



Equipe Tesouro Em Bytes

# TransfereChain

## Pitch Deck

### Desafio 3



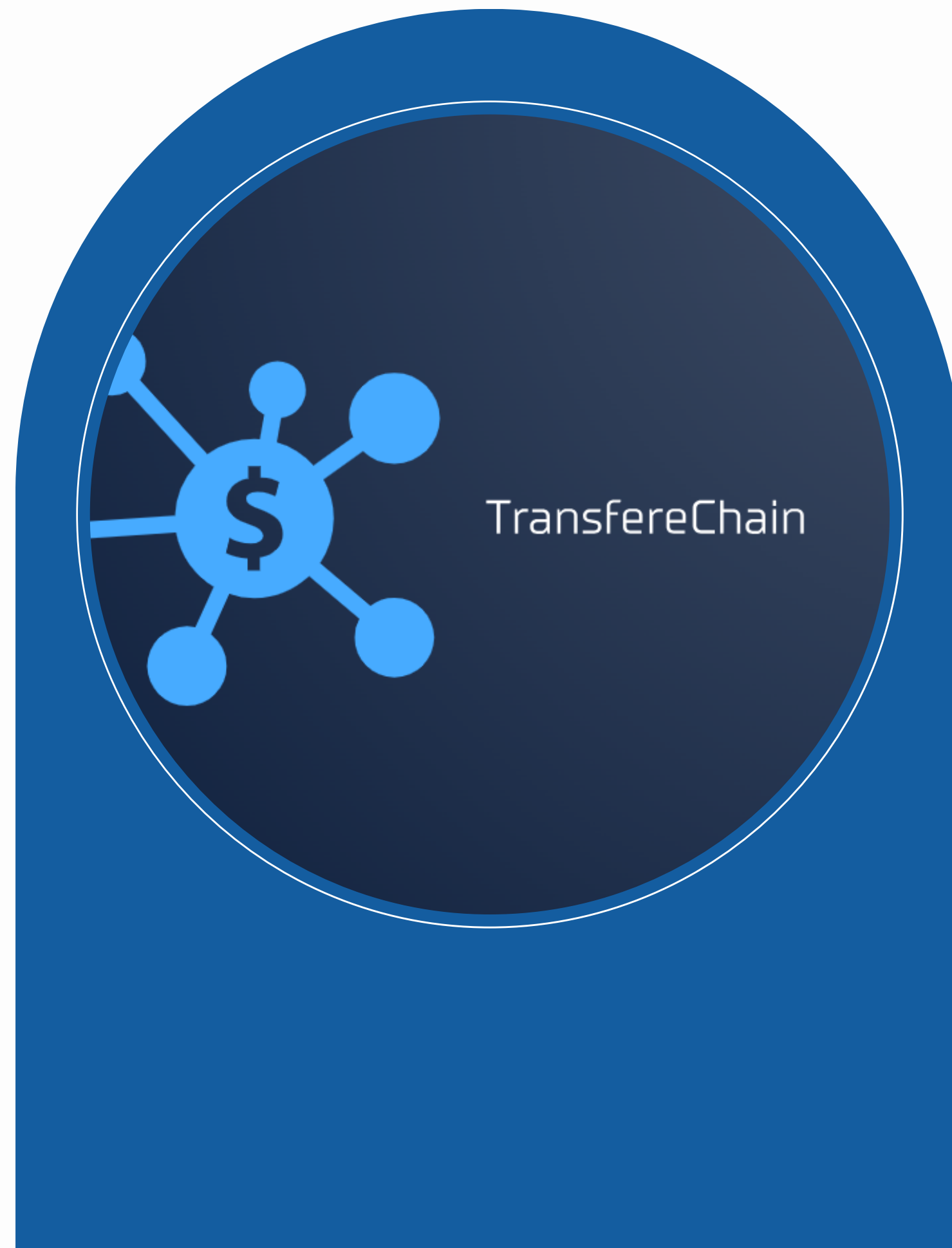
TransfereChain

# Solução TransfereChain

Sistema que utiliza a blockchain para efetuar transferências especiais para beneficiários (Municípios, Estados e Distrito Federal) com o DREX e aplicando condicionantes de repasse.

## Público Alvo

- Parlamentares
- Governo Brasileiro
- Cidadão





# Introdução

- Emenda Constitucional 105/2019: incertezas ao processo de execução – passa a ser do ente recebedor o poder de aplicação dos recursos transferidos.
- Para o parlamentar, ganha-se em celeridade mas perde-se em autonomia decisória.

## Problema 01

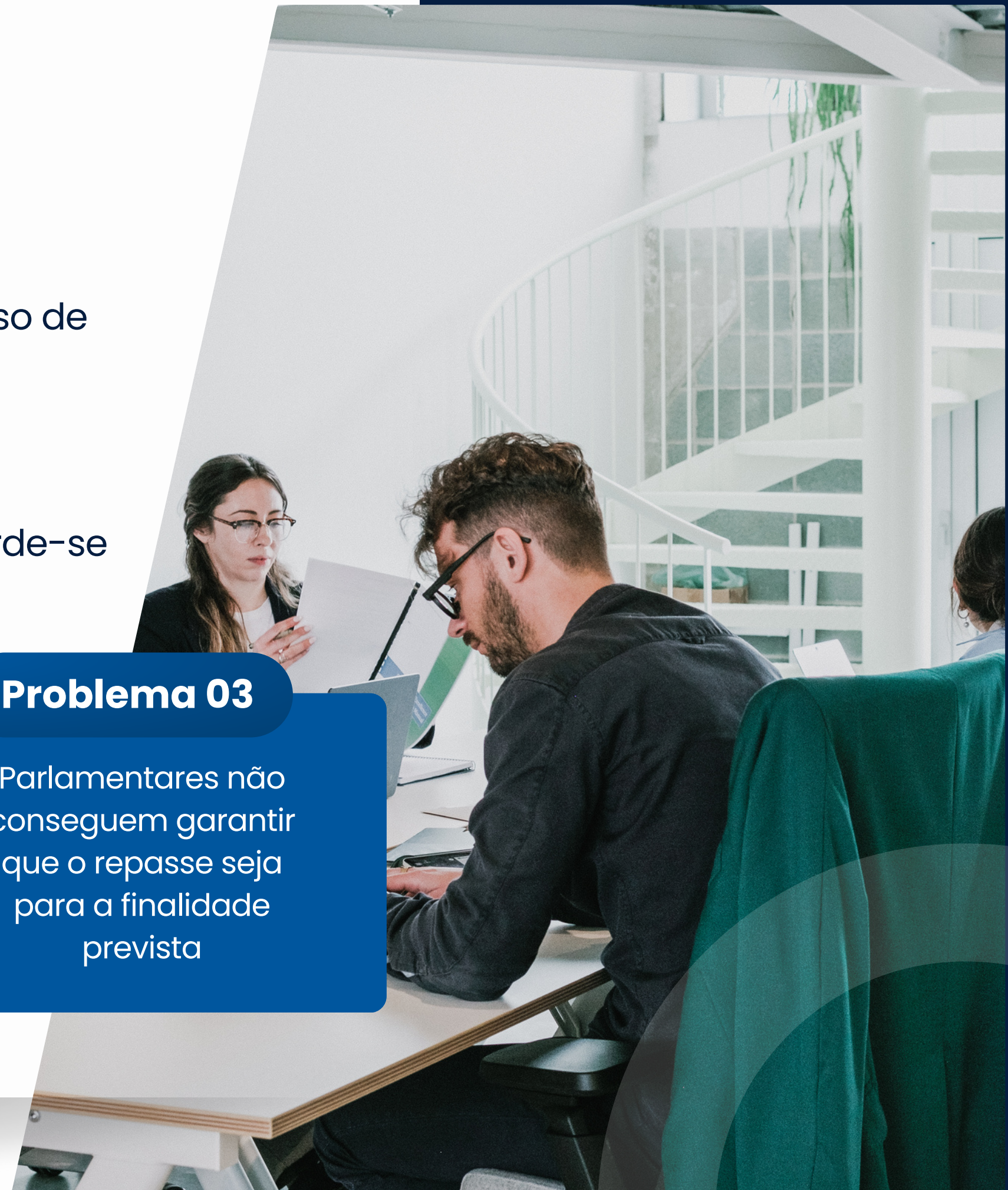
Para o governo, não há transparência na destinação de recursos da transferência especial

## Problema 02

O cidadão não tem como ver onde foi destinado o recurso repassado para a sua cidade/Estado

## Problema 03

Parlamentares não conseguem garantir que o repasse seja para a finalidade prevista





# Perfil TransfereChain: Governo

Construímos a solução demonstrando as ações de quatro perfis e suas ações:

## 1. Perfil Administrador do sistema

- Cadastrar parlamentares
- Cadastra beneficiários (Estados, Municípios e Distrito Federal)

## 2. Perfil parlamentar

- Visualiza emendas/programas e os beneficiários de cada programa
- Seleciona beneficiários

## 3. Perfil do beneficiário (Estados, Municípios e Distrito Federal)

- Dar ciência nas emendas disponíveis para o beneficiário
- Conectar a carteira do beneficiário

## 4. Perfil de acesso público a dados de portal da transparência

- Consultar os contratos para verificar todas as transferências especiais efetuadas.  
Tendo acesso a dados do beneficiário e documentos enviados

# Integração



Localhost  
Funciona !



Falha em  
produção

As variáveis de ambiente não foram para o ambiente  
de produção..

# Rodando script de criação do contrato e demonstrando carteira vazia do beneficiário

The image shows a Visual Studio Code editor with a project named 'TRANSFERECHAIN'. The file explorer on the left shows the project structure, including a 'script' directory containing 'DeployTransfereChain.s.sol'. The main editor displays the content of 'DeployTransfereChain.s.sol', which is a Solidity script for deploying a contract on the Sepolia network. The script includes imports for 'Script', 'TransfereChain', 'RealTokenizado', and 'RealDigital'. It contains a comment in Portuguese explaining the script's purpose: to create a contract, enable it, and simulate a transfer request. The script defines a 'DeployTransfereChain' contract with a 'run' function that sets up the contract and performs a broadcast.

```
1 // SPDX-License-Identifier: MIT
2
3 pragma solidity ^0.8.13 .0;
4
5 import {Script, console} from 'forge-std/Script.sol';
6 import {TransfereChain} from '../src/TransfereChain.sol';
7 import {RealTokenizado} from '../src/RealTokenizado.sol';
8 import {RealDigital} from '../src/RealDigital.sol';
9
10 /*
11 Script para criar o contrato na rede Sepolia, habilitá-lo pela autoridade do Real Digital a
12 transacionar, então habilitá-lo pelo administrador do Real Digital a mover Real Digital e imprimir
13 Real Tokenizado de um determinado banco.
14 Depois disso o script simula a criação de uma requisição de transferência por parte do parlamentar e
15 o aceite da 1ª parcela por parte do recebedor
16 */
17
18 contract DeployTransfereChain is Script {
19     function run() external returns (TransfereChain) {
20         RealDigital realDigital = RealDigital(0x740bc1AFefc3EF48BA06aCA16901565af4d9Fa83);
21         RealTokenizado contaDoTesouro = RealTokenizado(0x1AaDB34eE8fD0383A091EbeB0802e3c3d638C26d);
22         RealTokenizado bancoRecebedor = RealTokenizado(0x65A711ae4B00b49A25528FE4883054080BC80F6D);
23         address contaDoRecebedor = 0x45f2cB0F09AB00A4c62e7EC38487c464Ea59833E;
24         address carteiraDoParlamentar = 0xA104505CCeddC6aBD75d713A6af5a868CF82337f;
25         bytes32 MINTER_ROLE = 0x9f2df0fed2c77648de5860a4cc508cd0818c85b8b8a1ab4ceef8d981c8956a6;
26         bytes32 MOVER_ROLE = 0xe5ed70e23144309ce456cb48bf5e6d0d8e160f094a6d65ecf1d5b03cf292d8e6;
27
28         uint256 deployerPrivateKey = vm.envUint('PRIVATE_KEY');
29         vm.startBroadcast(deployerPrivateKey);
30         TransfereChain transfereChain = new TransfereChain(contaDoTesouro, realDigital, msg.sender);
31         vm.stopBroadcast();
32     }
33 }
```

The terminal window at the bottom shows the command prompt 'marcelofrayha@192 TransfereChain %'.

# Rodando script de criação e demonstrando carteira vazia do beneficiário

Sepolia Testnet

Search by Address

Overview Logs (4) State

[ This is a Sepolia Testnet transaction only ]

Transaction Hash: 0x829249dd8e69a9838c82906c67e88164f66445bcf9529f1b85ee61...

Status: Success

Block: 4828555 4 Block Confirmations

Timestamp: 56 secs ago (Dec-05-2023 05:43:24 PM +UTC)

Method: Executa Transferencia

From: 0x45f2cB0F09AB00A4c62e7EC38487c464Ea59833E

Interacted With (To): 0x2a6c006B1317e2E859f1Cf0F0685e69B30F3D59f ✓

ERC-20 Tokens Transferred: 2

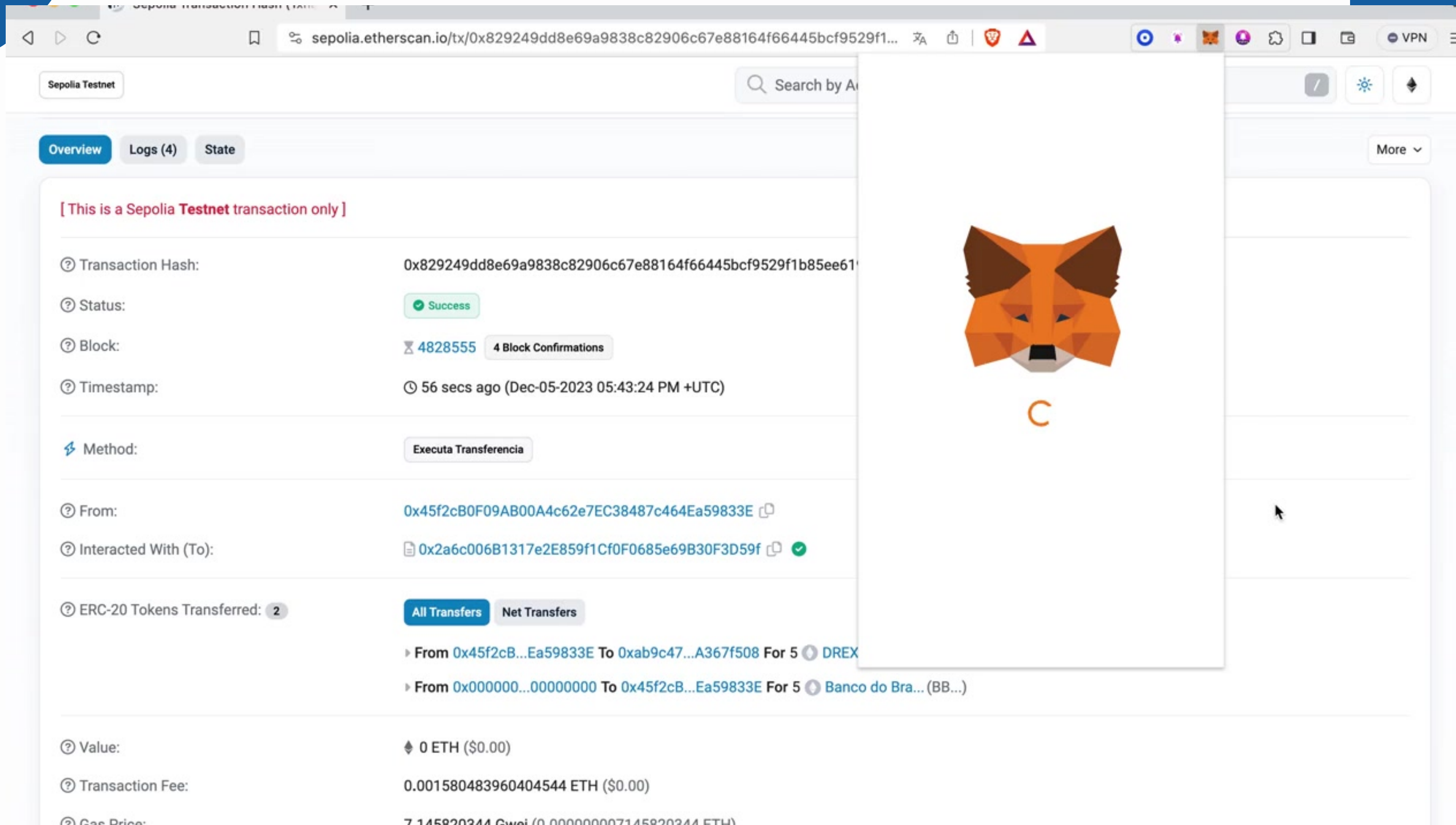
All Transfers Net Transfers

- From 0x45f2cB...Ea59833E To 0xab9c47...A367f508 For 5 DREX
- From 0x000000...00000000 To 0x45f2cB...Ea59833E For 5 Banco do Bra... (BB...)

Value: 0 ETH (\$0.00)

Transaction Fee: 0.001580483960404544 ETH (\$0.00)

Gas Price: 7145820244 Gwei (0.000000007145820244 ETH)



The image shows a screenshot of the Etherscan Sepolia Testnet interface. The transaction hash is 0x829249dd8e69a9838c82906c67e88164f66445bcf9529f1b85ee61... and the status is 'Success'. The transaction was included in block 4828555, 4 block confirmations ago, at 05:43:24 PM UTC on Dec-05-2023. The method is 'Executa Transferencia'. The transaction was sent from address 0x45f2cB0F09AB00A4c62e7EC38487c464Ea59833E to address 0x2a6c006B1317e2E859f1Cf0F0685e69B30F3D59f. The transaction transferred 2 ERC-20 tokens. The value is 0 ETH (\$0.00) and the transaction fee is 0.001580483960404544 ETH (\$0.00). The gas price is 7145820244 Gwei (0.000000007145820244 ETH). A modal window is open in the center of the screen, displaying a stylized orange fox head logo and the letter 'C' below it.



# Visualizando atividade do contrato na blockchain e o pagamento recebido

The screenshot displays the Etherscan Sepolia Testnet interface for a specific contract address. The page is divided into several sections:

- Source Code:** A tab at the top left of the main content area.
- Overview:** Shows the ETH balance as 0 ETH.
- More Info:** Displays the contract creator as 0xab9c47...A367f508 at transaction 0xef0523499fa8dfb35...
- Multi Chain:** Shows 0 addresses found via Blockscan.
- Transactions, Token Transfers (ERC-20), Contract, Events:** A row of tabs below the overview section, with 'Contract' currently selected.
- Code, Read Contract, Write Contract:** A row of tabs within the 'Contract' section, with 'Code' selected.
- Similar Match Source Code:** A warning section indicating that the contract matches the deployed bytecode of another contract (0xdEB851...B0Bb4a79) but that the constructor portion might be different.
- Contract Details:**
  - Contract Name:** TransfereChain
  - Optimization Enabled:** Yes with 500 runs
  - Compiler Version:** v0.8.21+commit.d9974bed
  - Other Settings:** shanghai EvmVersion
- Contract Source Code:** A section showing the Solidity source code in Standard Json-Input format. The code is for 'TransfereChain.sol' and includes imports for 'RealTokenizado.sol' and 'SwapTwoSteps.sol'.



# Visualizando atividade do contrato na blockchain e o pagamento recebido

Sepolia Testnet

Search by Address

Overview Logs (4) State

[ This is a Sepolia Testnet transaction only ]

Transaction Hash: 0x829249dd8e69a9838c82906c67e88164f66445bcf9529f1b85ee61...

Status: Success

Block: 4828555 4 Block Confirmations

Timestamp: 56 secs ago (Dec-05-2023 05:43:24 PM +UTC)

Method: Executa Transferencia

From: 0x45f2cB0F09AB00A4c62e7EC38487c464Ea59833E

Interacted With (To): 0x2a6c006B1317e2E859f1Cf0F0685e69B30F3D59f

ERC-20 Tokens Transferred: 2

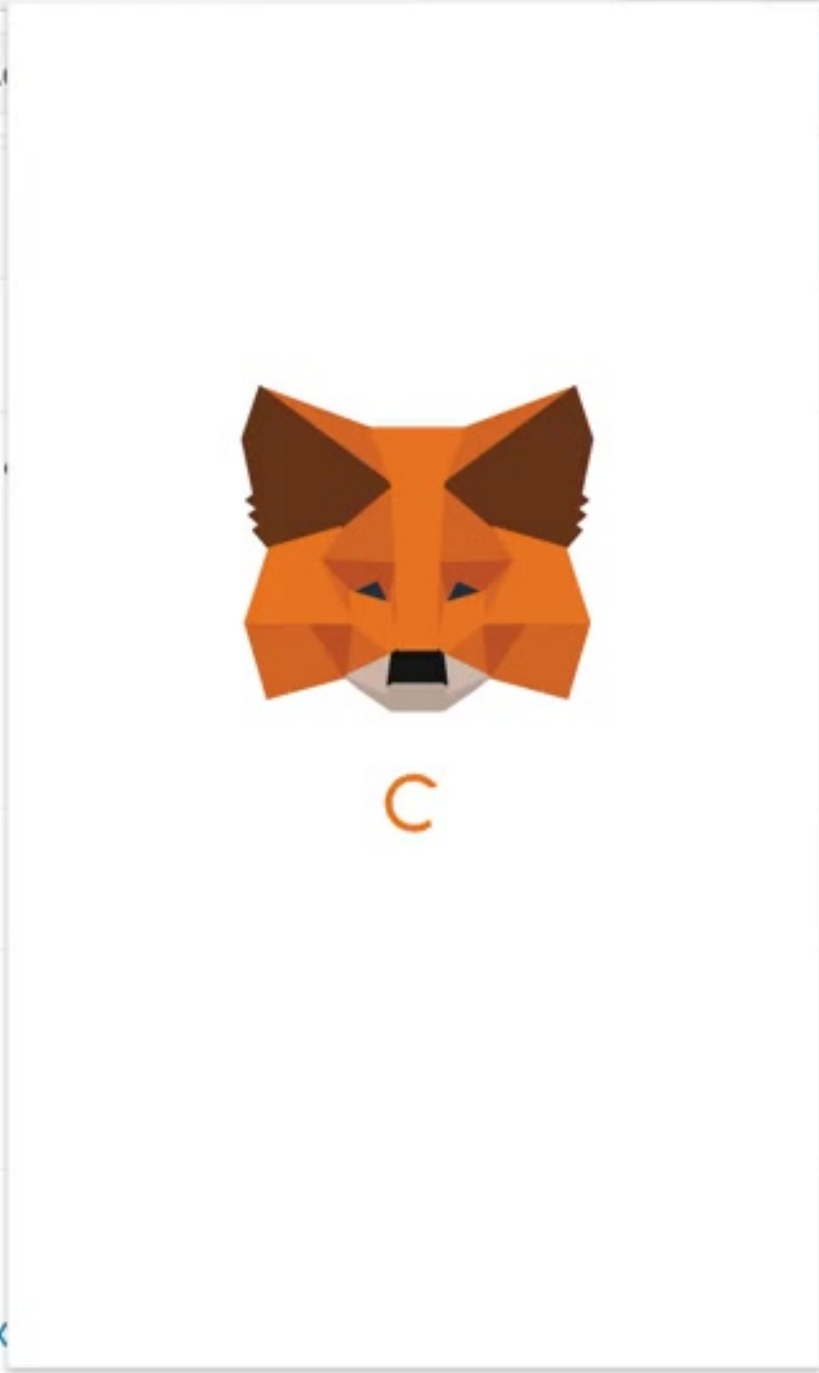
All Transfers Net Transfers

- From 0x45f2cB...Ea59833E To 0xab9c47...A367f508 For 5 DREX
- From 0x000000...00000000 To 0x45f2cB...Ea59833E For 5 Banco do Bra... (BB...)

Value: 0 ETH (\$0.00)

Transaction Fee: 0.001580483960404544 ETH (\$0.00)

Gas Price: 7 145820244 Gwei (0.000000007145820244 ETH)



# Modelo de negócio

8

## Parcerias Chave

Tesouro Nacional

Secretaria de Orçamento  
Federal

Banco central

7

## Atividades Chave

Desenvolver sistema de transferência com DREX que permita acessar as carteiras de parlamentar, Tesouro Nacional e beneficiário com uso de blockchain

Viabilizar o acesso à informação da finalidade do uso daquele recursos destinados

Marketing para adesão à nova forma de transferência

6

## Recursos Chave

Tecnologia de software para gerenciar as transferências

Base de dados de parlamentares e beneficiários  
(municípios, Estados e DF)

Segurança de dados e transparência

2

## Proposta de Valor

Permitir ao governo prestar conta do recurso repassado

Confere transparência para a população sobre o repasse de recursos

Rastreamento do recurso repassado para o parlamentar analisar se sua finalidade foi adequada

O uso do DREX possibilitaria uma possível redução na incidência de delitos como lavagem de dinheiro e fraudes, a criação de novas oportunidades de modelos de negócio, produtos e serviços mais eficientes e justos, além da inclusão financeira

# Diferenciais Competitivo

**01**

Transferências especiais com DREX

**02**

Aplicação de condicionantes para a execução do contrato

**03**

Prestação de conta obrigatória para a repasse de 50% restante do valor

**04**

Rastreabilidade da execução dos recursos



# Integração

Nós deixamos um script pronto para criar uma instância do contrato na rede Sepolia, conectá-lo à carteira do Tesouro Nacional e autorizá-lo, assim como aos outros participantes, a transacionar com o DREX. Para rodar o script deixamos um comando pronto para o CLI make deploytest. Para que ele funcione é preciso rodar o comando e criar um arquivo .env dentro da pasta “contratos-inteligentes” com o seguinte conteúdo:

```
PRIVATE_KEY=0x1d5b0bdba98776f403000b8f3b7cd64615b826708db3676d3485ec11f61831b2
SEPOLIA_RPC_URL=https://sepolia.infura.io/v3/f546297e6d854add87978a89e9bc3faf
ETHERSCAN_API_KEY=QA2K6VPW515HG68K9M3F2ET621DHACPCB4
AUTHORITY_KEY=0xa74e02ebea2e649567e90f071c2f57f52baea0c33cbc57e98bb8a61b3e88ec1f
ADMIN_KEY=0xb5f1bd4369c1fa0d4efd0cee3819c23b9d3b8e8e778e9137cf3b8ac5e82e5e8d
CITY_KEY=0x784f153636bfe90bb661ff6ee398d1a59239e5fd1d2fddf1c147250c04be1e8e
```

O script também criará o registro de um parlamentar e um beneficiário por parte do administrador do contrato e depois criará através do parlamentar uma requisição de transferência ao beneficiário cadastrado. Então o beneficiário aceitará a transferência e receberá a primeira parcela do pagamento, como mostra o vídeo. Após o prazo de carência definido em 30 dias, ele poderá sacar a segunda parcela.

Nota\*. Um bug no código impede que um segundo pedido de transferência seja aceito. Para consertar o bug, a linha 204 do contrato **TransfereChain.sol** deve ser substituída por **todosPedidosDeTransferencia[pedido.id - 1 == 0 ? pedido.id - 1 : (pedido.id - 1) / 2].estado = pedido.estado;**

# OBRIGAD@



## Equipe Tesouro Em Bytes



Karen Rapp Py-Daniel  
Analista de sistemas



Bruno Figueiredo  
Engenheiro de Software



Michael Matos  
Desenvolvedor full stack



Marcelo Frayha  
Desenvolvedor Solidity

