Desafio de Projeto - Bootcamp Blockchain Developer

Criando a sua primeira criptomoeda na rede Ethereum

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

Consiste basicamente das seguintes atividades:

- Instalar o Ganache - ambiente de blockchain local para deploy e execução do contrato;

- Instalar o plugin Metamask no google chrome e criar algumas contas associadas ao ambiente(rede) local Ganache para controlar a execução do contrato;

- Programar, compilar, fazer o deploy do contrato empregando a Remix IDE;

- interagir com o contrato empregando a Remix IDE;

- Consultar os blocos e as transações empregando a interface do Ganache.

Tecnologias envolvidas

- Ganache-CLI (https://archive.trufflesuite.com/ganache/)

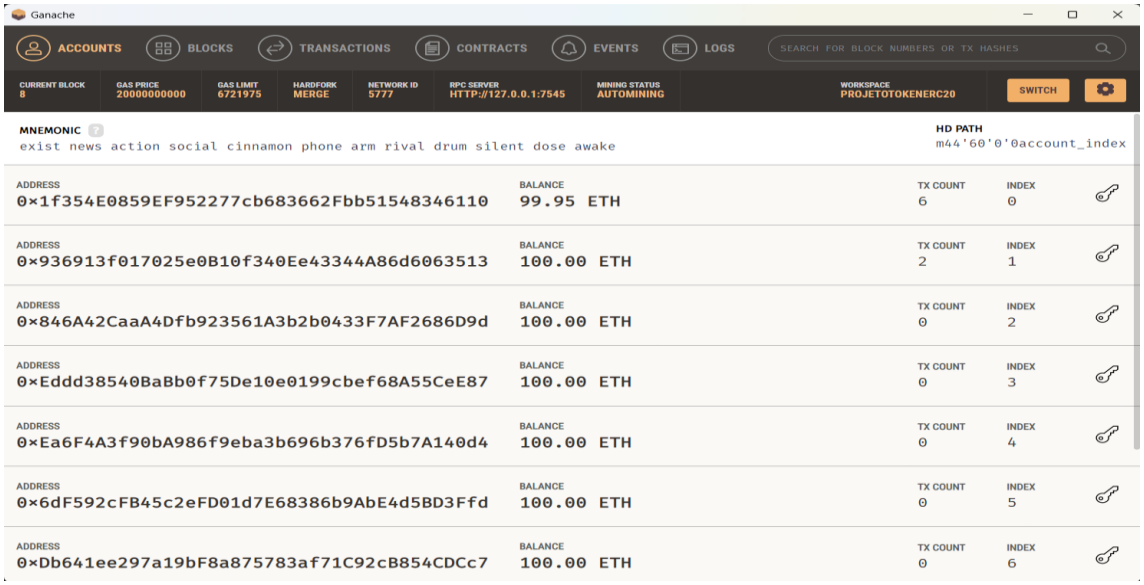
- MetaMask (https://metamask.io/download/)

- Remix IDE (https://remix.ethereum.org/)

- Solidity – Linguagem de programação (https://soliditylang.org/)

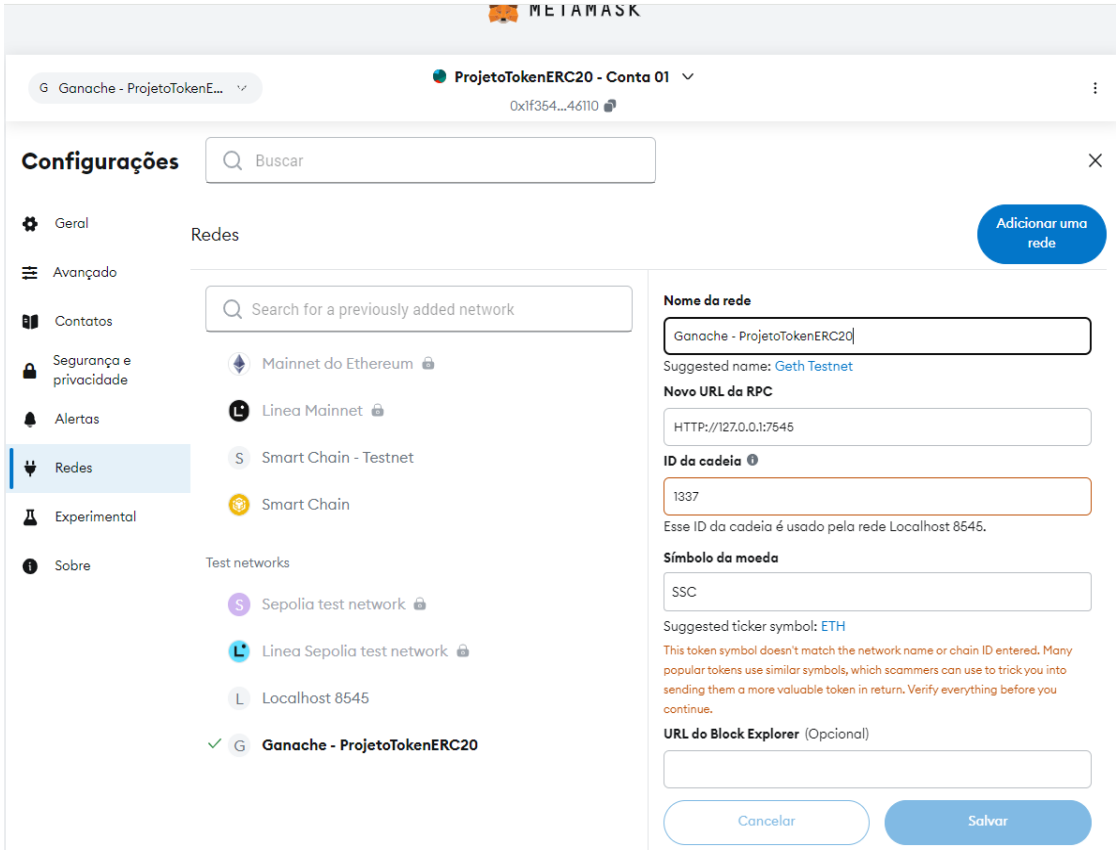
O passo a passo para concluir o desafio

1º - Instalação do ambiente Ganache

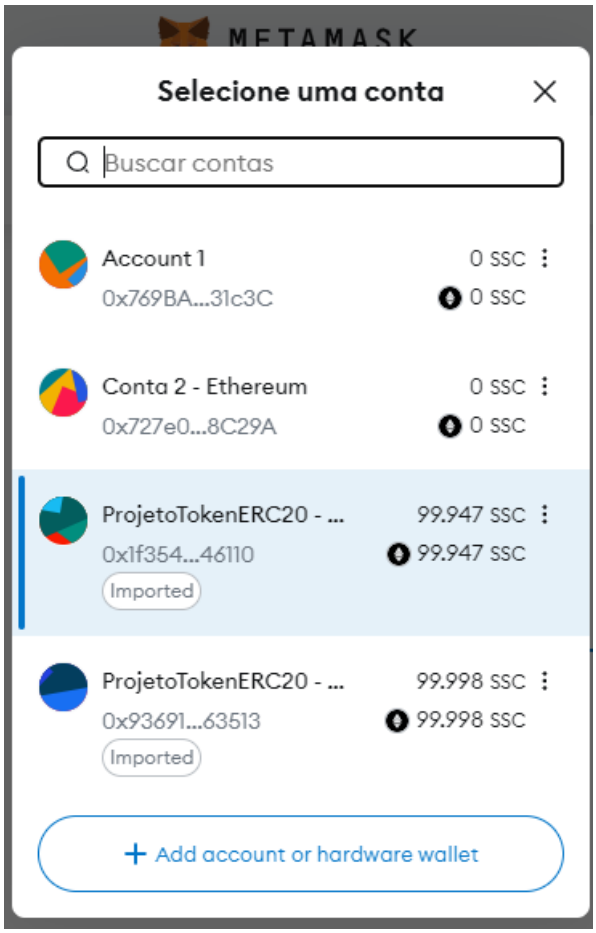


A cada workspace criado, são adicionados automaticamente 10 endereços de carteiras para a operação na rede blockchain.

2º - Instalar o MetaMask no google chrome e criar algumas contas



Plugin instalado. Adicionada a rede Ganache – ProjetoTokenERC20, vinculada ao servidor ganache HTTP://127.0.0.1:7545

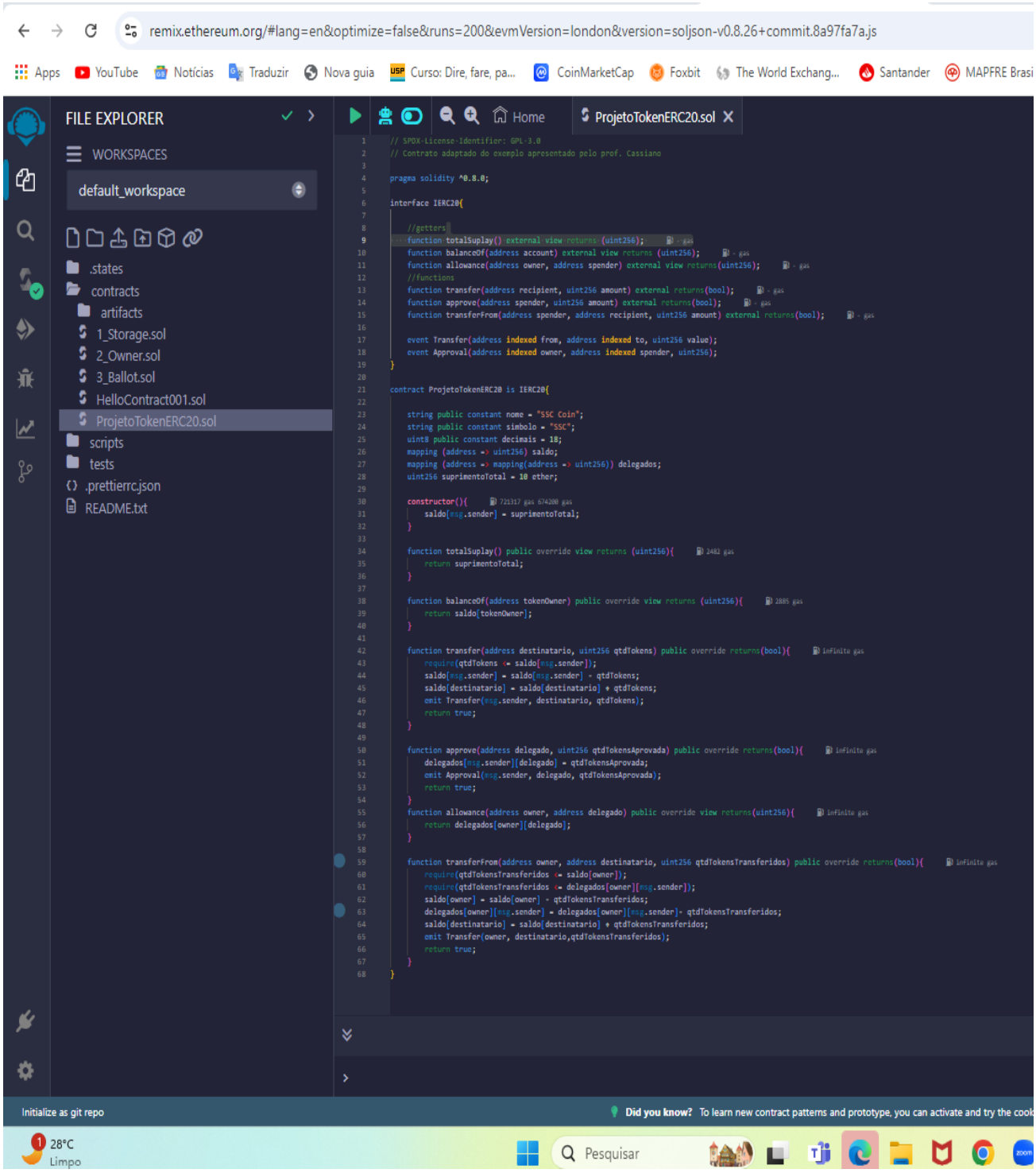


Criadas as contas ProjetoTokenERC20 – Conta 01 e ProjetoTokenERC20 – Conta 02.

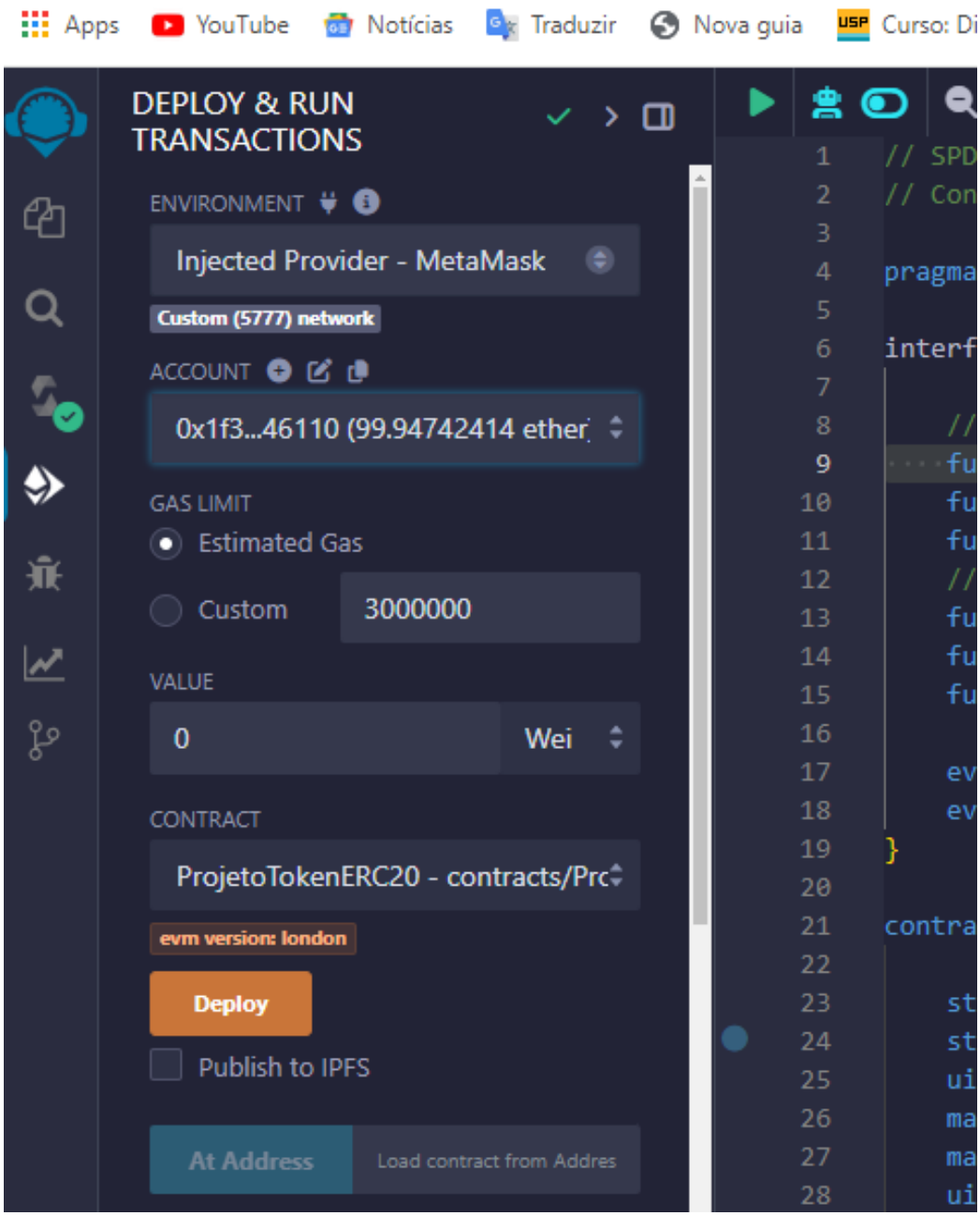
Ambas as contas criadas por importação dos dois primeiros endereços gerados pelo Ganache, copiando a chave privada de cada endereço lá no Ganache e colando na tela de importação da conta.

Essas contas serão utilizadas para as operações do contrato e foram vinculadas à Remix IDE para tanto.

3º - Programar, compilar e fazer o deploy do contrato com a Remix IDE



Contrato programado adaptando o exemplo desenvolvido pelo professor Cassiano. Feitas apenas adaptações de nome de variáveis, funções e eventos, além do nome do contrato.

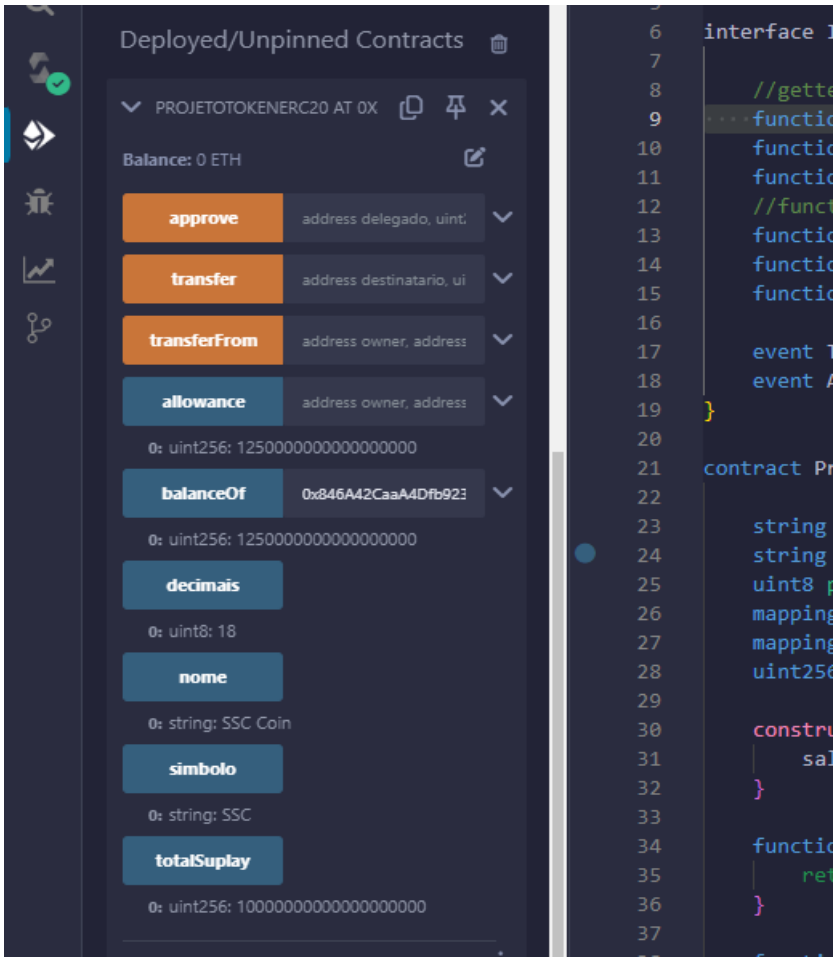


O deploy do contrato foi feito com sucesso para a blockchain local do Ganache por intermédio da carteira MetaMask (Injected Provider – MetaMask).

As duas contas (correspondentes aos dois primeiro endereços, finais 46110 e 63513, da blockchain Ganache) foram vinculados ao contrato em ACCOUNT para serem usados nas interações com o contrato.

4º - Interagir com o contrato empregando a Remix IDE

Consultando as variáveis do contrato

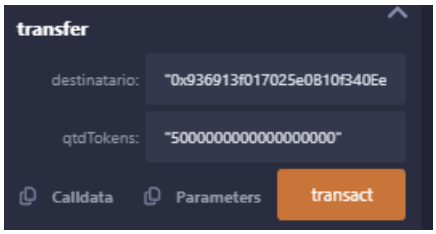


Variáveis do contrato conforme esperado, isto é:

Nome do Token ➔ SSC Coin;
Simbolo do Token ➔ SSC;
Casas decimais ➔ 18;
Total Suplay ➔ 10000000000000000000

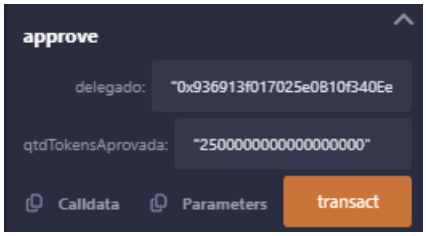
OBS.: Para fins deste contrato, adotou-se o endereço da primeira conta, final 46110 como Owner e o segundo endereço/conta, final 63513, como delegado

Chamada da função transfer



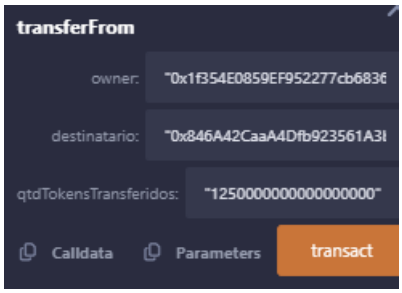
Transferiu metade do suprimento total, da primeira carteira (owner) para a segunda (delegado)

Chamada da função approve



O owner autoriza/delega para o delegado a permissão para gastar ¼ do saldo do owner.

Chamada da função transferFrom



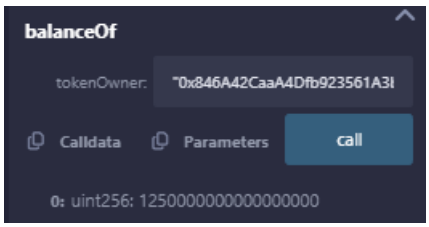
O delegado transfere (paga para) um terceiro endereço, o terceiro gerado pelo ganache, a quantia correspondente à metade do que lhes foi autorizado pelo owner. Esta quantia sai da conta do owner e não do saldo do delegado.

Obs: Para cada transação executada, a carteira da metamask é acionada na conta correspondente para autorizar a transação.

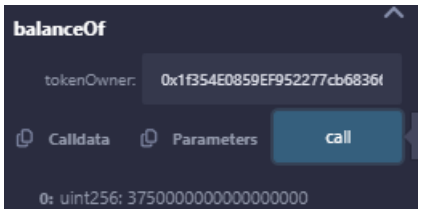
Obs.: O terceiro endereço do Ganache utilizado como destinatário na função transferFrom não precisa ter conta na MetaMask, pois apenas recebe, não necessitando, portanto autorizar nenhuma transação.

Consultando saldos

Consultando saldo do terceiro endereço, destinatário do transferFrom

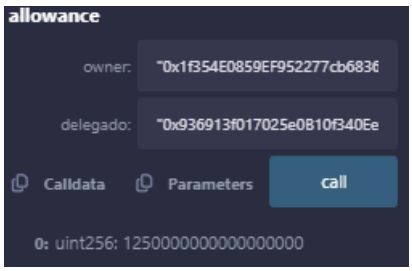


Consultando saldo final do owner, após todas as operações realizadas



Saldo OK, restante do total suply (10) menos 50%(5) transferidos para o segundo endereço, menos 12,5%(1,25) gastos pelo delegado do saldo do owner.

Consultando saldo aprovado restante – autorizado pelo owner ao delegado



Saldo OK, corresponde à metade do que foi autorizado.

Log das transações no terminal do Remix IDE

Welcome to Remix 0.53.1

Your files are stored in indexedDB, 5.14 MB / 558.16 GB used

You can use this terminal to:

- Check transactions details and start debugging.
- Execute JavaScript scripts:
  - Input a script directly in the command line interface
  - Select a Javascript file in the file explorer and then run `remix.execute()` or `remix.exeCurrent()` in the command line interface
  - Right click on a JavaScript file in the file explorer and then click `Run`

The following libraries are accessible:

- web3.js
- ethers.js
- sol-gpt *<your Solidity question here>*

Type the library name to see available commands.

Solidity copilot not activated!

creation of ProjetoTokenERC20 pending...
[block:5 txIndex:-]
from: *0x1f3...46110* *// owner do contrato*
to: *ProjetoTokenERC20.(constructor)*
value: *0 wei*
data: *0x608...a0033*
logs: *0*
hash: *0x92c...4c53c*

call to ProjetoTokenERC20.balanceOf
call[call]
from: *0x936913f017025e0B10f340Ee43344A86d6063513*
to: *ProjetoTokenERC20.balanceOf(address) // Consulta saldo do segundo endereço*
data: *0x70a...63513*

call to ProjetoTokenERC20.balanceOf
call[call]
from: *0x1f354E0859EF952277cb683662Fbb51548346110*
to: *ProjetoTokenERC20.balanceOf(address) // Consulta saldo do owner*
data: *0x70a...46110*

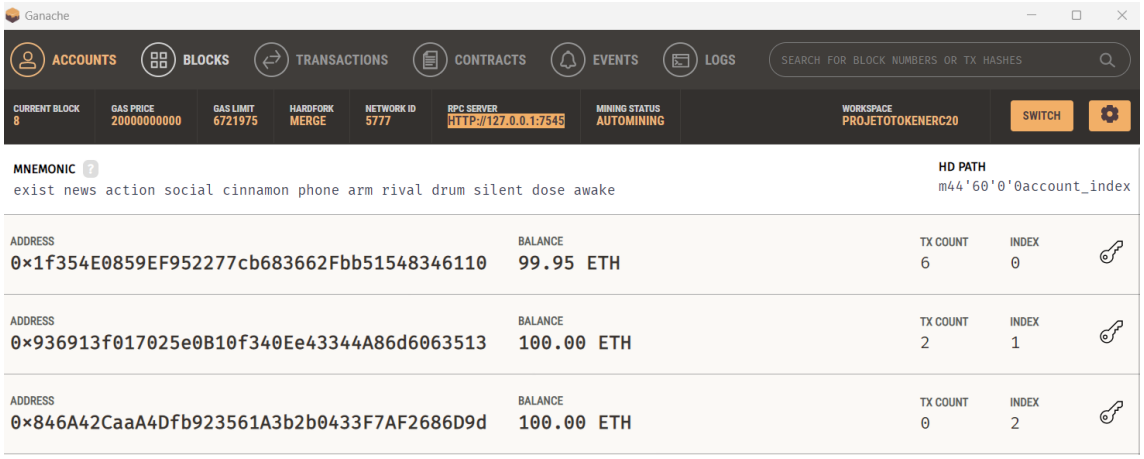
transact to ProjetoTokenERC20.transfer pending ...
[block:6 txIndex:-]
from: *0x1f3...46110*
to: *ProjetoTokenERC20.transfer(address,uint256) 0x443...c0e67 // owner transfere para endereço 2.*
value: *0 wei*
data: *0xa90...40000*
logs: *1*
hash: *0xcf9...a5c64*

call to ProjetoTokenERC20.balanceOf
call[call]
from: *0x1f354E0859EF952277cb683662Fbb51548346110*
to: *ProjetoTokenERC20.balanceOf(address)*
data: *0x70a...46110*
Debug
call to ProjetoTokenERC20.balanceOf
call[call]
from: *0x936913f017025e0B10f340Ee43344A86d6063513*
to: *ProjetoTokenERC20.balanceOf(address)*
data: *0x70a...63513*
Debug
call to ProjetoTokenERC20.balanceOf
call[call]
from: *0x936913f017025e0B10f340Ee43344A86d6063513*
to: *ProjetoTokenERC20.balanceOf(address)*
data: *0x70a...63513*
Debug
call to ProjetoTokenERC20.allowance
call[call]
from: *0x1f354E0859EF952277cb683662Fbb51548346110*
to: *ProjetoTokenERC20.allowance(address,address)*
data: *0xdd6...63513*
Debug
transact to ProjetoTokenERC20.approve pending ...
[block:7 txIndex:-]
from: *0x1f3...46110*
to: *ProjetoTokenERC20.approve(address,uint256) 0x443...c0e67*
value: *0 wei*
data: *0x095...a0000*
logs: *1*
hash: *0xe27...5be39*
Debug
call to ProjetoTokenERC20.allowance
call[call]
from: *0x1f354E0859EF952277cb683662Fbb51548346110*

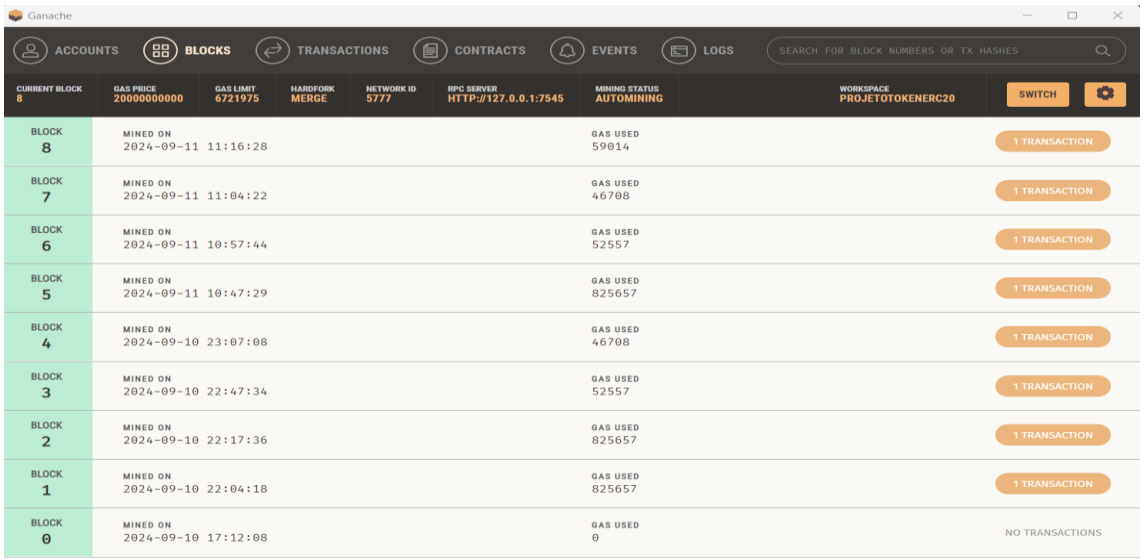
to: *ProjetoTokenERC20.allowance(address,address)*
data: *0xdd6...63513*
Debug
call to ProjetoTokenERC20.balanceOf
call[call]
from: *0x1f354E0859EF952277cb683662Fbb51548346110*
to: *ProjetoTokenERC20.balanceOf(address)*
data: *0x70a...63513*
Debug
call to ProjetoTokenERC20.balanceOf
call[call]
from: *0x1f354E0859EF952277cb683662Fbb51548346110*
to: *ProjetoTokenERC20.balanceOf(address)*
data: *0x70a...46110*
Debug
transact to ProjetoTokenERC20.transferFrom pending ...
[block:8 txIndex:-]
from: *0x936...63513*
to: *ProjetoTokenERC20.transferFrom(address,address,uint256) 0x443...c0e67*
value: *0 wei*
data: *0x23b...d0000*
logs: *1*
hash: *0x9db...a49f9*
Debug
call to ProjetoTokenERC20.allowance
call[call]
from: *0x936913f017025e0B10f340Ee43344A86d6063513*
to: *ProjetoTokenERC20.allowance(address,address)*
data: *0xdd6...63513*
Debug
call to ProjetoTokenERC20.balanceOf
call[call]
from: *0x936913f017025e0B10f340Ee43344A86d6063513*
to: *ProjetoTokenERC20.balanceOf(address)*
data: *0x70a...46110*
Debug
call to ProjetoTokenERC20.balanceOf
call[call]
from: *0x936913f017025e0B10f340Ee43344A86d6063513*
to: *ProjetoTokenERC20.balanceOf(address)*
data: *0x70a...63513*
Debug
call to ProjetoTokenERC20.balanceOf
call[call]
from: *0x936913f017025e0B10f340Ee43344A86d6063513*
to: *ProjetoTokenERC20.balanceOf(address)*
data: *0x70a...86d9d*
Debug
call to ProjetoTokenERC20.balanceOf
call[call]
from: *0x936913f017025e0B10f340Ee43344A86d6063513*
to: *ProjetoTokenERC20.balanceOf(address)*
data: *0x70a...86d9d*
Debug
call to ProjetoTokenERC20.totalSuplay
call[call]
from: *0x1f354E0859EF952277cb683662Fbb51548346110*
to: *ProjetoTokenERC20.totalSuplay()*
data: *0x16f...a99be*
Debug
call to ProjetoTokenERC20.simbolo
call[call]
from: *0x1f354E0859EF952277cb683662Fbb51548346110*
to: *ProjetoTokenERC20.simbolo()*
data: *0x0de...82bae*
Debug

call to ProjetoTokenERC20.nome
call[call]
from: *0x1f354E0859EF952277cb683662Fbb51548346110*
to: *ProjetoTokenERC20.nome()*
data: *0x2de...b124b*
Debug
call to ProjetoTokenERC20.decimais
call[call]
from: *0x1f354E0859EF952277cb683662Fbb51548346110*
to: *ProjetoTokenERC20.decimais()*
data: *0x45b...0a611*
Debug
call to ProjetoTokenERC20.balanceOf
call[call]
from: *0x1f354E0859EF952277cb683662Fbb51548346110*
to: *ProjetoTokenERC20.balanceOf(address)*
data: *0x70a...46110*
Debug
>

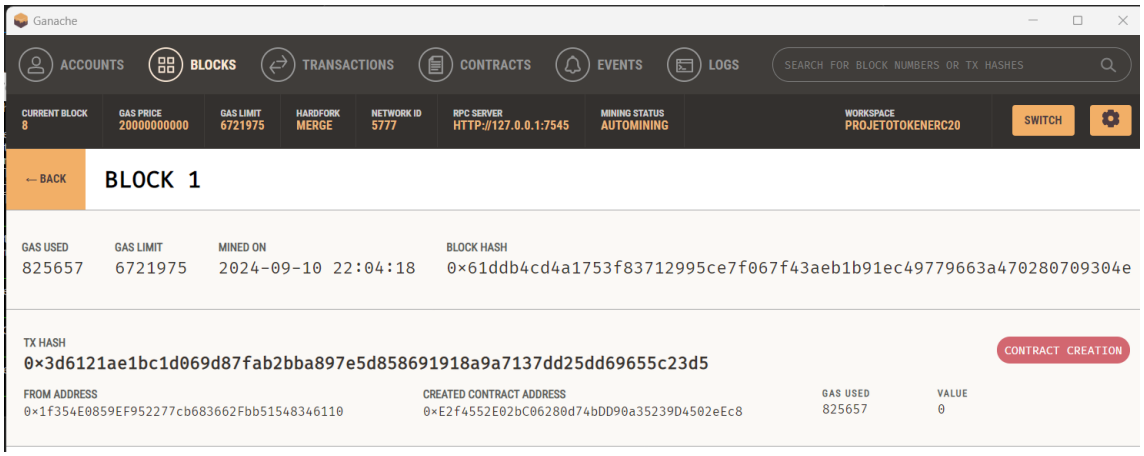
5º - Consultando blocos e transações no Ganache



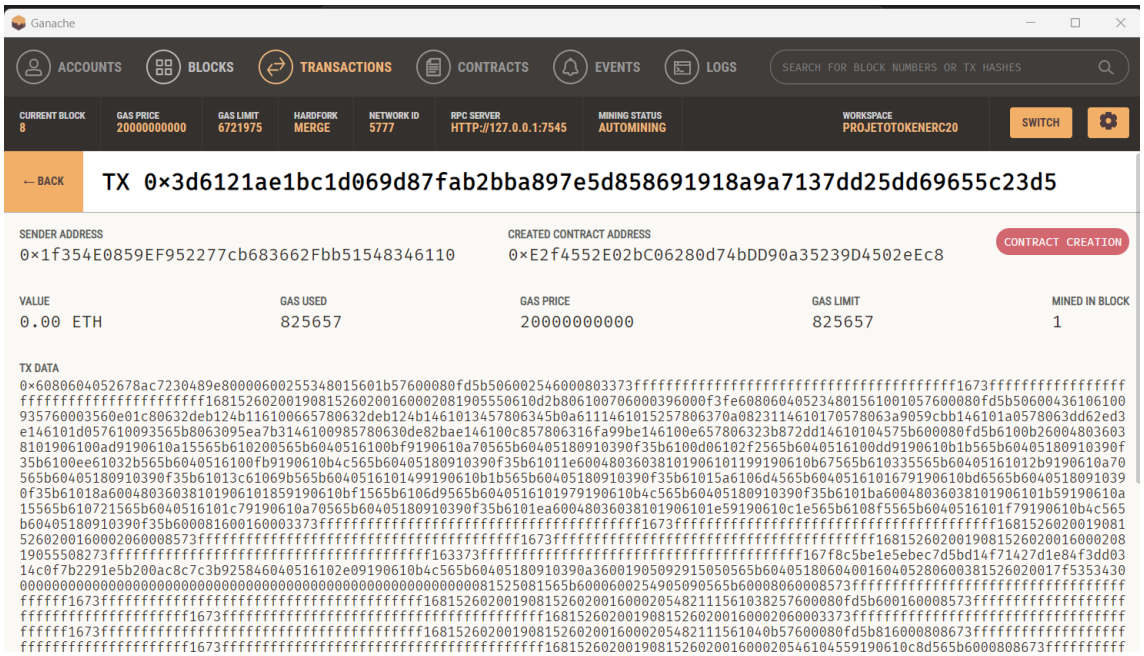
Transações executadas por cada um dos três primeiros endereços.



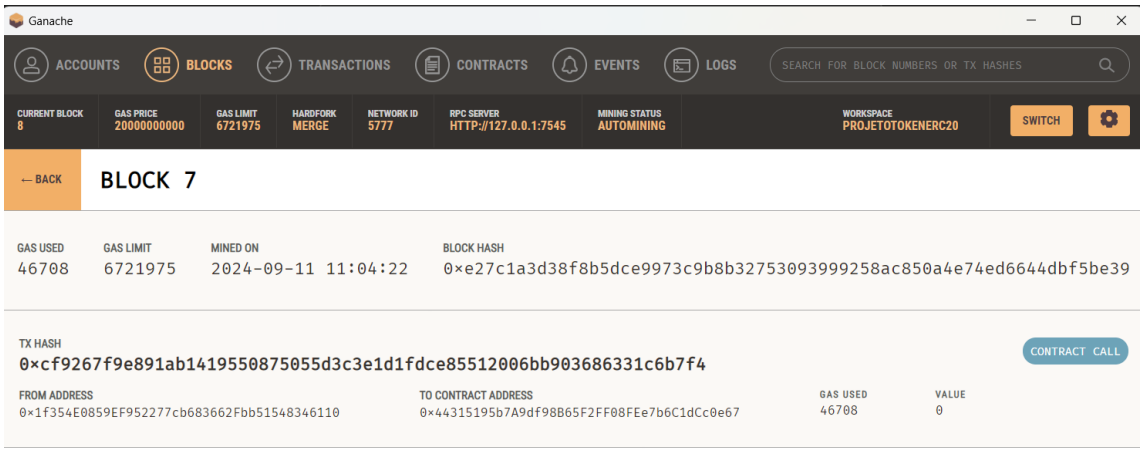
Total de blocos gerados



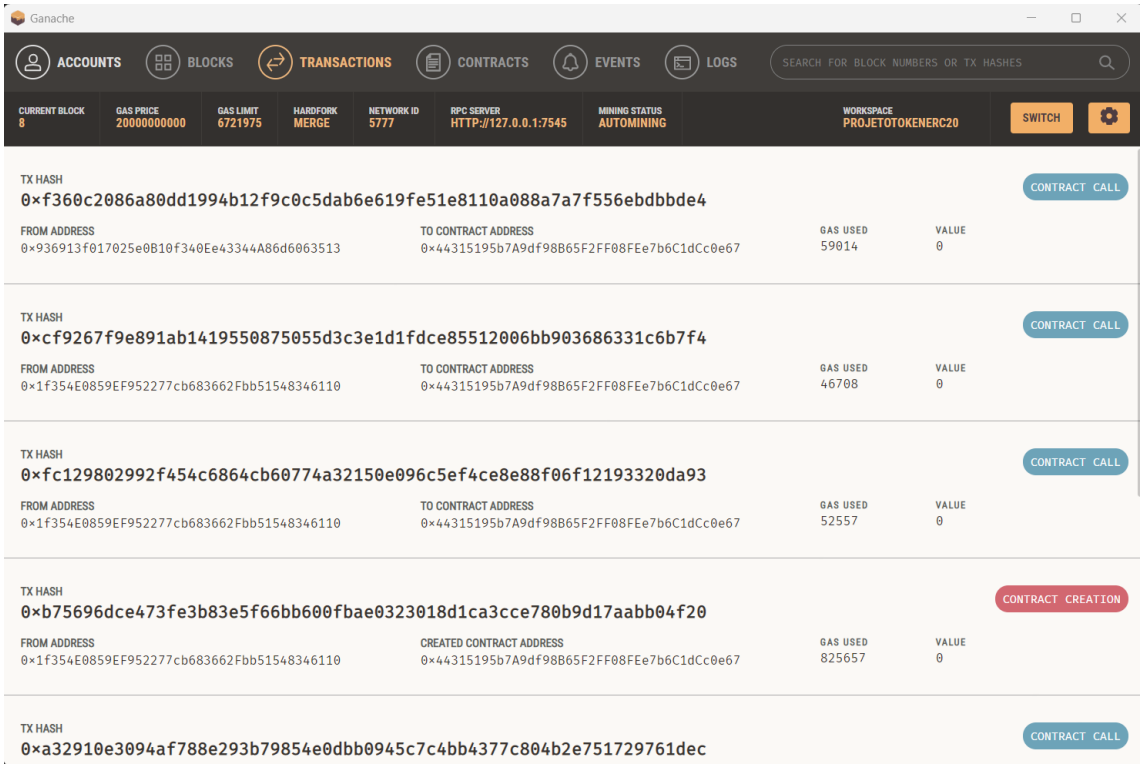
Bloco 1 – Criador do contrato



Detalhes do contrato



Bloco de chamada do contrato



Algumas das transações

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)