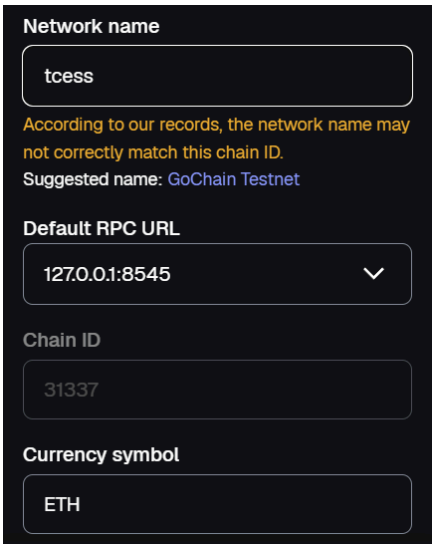
Local test for SCAD

First, choose an Environment on Remix. For the Test I used the Custom External Provider and connected to a wallet from MetaMask for my tests.

Remix:
https://remix.ethereum.org/#lang=en&optimize&runs=200&evmVersion&version=soljson-v0.8.30+commit.73712a01.js

The network of test that I used



Make sure the contractInfo.js and the App.js are compatible with the address of your contract.
App.js way - projeto\client\src
contractInfo.js way - projeto\client\src\constants
Then, in the /projeto/backend start a node with npx hardhat node. Add one node to your wallet on testnet and use it when Metamask asks you to associate your account.

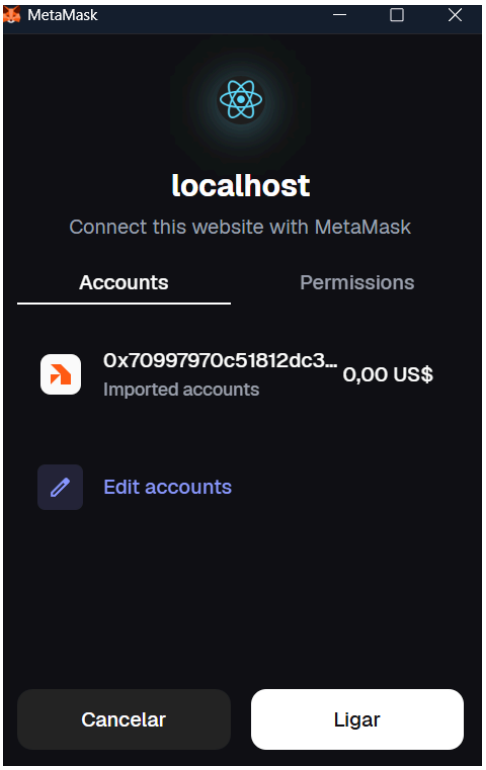
Account, proof of transaction:

```
Contract deployment: <UnrecognizedContract>
Contract address:    0x5fbdb2315678afecb367f032d93f642f64180aa3
Transaction:         0x4ae6ca65b12f4941cf454ccf738446cb6d741d51cd1d5f6cc9b75f5244656c2f
From:                0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
Value:               0 ETH
Gas used:            1256693 of 1256693
Block #1:            0xbda703bdd25affc5eeeb80e4e98d7915ce068c1f124194c177bb2ef723ce861b
```

After that, on the frontend past, start the program with Npm start.

```
Compiled successfully!

You can now view client in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://172.24.88.151:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Now, we will analyse the components of SCAD.
When the program asks you to connect the wallet, we will connect the one we used until that moment.



After, it shows us the whole App with the functions that we wanted. After guaranteeing that everything is well implemented, we can go ahead and connect with the TCESS environment. Thanks for the patience.