36
```

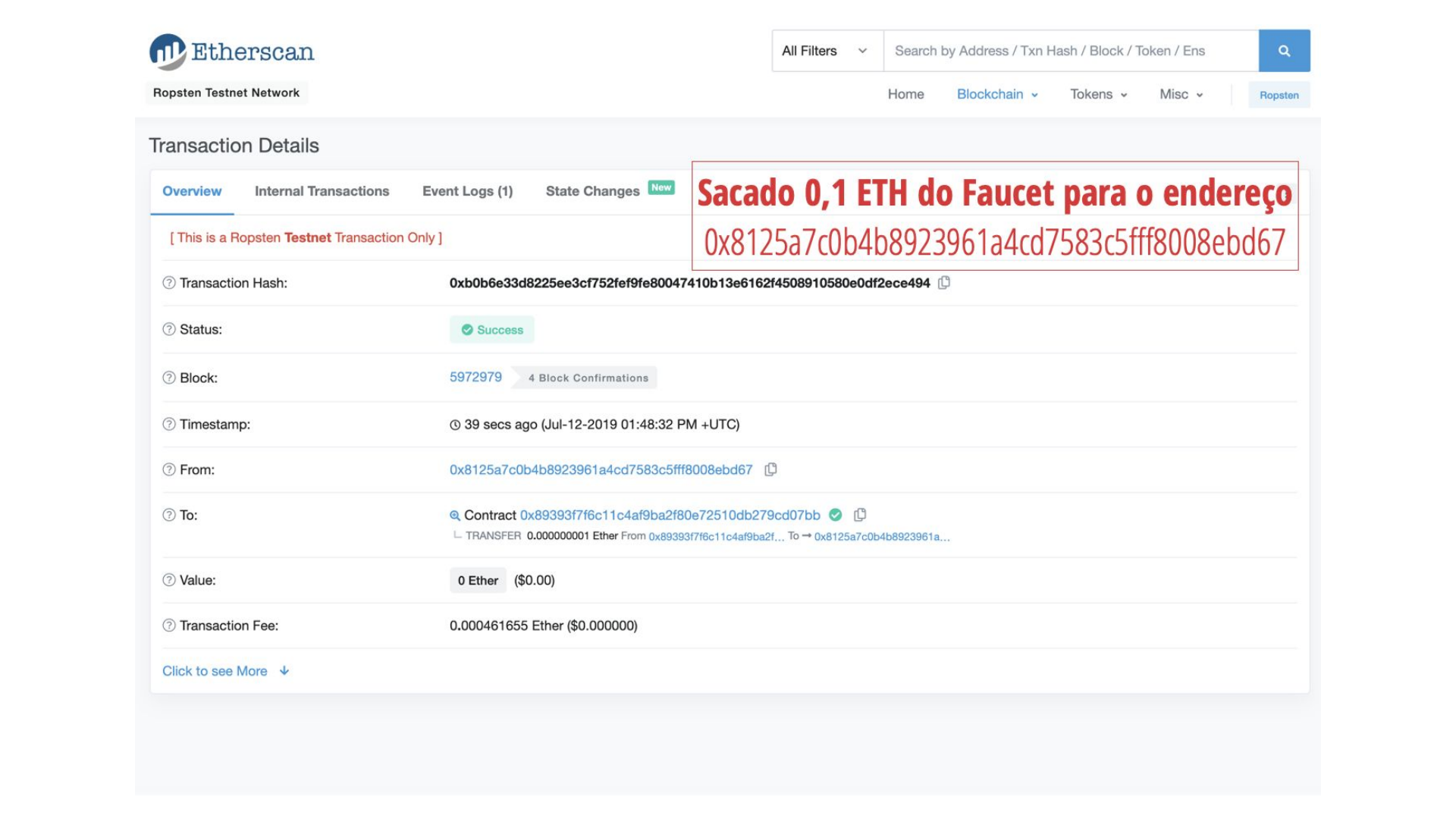
```
1  pragma solidity ^0.5.10;
2
3  contract Faucet {
4
5      address payable owner;
6
7      constructor() public {
8          owner = msg.sender;
9      }
10
11     function destroy() public onlyOwner {
12         selfdestruct(owner);
13     }
14
15     modifier onlyOwner {
16         require(msg.sender == owner);
17         _;
18     }
19
20     function withdraw(uint withdraw_amount) public {
21         require(withdraw_amount <= 100000000000000000);
22         msg.sender.transfer(withdraw_amount);
23     }
24
25     function () external payable {}
26
27 }
28
29
30
31
32
33
34
35
36
```

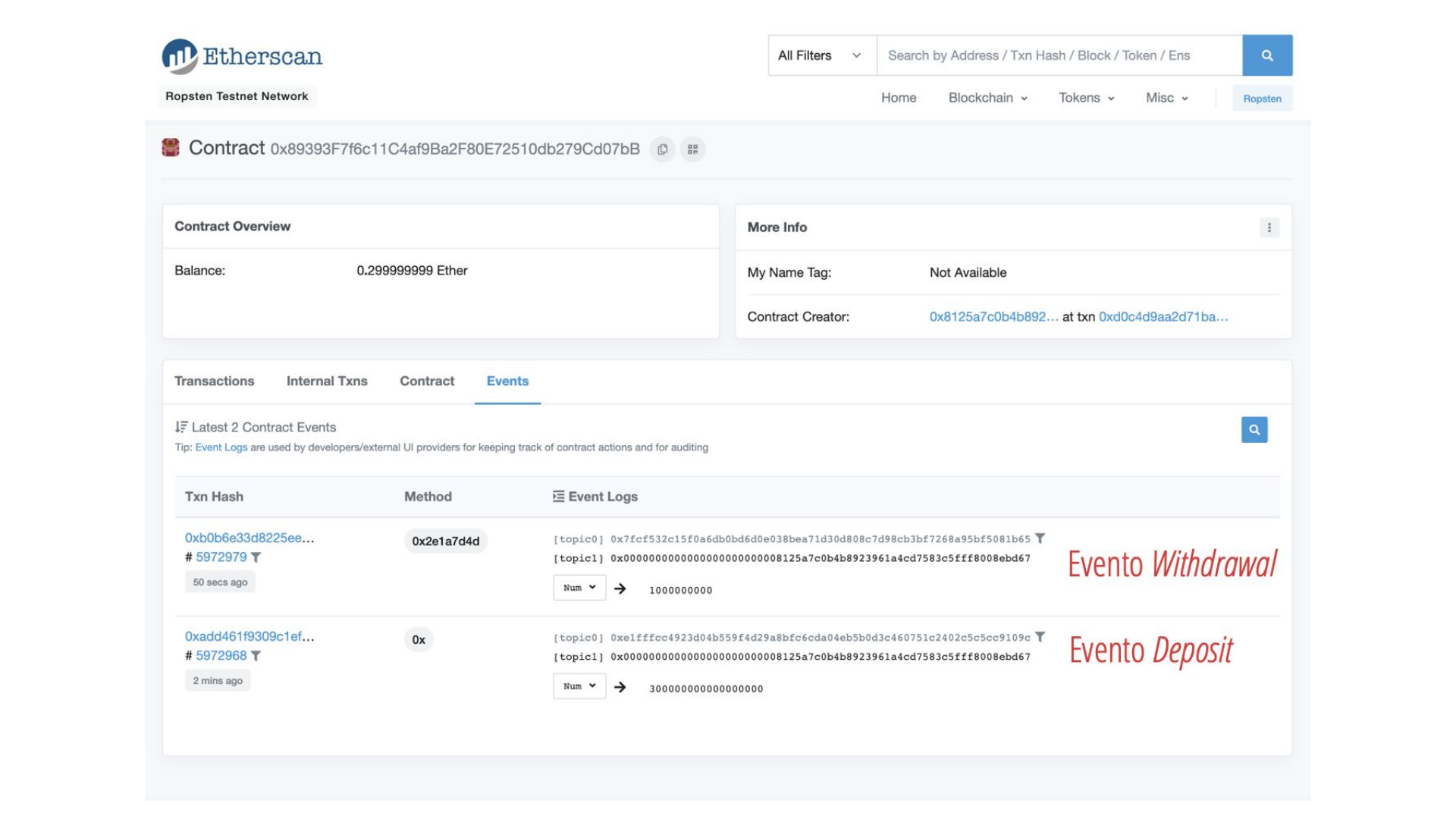
```
1  pragma solidity ^0.5.10;
2
3  contract owned {
4      address payable owner;
5
6      constructor() public {
7          owner = msg.sender;
8      }
9
10     modifier onlyOwner {
11         require(msg.sender == owner);
12         _;
13     }
14 }
15
16 contract mortal is owned {
17     function destroy() public onlyOwner {
18         selfdestruct(owner);
19     }
20 }
21
22 contract Faucet is mortal {
23
24     function withdraw(uint withdraw_amount) public {
25         require(withdraw_amount <= 0.1 ether);
26         msg.sender.transfer(withdraw_amount);
27     }
28
29     function () external payable {}
30
31 }
32
33
34
35
36
```

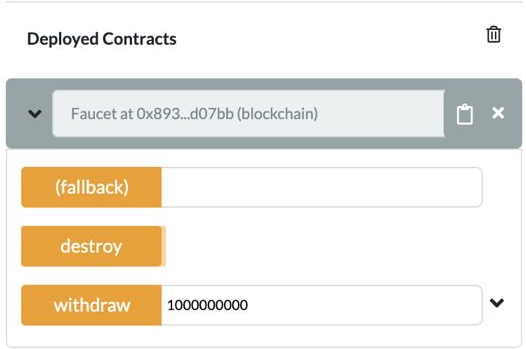
```
pragma solidity ^0.5.10;

contract owned {
    address payable owner;

    constructor() public {
        owner = msg.sender;
    }

    modifier onlyOwner {
        require(msg.sender == owner, "Somente o dono do contrato pode chamar essa função!");
        _;
    }
}

contract mortal is owned {
    function destroy() public onlyOwner {
        selfdestruct(owner);
    }
}

contract Faucet is mortal {

    function withdraw(uint withdraw_amount) public {
        require(withdraw_amount <= 0.1 ether, “Quantidade máxima atingida!”);
        msg.sender.transfer(withdraw_amount);
    }

    function () external payable {}

}
```

```
pragma solidity ^0.5.10;

import “./mortal.sol";

contract Faucet is mortal {

    event Withdrawal(address indexed to, uint amount);
    event Deposit(address indexed from, uint amount);

    function withdraw(uint withdraw_amount) public {
        require(withdraw_amount <= 0.1 ether, “Quantidade máxima atingida!”);
        msg.sender.transfer(withdraw_amount);
        emit Withdrawal(msg.sender, withdraw_amount);
    }

    function () external payable {
        emit Deposit(msg.sender, msg.value);
    }

}
```


Etherscan
All Filters
Search by Address / Txn Hash / Block / Token / Ens
Ropsten Testnet Network
Home
Blockchain
Tokens
Misc
Ropsten
Transaction Details
Overview
Event Logs (1)
State Changes
New
Enviado 0,3 ETH para o Faucet a partir de
0x8125a7c0b4b8923961a4cd7583c5fff8008ebd67
[This is a Ropsten Testnet Transaction Only]
Transaction Hash:
0xadd461f9309c1ef8e55752ef0f43c9a920608758605a6614a5750b13b441e55a
Status:
Success
Block:
5972968
1 Block Confirmation
Timestamp:
30 secs ago (Jul-12-2019 01:46:52 PM +UTC)
From:
0x8125a7c0b4b8923961a4cd7583c5fff8008ebd67
To:
Contract 0x89393f7f6c11c4af9ba2f80e72510db279cd07bb
Value:
0.3 Ether
($0.00)
Transaction Fee:
0.00044978 Ether ($0.000000)
Click to see More


Etherscan
Ropsten Testnet Network
All Filters
Search by Address / Txn Hash / Block / Token / Ens
Home
Blockchain
Tokens
Misc
Ropsten
Transaction Details
Overview
Internal Transactions
Event Logs (1)
State Changes
New
Sacado 0,1 ETH do Faucet para o endereço
0x8125a7c0b4b8923961a4cd7583c5fff8008ebd67
[This is a Ropsten Testnet Transaction Only]
Transaction Hash:
0xb0b6e33d8225ee3cf752fef9fe80047410b13e6162f4508910580e0df2ece494
Status:
Success
Block:
5972979
4 Block Confirmations
Timestamp:
39 secs ago (Jul-12-2019 01:48:32 PM +UTC)
From:
0x8125a7c0b4b8923961a4cd7583c5fff8008ebd67
To:
Contract 0x89393f7f6c11c4af9ba2f80e72510db279cd07bb
TRANSFER 0.000000001 Ether From 0x89393f7f6c11c4af9ba2f... To 0x8125a7c0b4b8923961a...
Value:
0 Ether
($0.00)
Transaction Fee:
0.000461655 Ether ($0.000000)
Click to see More


Etherscan
Ropsten Testnet Network
All Filters
Search by Address / Txn Hash / Block / Token / Ens
Home
Blockchain
Tokens
Misc
Ropsten
Contract 0x89393F7f6c11C4af9Ba2F80E72510db279Cd07bB
Contract Overview
Balance:
0.299999999 Ether
More Info
My Name Tag:
Not Available
Contract Creator:
0x8125a7c0b4b892... at txn 0xd0c4d9aa2d71ba...
Transactions
Internal Txns
Contract
Events
Latest 2 Contract Events
Tip: Event Logs are used by developers/external UI providers for keeping track of contract actions and for auditing
Txn Hash
Method
Event Logs
0xb0b6e33d8225ee...
5972979
50 secs ago
0x2e1a7d4d
[topic0] 0x7fcf532c15f0a6db0bd6d0e038bea71d30d808c7d98cb3bf7268a95bf5081b65
[topic1] 0x0000000000000000000000008125a7c0b4b8923961a4cd7583c5fff8008ebd67
Num
1000000000
Evento Withdrawal
0xadd461f9309c1ef...
5972968
2 mins ago
0x
[topic0] 0xe1fffcc4923d04b559f4d29a8bfc6cda04eb5b0d3c460751c2402c5c5cc9109c
[topic1] 0x0000000000000000000000008125a7c0b4b8923961a4cd7583c5fff8008ebd67
Num
300000000000000000
Evento Deposit



Função **destroy()**

Transaction Details

Overview | Internal Transactions | State Changes New

[This is a Ropsten **Testnet** Transaction Only]

Transaction Hash:	**0x72f0507ed4847d1c64a7352fe3bcfb1e1d320884feae6004d43a55c705f46e1b**
Status:	Success
Block:	5973047 3 Block Confirmations
Timestamp:	39 secs ago (Jul-12-2019 01:59:12 PM +UTC)
From:	0x8125a7c0b4b8923961a4cd7583c5fff8008ebd67
To:	Contract 0x89393f7f6c11c4af9ba2f80e72510db279cd07bb TRANSFER 0.299999998 Ether From 0x89393f7f6c11c4af9ba2f... To 0x8125a7c0b4b8923961a... SELF DESTRUCT Contract 0x89393f7f6c11c4af9ba2f...
Value:	0 Ether ($0.00)
Transaction Fee:	0.000121149 Ether ($0.000000)

Click to see More

Ropsten Testnet Network

All Filters | Search by Address / Txn Hash / Block / Token / Ens

Home | Blockchain | Tokens | Misc | Ropsten

Contract 0x89393F7f6c11C4af9Ba2F80E72510db279Cd07bB

Overview

Balance: 0 Ether

More Info

My Name Tag: Not Available

Contract Creator: 0x8125a7c0b4b892... at txn 0xd0c4d9aa2d71ba...

Transactions | Internal Txns | Contract Self Destruct

Contract Self Destruct called at Txn Hash 0x72f0507ed4847d1c64a7352fe3bcfb1e1d320884feae6004d43a55c705f46e1b

```
0x
```