

Desafio de Projeto - Bootcamp Blockchain Developer

Criando o seu Primeiro Token do zero usando os Padrões Web3

Projeto destina-se a desenvolver competências para criar contratos inteligentes. Mais especificamente para este desafio, criar um contrato (Token) no padrão ERC20.

Consiste basicamente das seguintes atividades:

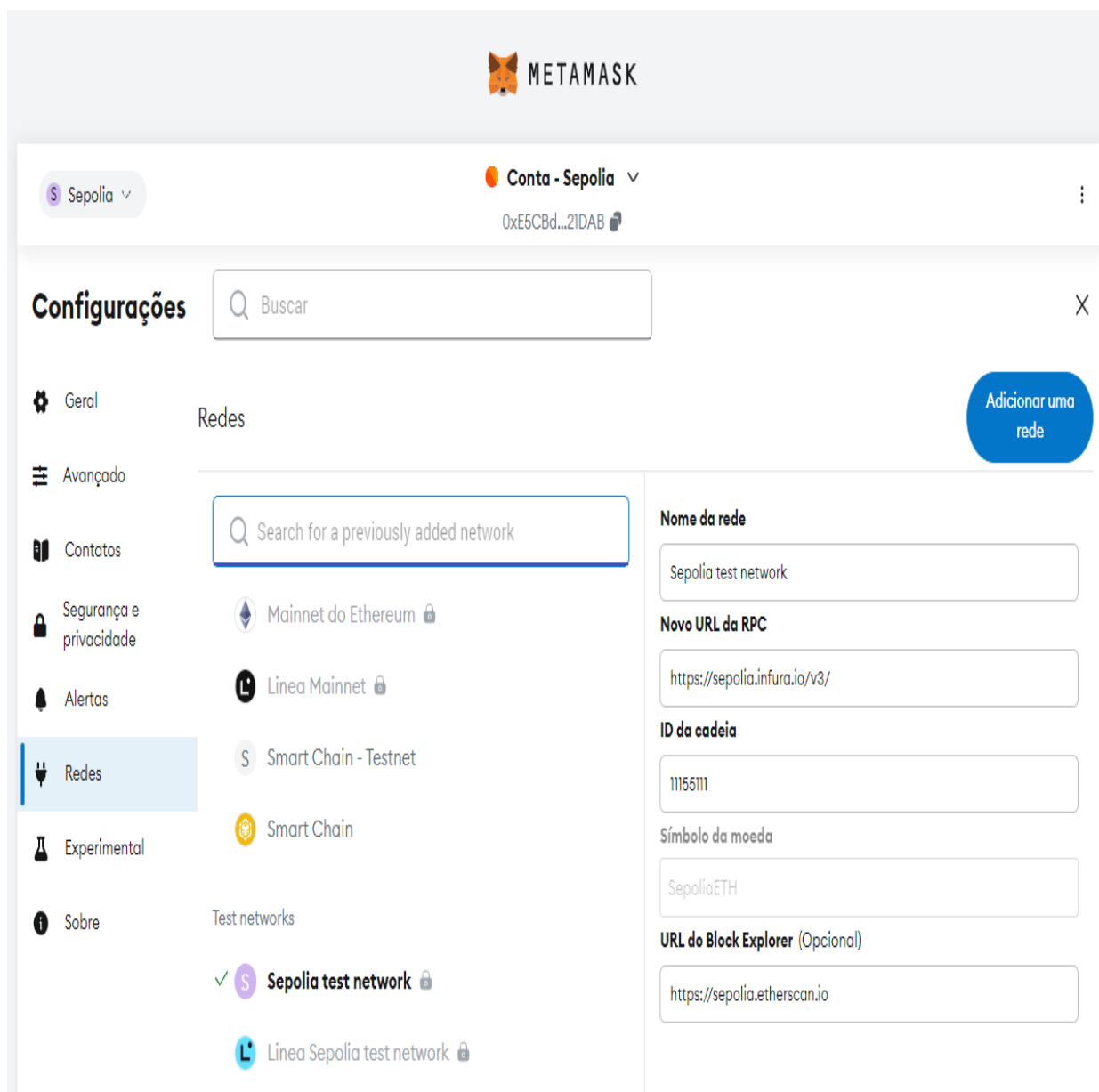
- Instalar o plugin MetaMask no google chrome e criar algumas contas associadas a uma rede (blockchain) de teste;
- Programar, compilar, fazer o deploy do contrato empregando a Remix IDE;
- Interagir com o contrato empregando a Remix IDE;
- Importar e consultar saldos do token criado para a conta criada na carteira da MetaMask.

Tecnologias envolvidas

- MetaMask (<https://metamask.io/download/>)
- Remix IDE (<https://remix.ethereum.org/>)
- Solidity – Linguagem de programação (<https://soliditylang.org/>)
- Sepolia – Blockchain (rede) de testes para deploy e execução do contrato
(<https://sepolia.infura.io/v3/>)
- Chainlink – Fornece Faucets para a rede Sepolia (<https://faucets.chain.link/sepolia>)

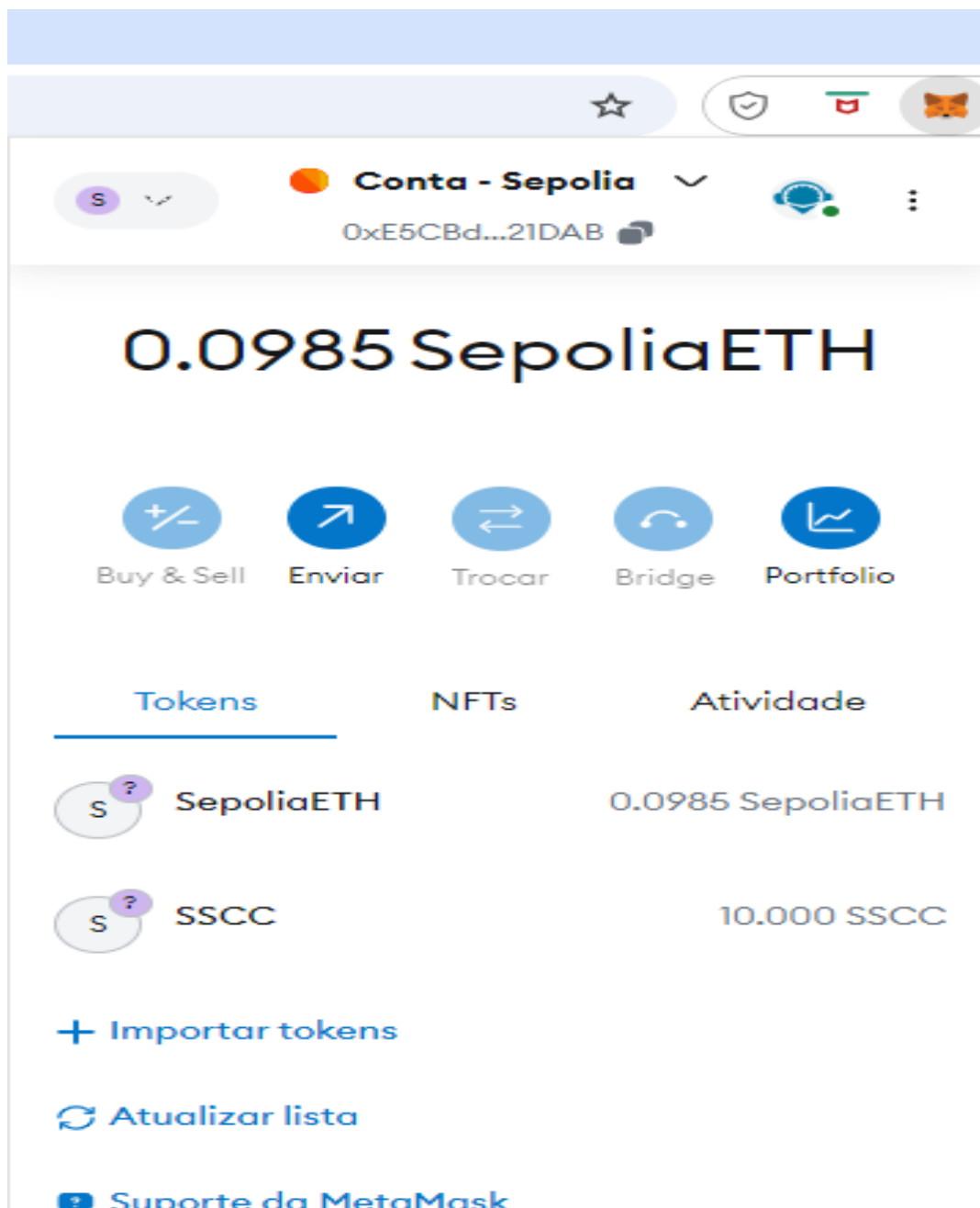
O passo a passo para concluir o desafio

1º - Instalar o MetaMask no google chrome e criar algumas contas



Plugin instalado. Adicionada a rede Sepolia (<https://sepolia.infura.io/v3/>)

Obs.: A rede de teste Sepolia é uma sugestão da própria MetaMask. Segundo o professor Ricardo Zago, pode-se encontrar conhecer as redes existentes por meio do site <https://rpc.info/>. Não encontrei a rede “Goerli” sugerida no exemplo, por isso utilizei a Sepolia.



Criada a **Conta – Sepolia**, vinculada à rede Sepolia e conectada com a Remix IDE, e abastecida com 0.1 SepoliaETH, obtidos da Chainlink (<https://faucets.chain.link/sepolia>)

2º - Programar, compilar e fazer o deploy do contrato com a Remix IDE

Apps

YouTube

Notícias

Traduzir

Nova guia

Curso: Dire, fare, pa...

CoinMarketCap

Foxbit

The World Exchang...

Santander

FILE EXPLORER

WORKSPACES

WS_SSC_001

states

contracts

artifacts

1_Storage.sol

2_Owner.sol

3_Ballot.sol

TokenERC20SSC.sol

scripts

tests

.prettierrc.json

README.txt

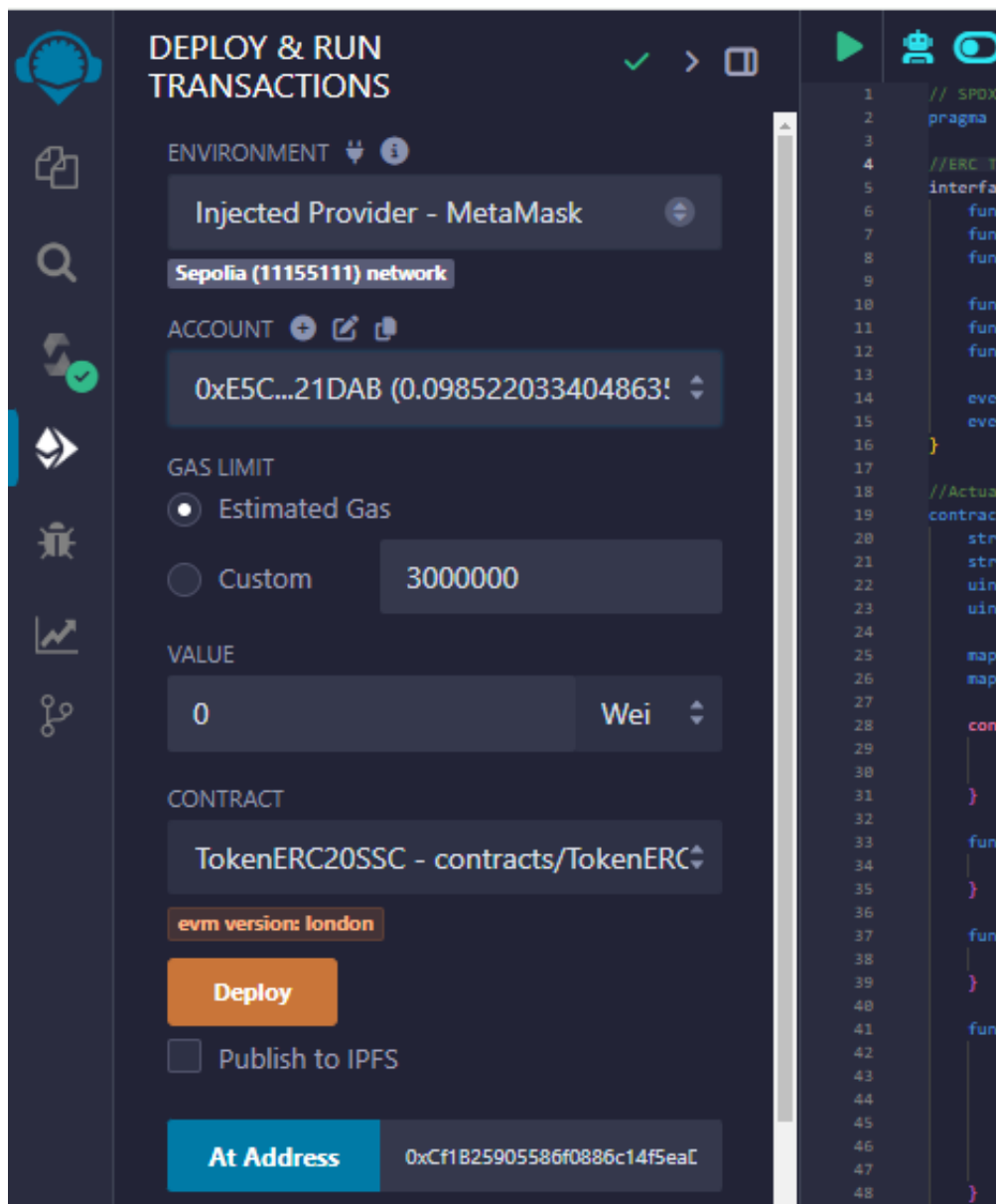
TokenERC20SSC.sol

```
1 // SPDX-License-Identifier: GPL-3.0
2 pragma solidity ^0.8.0;
3
4 //ERC Token Standard #20 Interface
5 interface ERC20Interface{
6     function totalSupply() external view returns (uint);
7     function balanceOf(address tokenOwner) external view returns (uint balance);
8     function allowance(address tokenOwner, address spender) external view returns (uint remaining);
9
10    function transfer(address to, uint tokens) external returns (bool success);
11    function approve(address spender, uint tokens) external returns (bool success);
12    function transferFrom(address from, address to, uint tokens) external returns (bool success);
13
14    event Transfer(address indexed from, address indexed to, uint tokens);
15    event Approval(address indexed tokenOwner, address indexed spender, uint tokens);
16 }
17
18 //Actual token contract
19 contract TokenERC20SSC is ERC20Interface{
20     string public symbol = "SSC";
21     string public name = "SSC Coin";
22     uint8 public decimals = 2;
23     uint256 public _totalSupply;
24
25     mapping(address => uint) balances;
26     mapping(address => mapping(address => uint)) allowed;
27
28     constructor() {
29         _totalSupply = 1000000;
30         balances[msg.sender] = _totalSupply;
31     }
32
33     function totalSupply() public override view returns (uint256) {
34         return _totalSupply;
35     }
36
37     function balanceOf(address tokenOwner) public override view returns (uint256) {
38         return balances[tokenOwner];
39     }
40
41     function transfer(address to, uint tokens) public override returns (bool success) {
42         require(tokens <= balances[msg.sender]);
43
44         balances[msg.sender] = balances[msg.sender] - tokens;
45         balances[to] = balances[to] + tokens;
46         emit Transfer(msg.sender, to, tokens);
47         return true;
48     }
49
50     function approve(address spender, uint256 tokens) public override returns (bool success) {
51         allowed[msg.sender][spender] = tokens;
52         emit Approval(msg.sender, spender, tokens);
53         return true;
54     }
55
56     function allowance(address tokenOwner, address spender) public override view returns (uint remaining) {
57         return allowed[tokenOwner][spender];
58     }
59
60     function transferFrom(address from, address to, uint256 tokens) public override returns (bool success) {
61         require(tokens <= balances[msg.sender]);
62         require(tokens <= allowed[from][msg.sender]);
63
64         balances[from] = balances[from] - tokens;
65         allowed[from][msg.sender] = allowed[from][msg.sender] - tokens;
66         balances[to] = balances[to] + tokens;
67         emit Transfer(from, to, tokens);
68         return true;
69     }
70 }
```

Initialize as git repo

Did you know? You can use the Recorder to record and replay your transactions to save

Contrato programado adaptando o exemplo desenvolvido pelo professor Ricardo. Feitas apenas adaptações de nome do contrato, símbolo e nome do token.

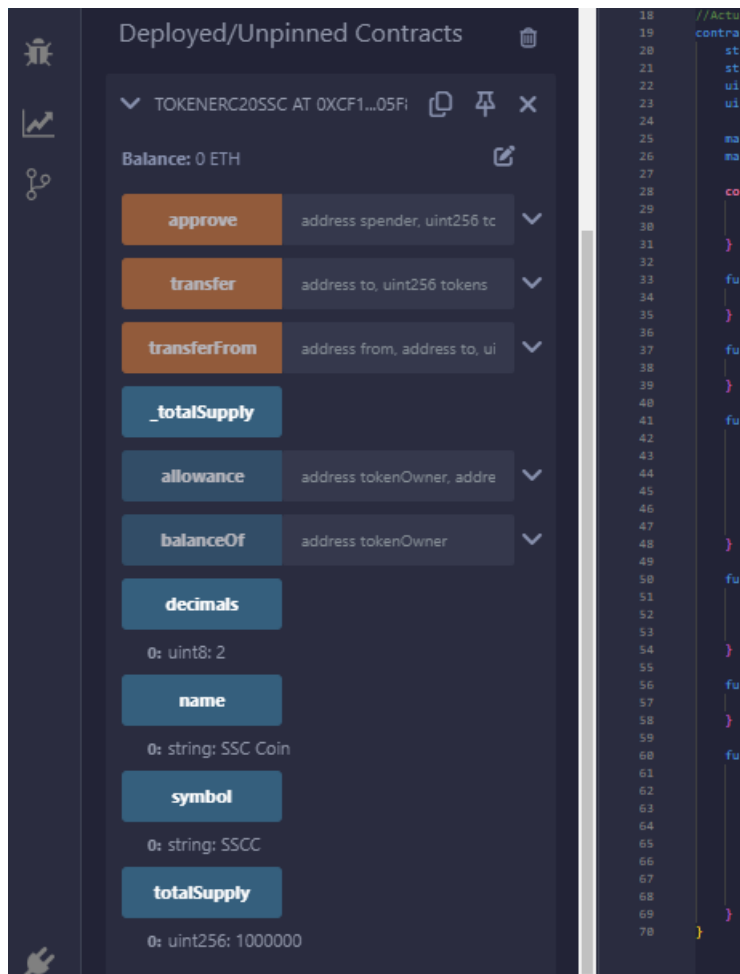


O deploy do contrato foi feito com sucesso para a blockchain Sepolia por intermédio da carteira MetaMask (Injected Provider – MetaMask).

A conta com endereço final 21DAB foi vinculada ao contrato em ACCOUNT para ser usada nas interações com o contrato e pagamento das taxas de transações. Assim, também é a conta owner do contrato.

3º - Interagir com o contrato empregando a Remix IDE

Consultando as variáveis do contrato



Variáveis do contrato conforme esperado, isto é:

Nome do Token → SSC Coin;

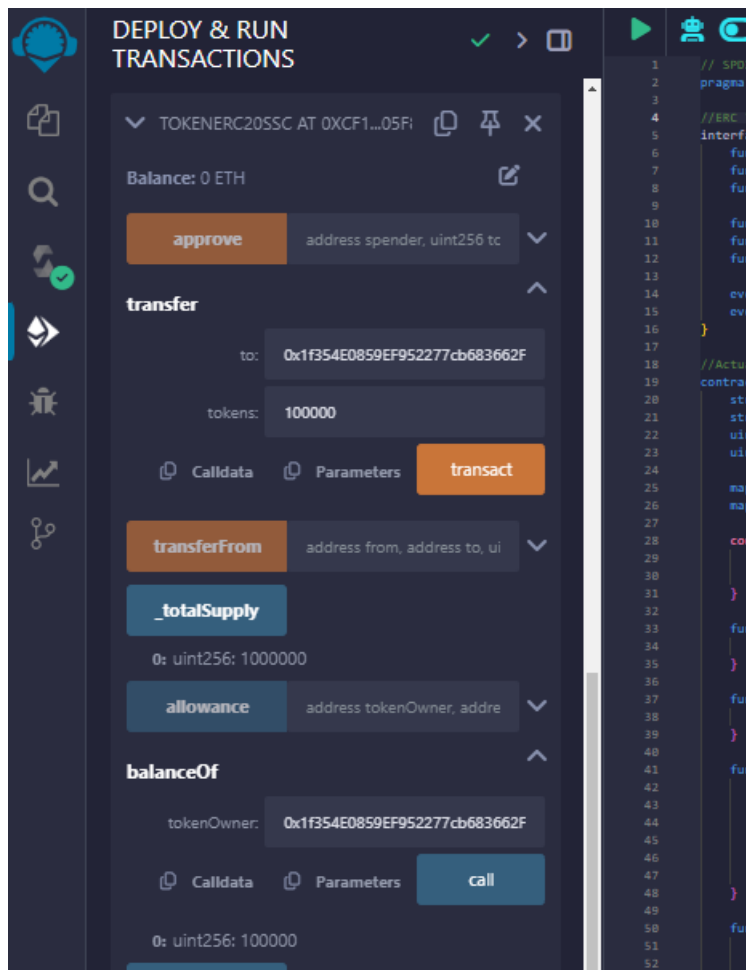
Símbolo do Token → SSCC;

Casas decimais → 2;

Total Suplay → 1.000.000

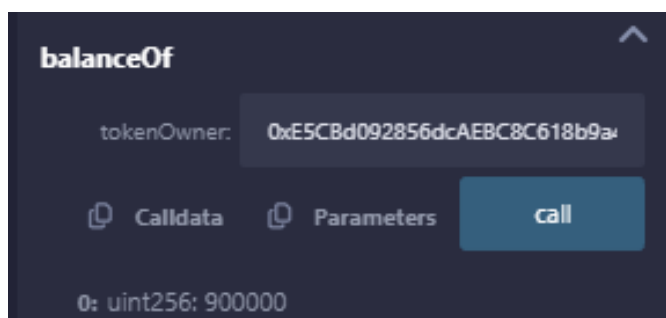
OBS.: Para fins deste contrato, adotou-se o endereço da primeira conta, final 46110 como Owner e o segundo endereço/conta, final 63513, como delegado

Chamada da função transfer



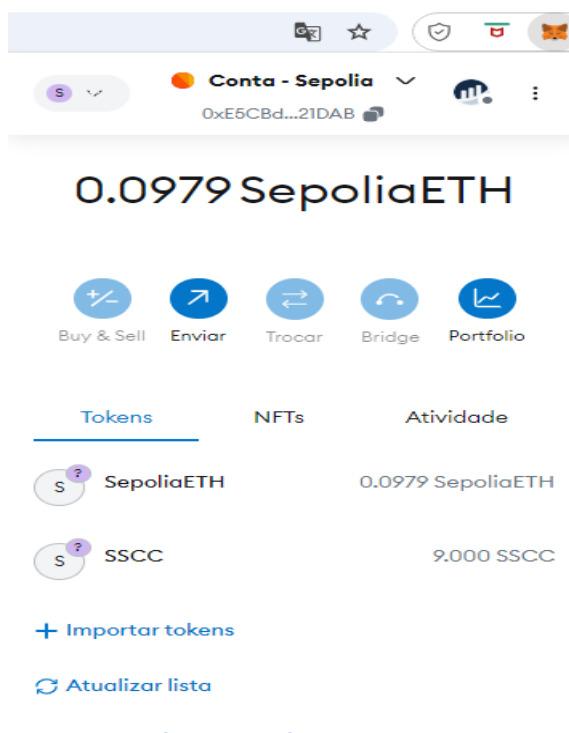
Transferiu 100.000 SSCC da **Conta Sepolia** para a **ProjetoTokenERC20 - Conta 01**, endereço final 46110

Consultando saldo da conta owner em SSCC após a transferência



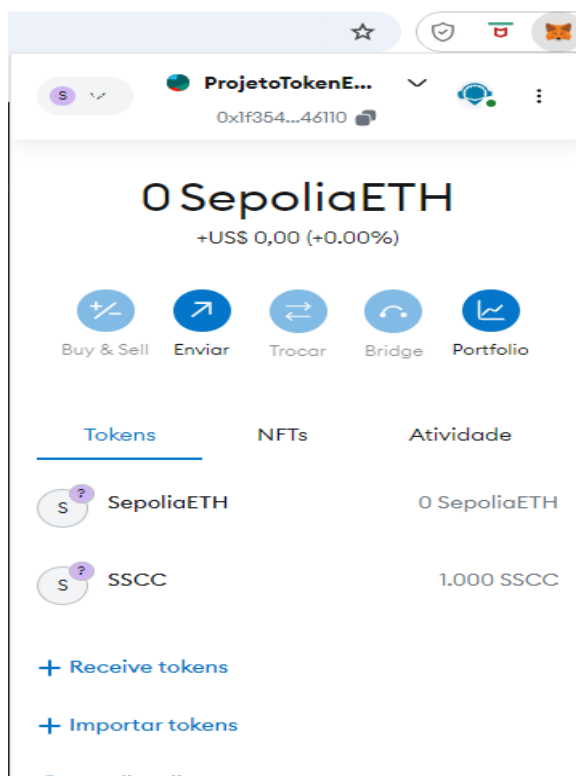
Saldo reduzido para 900.000 SSCC, conforme esperado.

Consultando saldo da conta owner do contrato na carteira MetaMask




OBS.: Na MetaMask o saldo em SSCC é de 9 mil, diferente dos 900 mil. Suponho que seja em função das duas decimais do token.




Consultando saldo da conta ProjetoTokenERC20 - Conta 01, destino da transferência



OBS.: Na MetaMask o saldo em SSCC é de 1 mil, diferente dos 100 mil. Suponho que seja em função das duas decimais do token.

Consultando as transações para a conta owner, Conta – Sepolia,

 Home Blockchain Tokens NFTs Misc

 Address 0xE5CBd092856dcAEBc8C618b9a4579C53ba121DAB  

Overview

ETH BALANCE
0.097886961958995956 ETH

TOKEN HOLDINGS
\$0.00 (1 Tokens)

More Info

TRANSACTIONS SENT
Latest: 30 mins ago First: 17 hrs ago



FUNDED BY
0x4281eCF0...30f3084eE at txn 0x71e31bcb9...
















Multichain Info


N/A


Transactions Token Transfers (ERC-20)



Latest 3 from a total of 3 transactions

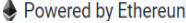

 Download Page Data 

Transaction Hash	Method	Block	Age	From	To	Amount	Txn Fee
 0x5559b9e8de... 	Transfer	6704375	30 mins ago	0xE5CBd092...3ba121DAB 	 0xCf1B2590...918605F8A 	0 ETH	0.00063507
 0x3e0d1a6ae4... 	0x60806040	6699895	17 hrs ago	0xE5CBd092...3ba121DAB 	 Contract Creation 	0 ETH	0.00147796
 0x71e31bcb9... 	Transfer	6698543	22 hrs ago	0x4281eCF0...30f3084eE 	 0xE5CBd092...3ba121DAB 	0.1 ETH	0.00000496

[Download: CSV Export ]

 A wallet address is a publicly available address that allows its owner to receive funds from another party. To access the funds in an address, you must have its private key. Learn more about addresses in our [Knowledge Base](#).

 This website uses cookies to improve your experience. By continuing to use this website, you agree to its [Terms](#) and [Privacy Policy](#). 

  Back to Top

Foram três transações para obter faucets (0.1 ETH), criação do contrato e chamada do contrato para a transferência de tokens SSCC para o endereço de final 6110.

Sobre mim

Eu sou Sérgio Santa Catarina, servidor público aposentado após longa carreira em TI, começando como programador em ambientes mainframe IBM, linguagem COBOL/CICS, e passando por diversas outras tecnologias e funções/cargos, como analista de suporte, administrador de banco de dados, analista de negócios e, por fim, gerente de projetos. Antes de me aposentar ainda trabalhei 6 anos com gestão de processos e planejamento estratégico. Se tem uma linguagem que ainda me atrevo a codificar algo é SQL.

Hoje busco me atualizar nas tecnologias atuais com ênfase em blockchain e demais que dão suporte às criptomoedas. Não faço ideia se vou trabalhar com isso. No momento busco aprender.

Conecte-se comigo

Perfil DIO; https://www.dio.me/users/sergio_santacatarina)

E-mail: (<mailto:sergio.santacatarina@gmail.com>)

LinkedIn: (www.linkedin.com/in/sérgio-santa-catarina-95b58273/)

[Instagram (<https://www.instagram.com/sergiosantacatarina/>)

Apêndice

Apenas lembrando que sou iniciante nestas tecnologias.

Autores

- [[@SergioSantaCatarina](https://www.github.com/SergioSantaCatarina)](<https://www.github.com/SergioSantaCatarina>)