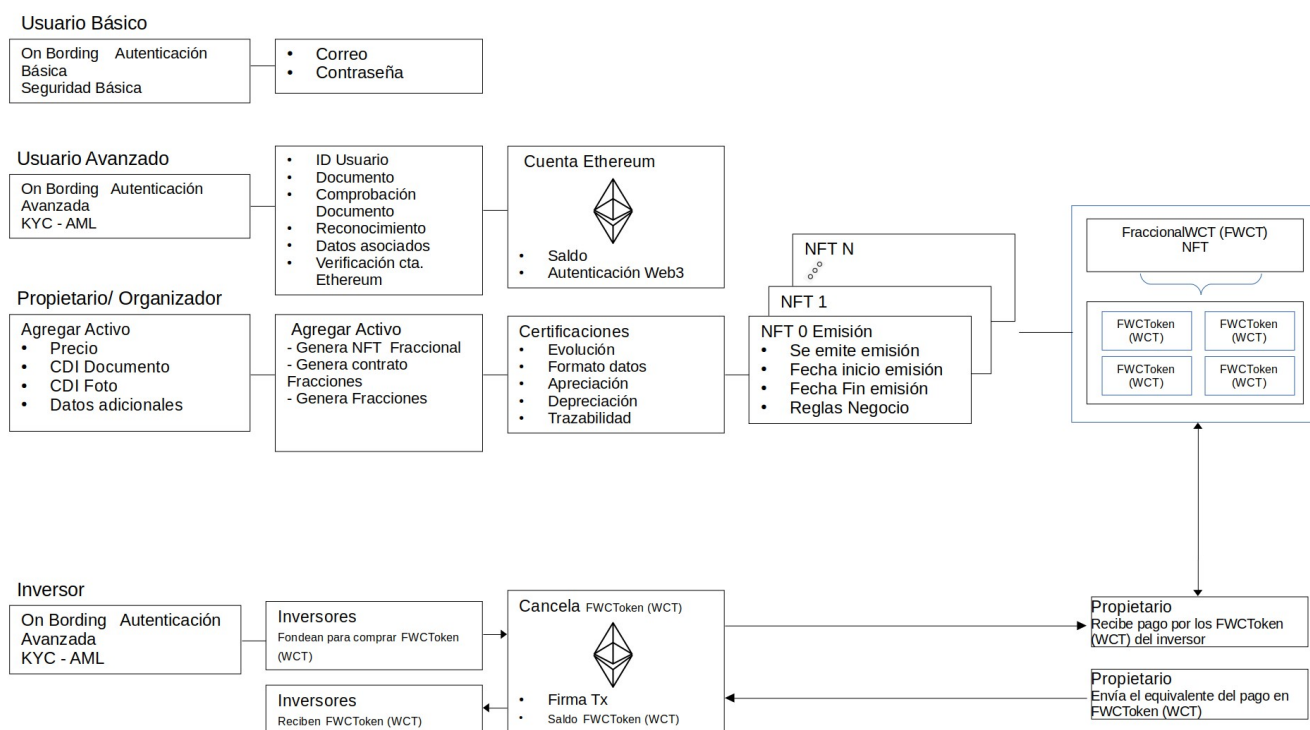


Procedimiento de manejo para Tokenización de Activos Digitales basados en blockchain y usando NFT Fraccionales.

Token NFT fraccional

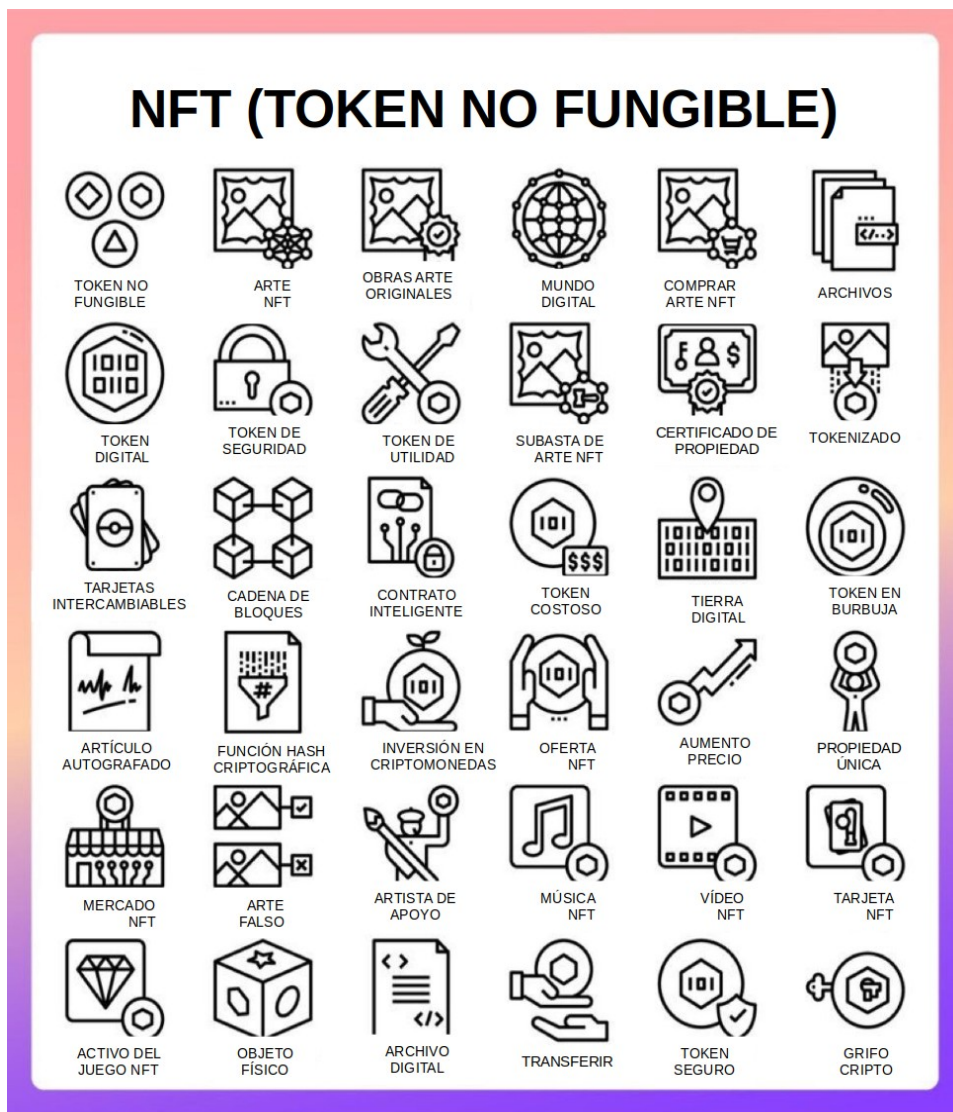
En lo analizado para desarrollar la Dapp Tokenización de Activos, procedí a incluir en el contrato inteligente de activos que había desarrollado lo referente a fraccionar el token NFT en varias parte aquí un acercamiento a su uso.

La fraccionalización la vamos a usar ya que queremos hacer que cualquier persona pueda obtener una parte de un token NFT y que su valor sea asequible y lo puedan comprar. Esta singularidad hace que un activo sea propiedad colectiva de un grupo de personas.



Entonces un NFT que se corta en varias partes y se distribuye a un grupo de personas registradas como propietarios parciales del NFT se conoce como NFT Fraccional. Esto es lo que vamos a trabajar en este documento.

Que es un NFT:



Hagamos un ejercicio para la comprensión, funcionamiento y la información requerida por cada actor.

Tengo un activo y es un buen candidato para ser tokenizado como NFT, como propietario puedo crear un NFT para representarlo. Para ello como propietario debo tener una wallet, un saldo y acceso al nodo (en este caso Ethereum).

Debo implementar el contrato inteligente que me va a permitir esta funcionalidad en nuestro caso lo creamos y vemos el registro en Etherscan.

The screenshot shows the Etherscan interface for the Ropsten Testnet Network. The main heading is "Contract 0x9fc1be2640bc3235f5441506d51852e4392f06c2". Below this, there's a "Contract Overview" section showing a balance of 0 Ether. To the right, a "More Info" section shows "My Name Tag: Not Available" and "Contract Creator: 0x6e927980222a26265a... at txn 0x2b925ecd39abaa66e3...". Below these sections is a "Transactions" tab, which is active, showing a table of transactions. The table has columns: Txn Hash, Method, Block, Age, From, To, Value, and Txn Fee. One transaction is listed: Txn Hash 0x2b925ecd39abaa66e3..., Method 0x60806040, Block 12579969, Age 3 mins ago, From 0x6e927980222a26265a..., To IN, Value 0 Ether, and Txn Fee 0.00952246. A "Download CSV Export" link is at the bottom right of the table.

Nuestra dirección del contrato para tokenizar activos es la siguiente:
0x9fc1be2640bc3235f5441506d51852e4392f06c2

The screenshot shows the Etherscan interface for the Ropsten Testnet Network, displaying the "Transaction Details" for the transaction 0x2b925ecd39abaa66e30adf5a3da564701f6e40e5758b7c9bdee24af6c825ce34. The transaction is labeled as a "Ropsten Testnet transaction only". The details are as follows: Transaction Hash: 0x2b925ecd39abaa66e30adf5a3da564701f6e40e5758b7c9bdee24af6c825ce34; Status: Success; Block: 12579969 (3 Block Confirmations); Timestamp: 53 secs ago (Jul-12-2022 08:53:24 PM +UTC); From: 0x6e927980222a26265a96b91d2cbe79e1f4ede0ac; To: [Contract 0x9fc1be2640bc3235f5441506d51852e4392f06c2 Created]; Value: 0 Ether (\$0.00); Transaction Fee: 0.009522462526662895 Ether (\$0.00); Gas Price: 0.000000002500000007 Ether (2.500000007 Gwei). A "Click to see More" link is at the bottom.

Luego para nuestro ejemplo buscaríamos inversores/amigos, para que co-compren el activo y van a ser registrados como co-propietarios del activo utilizando el concepto de NFT Fraccional. El inversor al igual que el propietario debe tener una wallet, un saldo y acceso al nodo (en este caso Ethereum) y es donde va a recibir el equivalente en tokens WeCoin (WCT) de su inversión registrada en la aplicación, el saldo se registra en la wallet del propietario o en la que se designe.

El propietario antes de acuñar el NFT debe tener a mano lo siguiente:

- Documento colateral del activo.
- Imagen o plano de la propiedad (metadata)

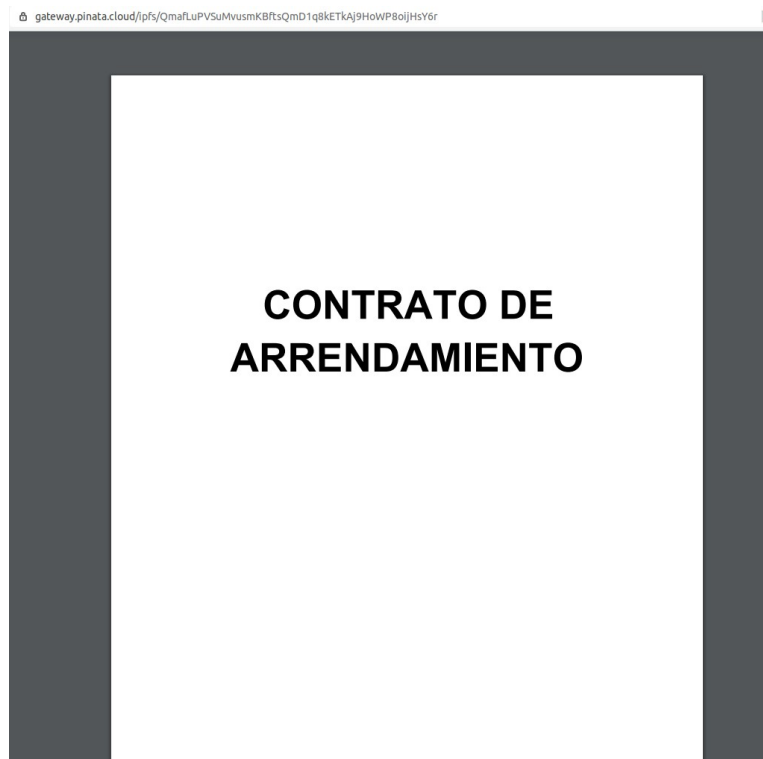
Esta información es para ser guardada en IPFS y se obtiene la huella digital que es un nombre de archivo único para cada documento.

Esta huella digital (nombre de archivo en IPFS) es lo que grabamos en la blockchain de Ethereum y con esto garantizamos la inmutabilidad, confianza y seguridad de la información asociada al activo.

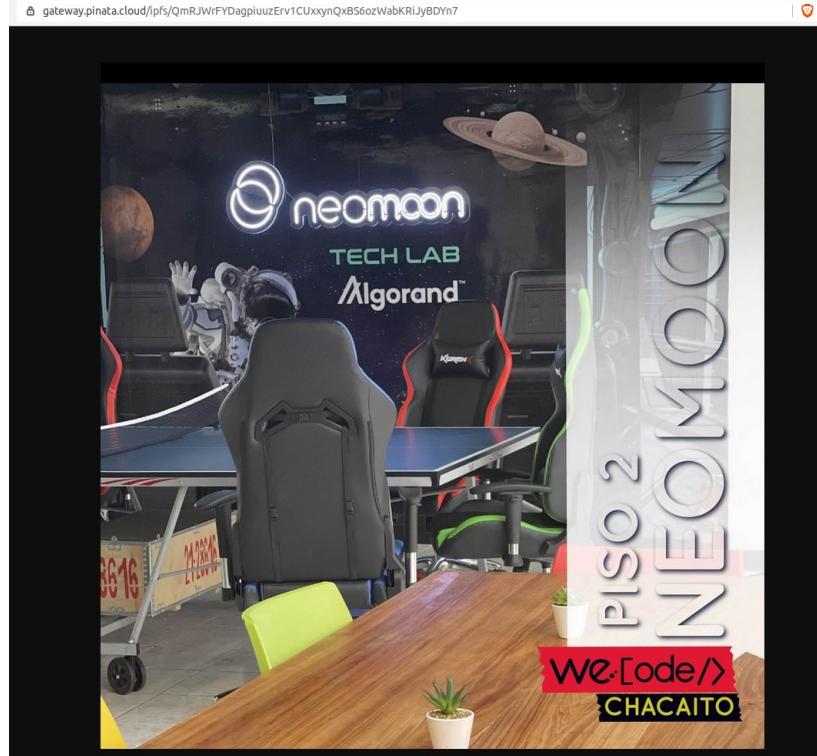
Esto lo resolvemos con <https://app.pinata.cloud/pinmanager> aquí cargamos la imagen y nos devuelve el CDI

Y luego podemos mostrarla así estos documentos son ejemplos solamente:

<https://gateway.pinata.cloud/ipfs/QmafLuPVSuMvusmKBftsQmD1q8kETkAj9HoWP8oijHsY6r>



<https://gateway.pinata.cloud/ipfs/QmRJWrFYDagpiuuzErv1CUxxynQxBS6ozWabKRiJyBDYn7>



El propietario decide que el activo será co-propietarios él y 2 inversores en 4 acciones o parte iguales, donde 1 inversor será propietario de 2 acciones del activo y otro inversor y él mismo propietario serán dueños de 1 acción cada uno. Y cada acción estará representada con un token ERC20 denominado “WeCoinToken” (WCT).

Con toda esta información a la mano el propietario procede a acuñar o minar el NFT para distribuirlo en 4 porciones.

El contrato inteligente con la información suministrada por el propietario acuña el NFT y lo distribuye en 4 porciones.

El contrato inteligente acuña 1 token NFT que representa el activo y 4 tokens ERC20 conocidos como WeCoinToken (WCT) para representar 4 acciones de propiedad.

Para hacer la acuñación o minado el token, el propietario debe ingresar lo siguiente:

- Precio de cada token a fraccionar.
- Numero de token a fraccionar.
- Tiempo del proyectos.
- Área del inmueble.
- Año de construcción
- Huella digital del documento contrato garantía.
- Huella digital de la fotografía o plano del activo.
- Dirección de cuenta Ethereum del dueño de la propiedad, en este caso igual al del emisor del token.

Esta información es grabada en la blockchain y es inmutable, Aquí vemos la función del contrato inteligente agregando un activo fraccional a la blockchain

http://localhost:3021/activos/agregarActivosFrac

POST http://localhost:3021/activos/agregarActivosFrac

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

| KEY | VALUE |
|-----------|--|
| price | 2600 |
| frac | 6 |
| tiempo | 12 |
| area | 950 |
| aconst | 1985 |
| tokenCDI | QmafLuPVSuMvumKBftsQmD1q8kETkAJ9HoWP8ojjHsY6r |
| tokenCDI1 | QmRJWFrFYDagpluuzErv1CUxxynQxBS6ozWabKRJyBDyn7 |
| address | 0x6e927980222a26265a96B91d2CbE79E1F4Ede0AC |
| Key | Value |

Body Cookies Headers (9) Test Results

Y vemos la respuesta a la solicitud http://localhost:3021/activos/agregarActivosFrac

Overview POST http://localhost:3021/activos/agregarActivosFrac GET http://localhost:3021/activos/agregarActivosFrac GET https://dev.finn GET https://dev.finn GET http://localhost:3021/activos/agregarActivosFrac GET http://localhost:3021/activos/agregarActivosFrac POST http://localhost:3021/activos/agregarActivosFrac GET http://localhost:3021/activos/agregarActivosFrac POST http://localhost:3021/activos/agregarActivosFrac + No Environment

http://localhost:3021/activos/agregarActivosFrac Save

POST http://localhost:3021/activos/agregarActivosFrac

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Body Cookies Headers (9) Test Results

Status: 200 OK Time: 30.50 s Size: 3.03 KB

Pretty Raw Preview Visualize JSON

```
1 {
2   "blockHash": "0x944bb49ced591ac1c4826190277b852613862ae231d6feef0283d8e2550b8973",
3   "blockNumber": 12608510,
4   "contractAddress": null,
5   "cumulativeGasUsed": 1531293,
6   "effectiveGasPrice": 1499934388,
7   "from": "0x6e927980222a26265a96b91d2cbe79e1f4ede0ac",
8   "gasUsed": 1335243,
9   "logs": [
10    {
11      "address": "0x91c18E2640bc3235f5441506d51852E4392f06c2",
12      "blockHash": "0x944bb49ced591ac1c4826190277b852613862ae231d6feef0283d8e2550b8973",
13      "blockNumber": 12608510,
14      "data": "0x",
15      "logIndex": 5,
16      "removed": false,
17      "topics": [
18        "0x4d252ad1be2c89b69c2b066f378d9a952ba7f163cda11628f55a4df523b3ef",
19        "0x0000000000000000000000000000000000000000000000000000000000000000",
20        "0x0000000000000000000000000000000000000000000000000000000000000000",
21        "0x0000000000000000000000000000000000000000000000000000000000000000",
22      ],
23      "transactionHash": "0xa4744758853b02415bb7f7aba8a9f7bc3f30fe8f56f343b337c6b5745385efe9",
24    }
25  ]
26 }
```

Al indicarle al contrato inteligente que acuñe se crea lo siguiente:

- El contrato NFT fraccional (ERC721)
- Token NFT fraccional FNFT FraccionalWCT (FWCT)
- Los token fraccionarios (ERC20) FWCToken (WCT)

El token FWCT es un token NFT que representa la propiedad con un vínculo a su documento colateral y su foto o plano en IPFS y los datos solicitados.

Cada uno de los 4 tokens WCT representa la propiedad parcial del token FWCT .

}

Podemos verificar lo que se genera al momento de Agregar Activo Fraccional en Etherscan

<https://ropsten.etherscan.io/tx/0xa4744758853b02415bb7f7aba8a9f7bc3f30fe8f56f343b337c6b5745385efe9>

Aquí vemos la Tx con la que se creo el FraccionalWCT (FWCT).

ropsten.etherscan.io/tx/0xa4744758853b02415bb7f7aba8a9f7bc3f30fe8f56f343b337c6b5745385efe9

This testnet will be deprecated soon. Migrate your contracts and deploy new ones on Goerli or Sepolia. Read more

Etherscan

Ropsten Testnet Network

Home

Transaction Details

Overview Internal Txns Logs (3) State

[This is a Ropsten Testnet transaction only]

| | |
|-----------------------|---|
| Transaction Hash: | 0xa4744758853b02415bb7f7aba8a9f7bc3f30fe8f56f343b337c6b5745385efe9 |
| Status: | Success |
| Block: | 12608510 22641 Block Confirmations |
| Timestamp: | 3 days 12 hrs ago (Jul-17-2022 02:50:48 AM +UTC) |
| From: | 0x6e927980222a26265a96b91d2cbe79e1f4ede0ac |
| Interacted With (To): | Contract 0x9fc1be2640bc3235f5441506d51852e4392f06c2 |
| Tokens Transferred: | From 0x00 To 0x6e927980222a26265a96b91d2cbe79e1f4ede0ac For 6 FWCToken (WCT) |
| Tokens Transferred: | From 0x00 To 0x6e927980222a26265a96b91d2cbe79e1f4ede0ac For ERC-721 Token ID [5] FractionalWC... (FWCT) |
| Value: | 0 Ether (\$0.00) |
| Transaction Fee: | 0.002002776892036284 Ether (\$0.00) |
| Gas Price: | 0.000000001499934388 Ether (1.499934388 Gwei) |

Aquí podemos ver el FraccionalWCT (FWCT) y su Token ID 5 y el WCTToken (WCT) con las 6 fracciones al darle click en FWCToken (WCT) muestra la siguiente información.

ropsten.etherscan.io/token/0x1200d0c6837474c04d60e8a82a2d4c19a53db0b7

This testnet will be deprecated soon. Migrate your contracts and deploy new ones on Goerli or Sepolia. Read more here.

Etherscan

Search by Address / Txn Hash / Block / Token / Ens

Ropsten Testnet Network

Home Blockchain Tokens Misc Ropsten

Token FWCToken

Overview (ERC-20)

Max Total Supply: 6 WCT

Holders: 1

Transfers: 1

Profile Summary

Contract: 0x1200d0c6837474c04d60e8a82a2d4c19a53db0b7

Decimals: 18

Transfers Holders Contract

A total of 7 transactions found

| Txn Hash | Method | Age | From | To | Quantity |
|-------------------------|-----------|-------------------|-------------------------|-------------------------|----------|
| 0xa4744758853b02415b... | 0x2a3c5b7 | 3 days 12 hrs ago | 0x000000000000000000... | 0x6e927980222a26265a... | 6 |

Aquí podemos ver la Tx que creo el FWCToken (WCT), la dirección propietaria de los Tokens y la cantidad de Tokens fraccionados, y un dato importante es que genera una dirección de contrato que asocia el FraccionalWCT (FWCT) 1, hash **0x9fc1be2640bc3235f5441506d51852e4392f06c2** con sus partes el FWCToken (WCT) 6, el hash del contrato inteligente de las fracciones **0x1200D0C6837474c04d60E8a82A2d4c19A53DB0b7** esta relación de estos dos contratos se mapea y con ella tenemos en el estado del la EMV siempre esta información disponible. Es decir que cada activo tiene un smart contract asociado a sus partes diferente con el que será identificado.

A continuación el servicio que lista los Activos y asocia el FWCToken (WCT) con sus fracciones y muestra el hash del contrato que los asocia :

Workspaces ▾ API Network ▾ Explore 🔍 Search

Overview POST http://localhost GET http://localhost GET https://dev.finh POST https://dev.finh GET http://localhost GET http://localhost POST http://localhost GET http://localhost POST http://localhost POST http://localhost + ... No Environment

http://localhost:3021/activos/activosFracFWCT Save

GET http://localhost:3021/activos/activosFracFWCT Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

| KEY | VALUE | DESCRIPTION |
|-----|-------|-------------|
| Key | Value | Description |

Body Cookies Headers (9) Test Results Status: 200 OK Time: 4.19 s Size: 2.51 KB Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   {
3     "id": "0",
4     "precio": "1050",
5     "frac": "4",
6     "tiempo": "12",
7     "tokenURI": "QmaTLuPVsuMvusmKBftsQnD1q8KETKAj9HomPB0iJHsY6z",
8     "propietario": "0x6e92798922a2a26265a96891d2cbe79E1F4Ede0AC",
9     "aprobado": "None",
10    "diToken": {
11      "0": "0",
12      "1": "0x015C4b8e3B36F8c2B1f03d433f96a9032928A52",
13      "tokenId": "0",
14      "fractionalToken": "0x015C4b8e3B36F8c2B1f03d433f96a9032928A52"
15    },
16    "totalToken": "4000000000000000000"
17  },
18  {
19    "id": "1",

```

Detalle del servicio activosFracFWCT

```
{
  "id": "0",
  "precio": "1050",
  "frac": "4",
  "tiempo": "12",
  "tokenURI": "QmafLuPVSuMvusmKBftsQmD1q8kETkAj9HoWP8oijHsY6r",
  "propietario": "0x6e927980222a26265a96B91d2CbE79E1F4Ede0AC",
  "aprobado": "None",
  "dirToken": {
    "0": "0",
    "1": "0x015C4b8e3B36FBc2B1f03d433fb96a9032928A52",
    "tokenId": "0",
    "fractionalToken": "0x015C4b8e3B36FBc2B1f03d433fb96a9032928A52"
  },
  "totalToken": "4000000000000000000"
},
{
  "id": "1",
  "precio": "2000",
  "frac": "4",
  "tiempo": "6",
  "tokenURI": "QmafLuPVSuMvusmKBftsQmD1q8kETkAj9HoWP8oijHsY6r",
  "propietario": "0x6e927980222a26265a96B91d2CbE79E1F4Ede0AC",
  "aprobado": "None",
  "dirToken": {
    "0": "1",
    "1": "0x581A3A88C980Fdd90d6716C6254E8293efB8A049",
    "tokenId": "1",
    "fractionalToken": "0x581A3A88C980Fdd90d6716C6254E8293efB8A049"
  },
  "totalToken": "4000000000000000000"
},
{
  "id": "2",
  "precio": "3000",
  "frac": "6",
  "tiempo": "12",
  "tokenURI": "QmafLuPVSuMvusmKBftsQmD1q8kETkAj9HoWP8oijHsY6r",
```

```

    "propietario": "0x6e927980222a26265a96B91d2CbE79E1F4Ede0AC",
    "aprobado": "None",
    "dirToken": {
      "0": "2",
      "1": "0x609e912E4755B202F43EF98dEC5D3287b21327E6",
      "tokenId": "2",
      "fractionalToken": "0x609e912E4755B202F43EF98dEC5D3287b21327E6"
    },
    "totalToken": "6000000000000000000"
  },
  {
    "id": "3",
    "precio": "2500",
    "frac": "5",
    "tiempo": "12",
    "tokenURI": "QmafLuPVSuMvusmKBftsQmD1q8kETkAj9HoWP8oijHsY6r",
    "propietario": "0x6e927980222a26265a96B91d2CbE79E1F4Ede0AC",
    "aprobado": "None",
    "dirToken": {
      "0": "3",
      "1": "0x456bb86b93320CE7010C6fb3AF17082a1f629A26",
      "tokenId": "3",
      "fractionalToken": "0x456bb86b93320CE7010C6fb3AF17082a1f629A26"
    },
    "totalToken": "5000000000000000000"
  },
  {
    "id": "4",
    "precio": "3000",
    "frac": "6",
    "tiempo": "12",
    "tokenURI": "QmafLuPVSuMvusmKBftsQmD1q8kETkAj9HoWP8oijHsY6r",
    "propietario": "0x6e927980222a26265a96B91d2CbE79E1F4Ede0AC",
    "aprobado": "None",
    "dirToken": {
      "0": "4",
      "1": "0x834ce29afaB66c7D762e853B5FcBCa8fe8F34E02",
      "tokenId": "4",
      "fractionalToken": "0x834ce29afaB66c7D762e853B5FcBCa8fe8F34E02"
    },
    "totalToken": "6000000000000000000"
  },
  {
    "id": "5",
    "precio": "2600",
    "frac": "6",
    "tiempo": "12",
    "tokenURI": "QmafLuPVSuMvusmKBftsQmD1q8kETkAj9HoWP8oijHsY6r",
    "propietario": "0x6e927980222a26265a96B91d2CbE79E1F4Ede0AC",
    "aprobado": "None",
    "dirToken": {
      "0": "5",
      "1": "0x1200D0C6837474c04d60E8a82A2d4c19A53DB0b7",
      "tokenId": "5",
      "fractionalToken": "0x1200D0C6837474c04d60E8a82A2d4c19A53DB0b7"
    },
    "totalToken": "6000000000000000000"
  }
}

```

Servicio **balanceWCT** por dirección de cartera del propietario del token y el Id del activos, :

Overview

POST http://localhost:3021/activos/balanceWCT

GET http://localhost:3021/activos/balanceWCT

GET https://dev.finhac.com/activos/balanceWCT

POST https://dev.finhac.com/activos/balanceWCT

GET http://localhost:3021/activos/balanceWCT

GET http://localhost:3021/activos/balanceWCT

POST http://localhost:3021/activos/balanceWCT

http://localhost:3021/activos/balanceWCT

GET

http://localhost:3021/activos/balanceWCT

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

| KEY | VALUE |
|---|--|
| <input checked="" type="checkbox"/> assetId | 0 |
| <input checked="" type="checkbox"/> address | 0x6e927980222a26265a96b91d2cbe79e1f4ede0ac |
| Key | Value |

Body

Cookies

Headers (9)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "id": "0",
3   "contractWCT": "0x015C4b8e3B36FBc2B1f03d433fb96a9032928A52",
4   "walletAddress": "0x6e927980222a26265a96b91d2cbe79e1f4ede0ac",
5   "balanceWCT": "4"
6 }
7
8
```

Distribución parcial del activo:

Supongamos como enunciamos en el ejemplo que el dueño de la propiedad ha vendido la mitad de la propiedad a un amigo/inversor y una cuarta parte de la propiedad a otro amigo/inversor.

Para representar esto, simplemente se transfiere 1 tokens FNT al primer inversor y 2 token FNT al segundo inversor. Solo haremos el envío de 1 Token a la dirección:

0x1D3bF2Bf979dE2BC9D2D00CC6246ad8Bc55ae292

Procedemos a hacer el envío:

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:3021/activos/enviarWCT`
- Method:** `POST`
- Body Type:** `x-www-form-urlencoded`
- Body Data:**

| KEY | VALUE | DESCRIPTION |
|---|---|-------------|
| <input checked="" type="checkbox"/> <code>activoid</code> | <code>5</code> | |
| <input checked="" type="checkbox"/> <code>receiver</code> | <code>0x1D3bF2Bf979dE2BC9D2D00CC6246ad8Bc55ae292</code> | |
- Response:** Status: 200 OK, Time: 6.54 s, Size: 2.09 KB. The response body is a JSON object with fields: `Id`, `from`, `Contract`, `to`, `amount`, and `transactionReceipt`.

El servicio me envía el hash de la transacción que es:

transactionHash: **'0xc23bc675c12fa16cdabc4f5fe3bb8d3d11a442f65ce9454a755d7027636593020'**

```
[
  {
    "Id": "5",
    "from": "0x6e927980222a26265a96B91d2CbE79E1F4Ede0AC",
    "Contract": "0x1200D0C6837474c04d60E8a82A2d4c19A53DB0b7",
    "to": "0x1D3bF2Bf979dE2BC9D2D00CC6246ad8Bc55ae292",
    "amount": "1",
    "transactionReceipt": {
      "blockHash": "0x40c29e27fc052ec3543cbcf41de1a8082ed5e2330551e16ef80f46c9aa3300cf",
      "blockNumber": 12608632,
      "contractAddress": null,
      "cumulativeGasUsed": 1001282,
      "effectiveGasPrice": 1499934388,
      "from": "0x6e927980222a26265a96b91d2cbe79e1f4ede0ac",
      "gasUsed": 29668,
      "logs": [
```


ropsten.etherscan.io/tx/0xf1b6444a34fbd9381123e2858af3a23f345ef58bda573bbca28f496ebde362f

Etherscan Ropsten Testnet Network

All Filters 0xf1b6444a34fbd9381123e2858af3a23f345ef58bda573bbca28f496ebde362f

Home Blockchain Tokens Misc Ropster

Transaction Details

Overview Logs (1) State

[This is a Ropsten Testnet transaction only]

Transaction Hash: 0xf1b6444a34fbd9381123e2858af3a23f345ef58bda573bbca28f496ebde362f

Status: Success

Block: 12580337 3 Block Confirmations

Timestamp: 36 secs ago (Jul-12-2022 10:16:48 PM +UTC)

From: 0x6e927980222a26265a96b91d2cbe79e1f4ede0ac

Interacted With (To): Contract 0x015c4b8e3b36fbc2b1f03d433fb96a9032928a52

Tokens Transferred: From 0x6e927980222a2... To 0x1d3bf2bf979de2... For 1 FWCToken (WCT)

Value: 0 Ether (\$0.00)

Transaction Fee: 0.000076901352817344 Ether (\$0.00)

Gas Price: 0.000000001491261108 Ether (1.491261108 Gwei)

[Click to see More](#)

La transacción del envío del FWCToken (WCT) muestra el contrato y las direcciones de y para

ropsten.etherscan.io/token/0x015c4b8e3b36fbc2b1f03d433fb96a9032928a52

Etherscan Ropsten Testnet Network

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

Home Blockchain Tokens Misc Ropster

Token FWCToken

Overview [ERC-20]

Max Total Supply: 4 WCT

Holders: 1

Transfers: 2

Profile Summary

Contract: 0x015c4b8e3b36fbc2b1f03d433fb96a9032928a52

Decimals: 18

Transfers Holders Contract

A total of 2 transactions found

| Txn Hash | Method | Age | From | To | Quantity |
|-------------------------|------------|-------------|-------------------------|--------------------------|----------|
| 0xf1b6444a34fbd93811... | Transfer | 3 mins ago | 0x6e927980222a26265a... | 0x1d3bf2bf979de2bc9d2... | 1 |
| 0xf44a501824a893d21c... | 0x2a3c5cb7 | 45 mins ago | 0x000000000000000000... | 0x6e927980222a26265a... | 4 |

[Download CSV Export]

Y luego vemos el detalle de la transacción

Presentamos el balance FWCToken (WCT), indicando el Id del activo y la cuenta a consultar: _____

http://localhost:3021/activos/balanceWCT

GET

http://localhost:3021/activos/balanceWCT

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

| | KEY | VALUE | DESCRIPTION |
|-------------------------------------|---------|--|-------------|
| <input checked="" type="checkbox"/> | assetId | 0 | |
| <input checked="" type="checkbox"/> | address | 0x6e927980222a26265a96b91d2cbe79e1f4ede0ac | |
| | Key | Value | Description |

Body

Cookies

Headers (9)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "id": "0",
3   "contractWCT": "0x015C4b8e3B36F8c2B1f03d433fb96a9032928A52",
4   "walletAddress": "0x6e927980222a26265a96b91d2cbe79e1f4ede0ac",
5   "balanceWCT": "3"
6 }
7
8
```

Aquí vemos que el propietario tiene 3 tokens

http://localhost:3021/activos/balanceWCT

GET

http://localhost:3021/activos/balanceWCT

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

| | KEY | VALUE | DESCRIPTION | | Bulk 1 |
|-------------------------------------|---------|--|-------------|--|--------|
| <input checked="" type="checkbox"/> | assetId | 0 | | | |
| <input checked="" type="checkbox"/> | address | 0x1D3bf28f979de28C902D00CC6246ad88c55ae292 | | | |
| | Key | Value | Description | | |

Body

Cookies

Headers (9)

Test Results

Pretty

Raw

Preview

Visualize

JSON

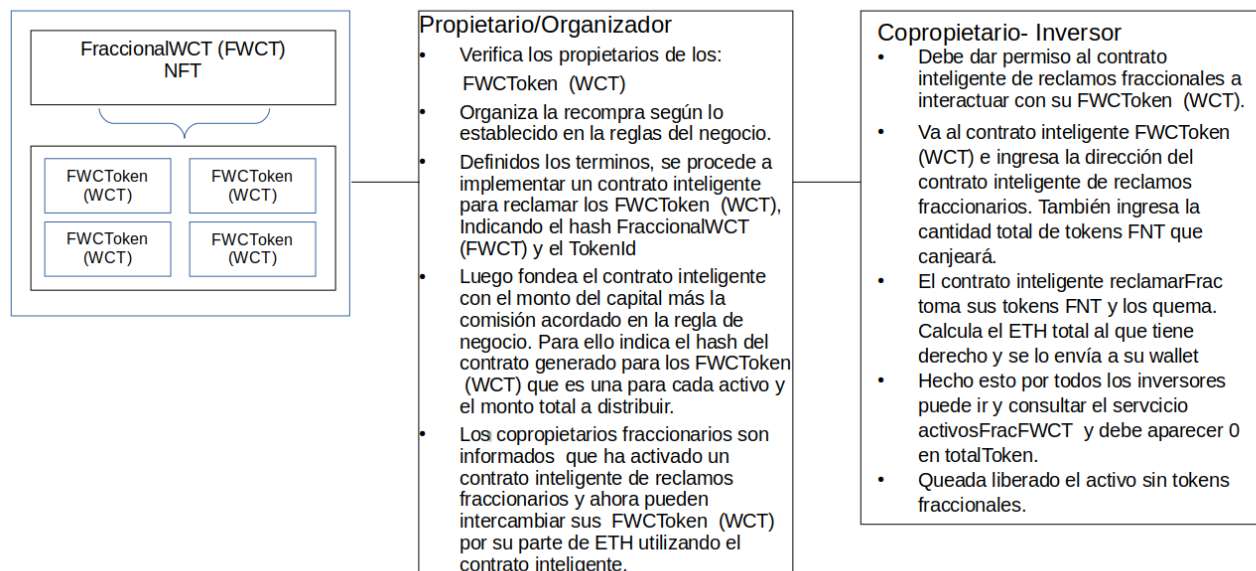
```
1 {
2   "id": "0",
3   "contractWCT": "0x015C4b8e3B36F8c2B1f03d433fb96a9032928A52",
4   "walletAddress": "0x1D3bf28f979de28C902D00CC6246ad88c55ae292",
5   "balanceWCT": "1"
6 }
7
8
```

Status: 200 OK Time: 259 ms Size: 495 B Save Response

Y la cuenta a la que enviamos tiene 1 token

Con esto completamos lo que había denominado 1ra fase y podemos cubrir hasta los inversores y su envío de tokens.

La segunda fase es la siguiente:



Aquí se hace lo siguiente:

El propietario u organizador anuncia la recompra y para ello implementa un contrato inteligente para que los co-propietarios reclamen su inversión y su comisión según la reglas de negocio.

En este contrato para reclamar inversión el propietario coloca el monto total de inversión + comisiones.

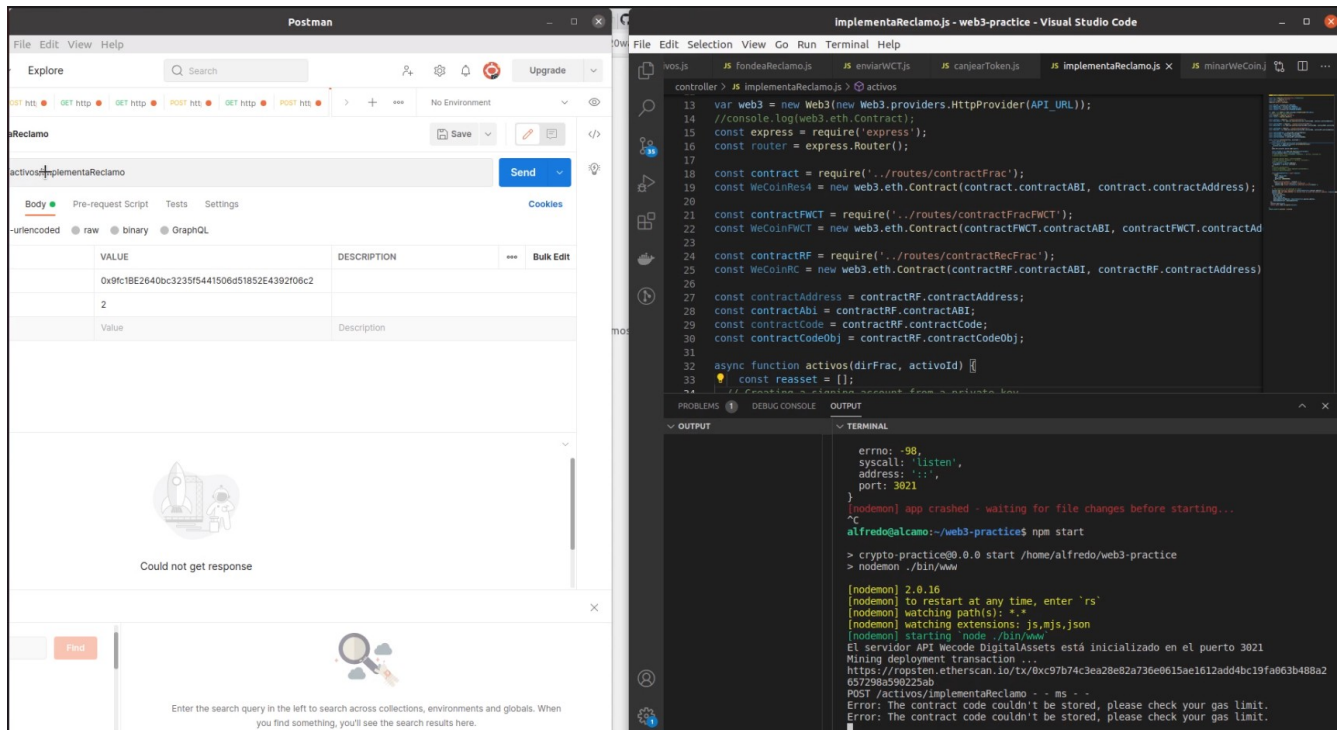
Luego cada uno de los co-propietarios reclaman su inversión y comisión y una vez hecho se libera el token y se quema.

Al hacerlo todos el token NFT queda libre sin co-propietarios y se puede vender o quemar.

Esta segunda parte esta programado y probado lo del contrato de reclamar inversion.

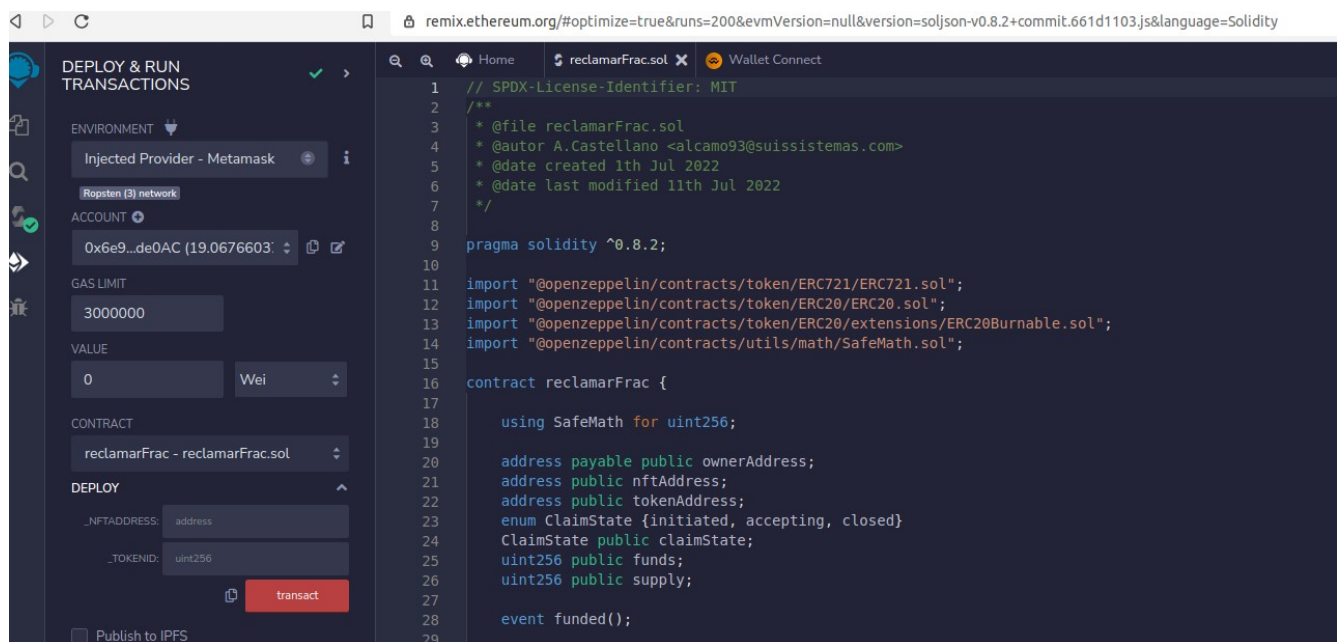
Estoy probando los servicios asociados al contrato y luego lo documento.

Se programo el servicio `implementaReclamo` para poder recomprar los FWCToken (WCT) es decir las partes del token NFT FraccionalWCT (FWCT) , para ello en este servicio debemos indicar el hash del FraccionalWCT (FWCT) y el TokenId del activo a recomprar como se ve en la siguiente pantalla:



El servicio `implementaReclamo` se esta programando como los anteriores, para efecto de nuestro ejemplo lo probaremos con Remix y Red Ropsten

Lo primero que vamos hacer es implementar el contrato inteligente `reclamoFrac` vía Remix, para luego fondear y que los inversores puedan canjear sus token por su dinero e intereses y liberen para quemar los tokens.



Lo que debemos hacer en esta pantalla de Remix es los siguiente:

Indicar el hash del FraccionalWCT (FWCT): **0x9fc1be2640bc3235f5441506d51852e4392f06c2**
y el TokenId del activo a recomprar: **2**

En Remix vamos al IDE en el navegador y conectamos por esta vía al nodo Ethereum y formamos las Tx a través de Metamask, esto me permite certificar y generar transacciones de los contratos en uso, esto lo había hecho inicialmente y funciona, como indique el error se genera al conectarnos via Infura para y no usar ni Remix, ni Metamask.

Acá el funcionamiento de las 3 (tres) funciones que se requieren para devolverle la inversión y el fee a los inversores:

implementaReclamo: Permite que el organizador/ propietario despliegue el contrato inteligente que su función es fondear el contrato del activo que se desea recomprar y permitir que el inversor reclame el capital y el fee de su inversión canjeando los token.

fondeaReclamo: Esta función permite al organizador/ propietario colocar el el monto y fee total a pagar a los inversores, acá indica el contrato del token que tiene las fracciones y que es uno por cada activo.

canjearToken: Permite al inversor retirar la inversión y el fee y devuelve la propiedad del activo al organizador/propietario, el inversor va al contrato de su fracción y le indica el contrato del reclamo y el monto de os token a canjear, al hacerlo se devuelve el monto +fee y se queman os token.

Como indicamos en Remix ingresamos

_NFTADDRESS: **0x9fc1be2640bc3235f5441506d51852e4392f06c2** y

_TOKENID: **2**

Y presionamos transact, en la pantalla a continuación vemos que nuestro contrato se implemento en la red de prueba roptsten.

remix.ethereum.org/#optimize=true&runs=200&evmVersion=null&version=soljson-v0.8.2+commit.661d1103.js&language=Solidity

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT
Injected Provider - Metamask
Ropsten (3) network
ACCOUNT
0x6e9...de0AC (19.0665318!)
GAS LIMIT
3000000
VALUE
0 Wei
CONTRACT
reclamarFrac - reclamarFrac.sol
DEPLOY
_NFTADDRESS: 0x9fc1be2640bc3235f5441506d5
_TOKENID: 2
transact
Publish to IPFS
OR
At Address Load contract from Address
Transactions recorded 1
Deployed Contracts
RECLAMARFRAC AT 0XB30...A9AB7

```
1 // SPDX-License-Identifier: MIT
2 /**
3  * @file reclamarFrac.sol
4  * @author A.Castellano <alcamo93@suissistemas.com>
5  * @date created 1th Jul 2022
6  * @date last modified 11th Jul 2022
7  */
8
9 pragma solidity ^0.8.2;
10
11 import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
```

0 Listen on all transactions Search with transaction hash or address

- Select a Javascript file in the file explorer and then run 'remix.execute()' or 'remix.executeCurrent()' in the console
- Right click on a JavaScript file in the file explorer and then click 'Run'

The following libraries are accessible:

- web3 version 1.5.2
- ethers.js
- remix

Type the library name to see available commands.
creation of reclamarFrac pending...

[view on etherscan](#)

✓ [block:12636845 txIndex:2] from: 0x6e9...de0AC to: reclamarFrac.(constructor) value: 0 wei data: 0x608...00002

| | |
|------------------|---|
| status | true Transaction mined and execution succeed |
| transaction hash | 0x9cb7046ae44fe2ff5a3012de17683b76d1a6da7fb0a97ee4c69639bf5a8741d |
| from | 0x6e927980222a26265a96891d2CbE79E1F4Ede0AC |
| to | reclamarFrac.(constructor) |
| gas | 451409 gas |
| transaction cost | 451409 gas |
| input | 0x608...00002 |
| decoded input | { "address _nftAddress": "0x9fc1BE2640bc3235f5441506d51852E4392f06c2", "uint256 _tokenId": "2" } |
| decoded output | - |
| logs | [] |
| val | 0 wei |

Para verificar lo ocurrido vamos a Etherscan y revisamos la Tx, como se observa en la siguiente pantalla:

All Filters
Search by Address / Txn Hash / Block / Token / E

Ropsten Testnet NetworkHomeBlockchain

Transaction Details

Overview
Internal Txns
State

[This is a Ropsten Testnet transaction only]

Transaction Hash:

0x9cb7046ae44fe2ff5a3012de17603b76d1a6da7fb0a97eea4c69639b5a8741d

Status:

Success

Block:

126368459 Block Confirmations

Timestamp:

1 min ago (Jul-21-2022 12:32:36 PM +UTC)

From:

0x6e927980222a26265a96b91d2cbe79e1f4ede0ac

To:

[Contract 0xb30426dbd82e270a12897984cc7dc424383a9ab7 Created]

Value:

0 Ether (\$0.00)

Transaction Fee:

0.001128522503159863 Ether (\$0.00)

Gas Price:

0.000000002500000007 Ether (2.500000007 Gwei)

Click to see More

0xb30426dbd82e270a12897984cc7dc424383a9ab7

El contrato creado:

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

Ropsten Testnet NetworkHomeBlockchainTokensMiscRopsten

Contract 0xB30426dBd82e270A12897984cC7dC424383A9ab7

Contract Overview

Balance: 0 Ether

More Info

My Name Tag: Not Available

Contract Creator: 0x6e927980222a26265a... at txn 0x9cb7046ae44fe2ff5a3...

Transactions
Internal Txns
Erc20 Token Txns
Contract
Events

Latest 1 from a total of 1 transactions

| Txn Hash | Method | Block | Age | From | To | Value | Txn Fee |
|--------------------------|------------|----------|------------|-------------------------|----------------------|---------|------------|
| 0x9cb7046ae44fe2ff5a3... | 0x60806040 | 12636845 | 4 mins ago | 0x6e927980222a26265a... | IN Contract Creation | 0 Ether | 0.00112852 |

[Download CSV Export]

ⓘ A contract address hosts a smart contract, which is a set of code stored on the blockchain that runs when predetermined conditions are met. Learn more about addresses in our Knowledge Base.

El propietario de la propiedad "comprará" los tokens FNT de los propietarios fraccionales. Para hacerlo, deposita en monto de lo invertido y el fee 1 ETH en el contrato inteligente de reclamos fraccionarios. Al hacerlo, proporciona la dirección del contrato inteligente del token FNT.

Para ello creamos el servicio **fondeaReclamo** y solicita los siguientes datos:

dirReclamo: Debemos proporcionar la dirección del contrato inteligente de reclamoFrac que es el que generamos en el punto anterior.

0xb30426dbd82e270a12897984cc7dc424383a9ab7

dirWCT: Debemos proporcionar la dirección del contrato inteligente FWCToken (WCT), dirección del token que es uno por cada activo y lo vemos acá: Para el ejemplo:

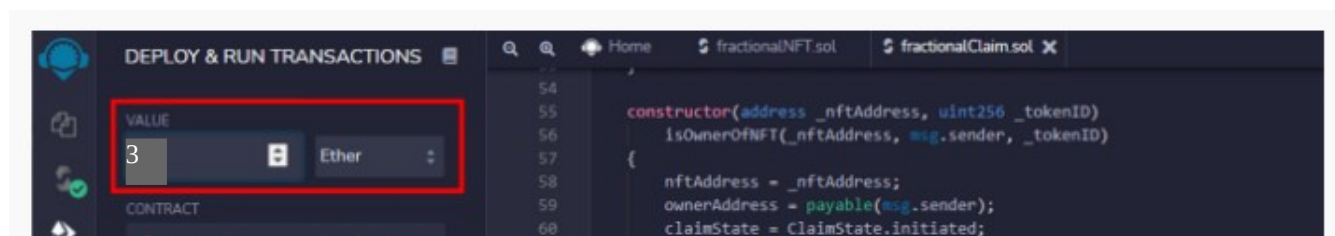
0x609e912E4755B202F43EF98dEC5D3287b21327E6

```
{
  "id": "2",
  "precio": "3000",
  "frac": "6",
  "tiempo": "12",
  "tokenURI": "QmafLuPVSuMvumKBftsQmD1q8kETkAj9HoWP8oijHsY6r",
  "propietario": "0x6e927980222a26265a96B91d2CbE79E1F4Ede0AC",
  "aprobado": "None",
  "dirToken": {
    "0": "2",
    "1": "0x609e912E4755B202F43EF98dEC5D3287b21327E6",
    "tokenId": "2",
    "fractionalToken": "0x609e912E4755B202F43EF98dEC5D3287b21327E6"
  },
  "totalToken": "6000000000000000000"
}
```

amount: Se indica el monto inversión + fee total repartir entre los tokens, se indica en Eth.

3

En Remix indicamos el valor, debemos considerar que el valor es en Ether.



Y luego en el contrato inteligente reclamoFrac colocamos el hash FWCToken (WCT), dirección del token que es uno por cada activo **0x609e912E4755B202F43EF98dEC5D3287b21327E6**

Al presionar transact, se fondea el contrato reclamarFrac que va a permitir al inversor canjear sus token, vemos la TX en Etherscan, en este caso coloque 3wei se debe colocar en Eth, para fin de la pruebas funciona.

This testnet will be deprecated soon. Migrate your contracts and deploy new ones on Goerli or Sepolia. Read more here.

Etherscan
Ropsten Testnet Network

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

Home Blockchain Tokens Misc Ropsten

Transaction Details

Overview Logs (1) State

[This is a Ropsten Testnet transaction only]

Transaction Hash: 0x2383feca49316b4900cc6c5175bc61af30ac381c4a2161ae91d02ab04a313ce1

Status: Success

Block: 12636901 1 Block Confirmation

Timestamp: 4 secs ago (Jul-21-2022 12:44:24 PM +UTC)

From: 0x6e927980222a26265a96b91d2cbe79e1f4ede0ac

To: Contract 0xb30426dbd82e270a12897984cc7dc424383a9ab7

Value: 3 wei (\$0.00)

Transaction Fee: 0.000241922500677383 Ether (\$0.00)

Gas Price: 0.000000002500000007 Ether (2.500000007 Gwei)

Click to see More

<https://ropsten.etherscan.io/address/0xb30426dbd82e270a12897984cc7dc424383a9ab7>

ropsten.etherscan.io/address/0xb30426dbd82e270a12897984cc7dc424383a9ab7

This testnet will be deprecated soon. Migrate your contracts and deploy new ones on Goerli or Sepolia. Read more here.

Etherscan
Ropsten Testnet Network

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

Home Blockchain Tokens Misc Ropsten

Contract 0xb30426dbd82e270a12897984cc7dc424383a9ab7

Contract Overview

Balance: 3 wei

More Info

My Name Tag: Not Available

Contract Creator: 0x6e927980222a26265a... at txn 0x9cb7046ae44fe2ff5a3...

Transactions Internal Txns Erc20 Token Txns Contract Events

Latest 2 from a total of 2 transactions

| Txn Hash | Method | Block | Age | From | To | Value | Txn Fee |
|--------------------------|------------|----------|-------------|-------------------------|-------------------------|---------|------------|
| 0x2383feca49316b4900... | Fund | 12636901 | 5 mins ago | 0x6e927980222a26265a... | 0xb30426dbd82e270a12... | 3 wei | 0.00024192 |
| 0x9cb7046ae44fe2ff5a3... | 0x60806040 | 12636845 | 17 mins ago | 0x6e927980222a26265a... | Contract Creation | 0 Ether | 0.00112852 |

[Download CSV Export]

A contract address hosts a smart contract, which is a set of code stored on the blockchain that runs when predetermined conditions are met. Learn more about addresses in our Knowledge Base.

Aqui vemos que nuestro contrato inteligente reclamarFrac a sido fondeado.

Canjear tokens FNT por ETH

Luego, el propietario informa a sus propietarios fraccionarios que ha activado un contrato inteligente de reclamos fraccionarios y ahora pueden intercambiar sus tokens FWCToken (WCT) por su parte de ETH utilizando el contrato inteligente. (Le debe indicar el Hash)

En el contrato inteligente de reclamos fraccionarios, ingresa la dirección del contrato inteligente del token FWCToken (WCT) y la cantidad que está canjeando.

El contrato inteligente Fractional Claims toma sus tokens FNT y los quema. Calcula el ETH total al que tiene derecho y se lo envía.

Se creo un servicio **canjearToken** se le debe indicar la siguiente información:

ProFrac: Dirección de la cuenta del propietario de la fracción.

ActivoId: TokenId que identifica el activo.

Spender: Hash contrato inteligente de reclamos fraccionarios

amount: Cantidad de tokens a canjear

En Remix nos vamos al contrato inteligente FWCToken (WCT) a la función approve, indicamos el hash del contrato inteligente reclamoFrac **0xb30426dbd82e270a12897984cc7dc424383a9ab7** y la cantidad de tokens a canjear, en este caso 6

The screenshot displays the Remix IDE interface with the following components:

- Left Panel (DEPLOY & RUN TRANSACTIONS):** Shows the 'approve' function being executed. The 'spender' field is set to '0xb30426dbd82e270a12897984cc7dc424383a9ab7' and the 'amount' is '6'. The 'Transact' button is highlighted.
- Center Panel (Solidity Code):** Displays the Solidity code for the 'reclamoFrac' contract, including a license identifier, author information, and the 'approve' function signature.
- Right Panel (Transaction Details):** Shows the transaction status as 'true Transaction mined and execution succeed'. It lists the transaction hash, from address, to address, gas used, and transaction cost.
- Bottom Panel (Logs):** Displays the transaction logs, including the 'approve' function call and the resulting output.
- MetaMask Notification:** A pop-up window on the right side of the screen asks for permission to access the WCT, showing the transaction hash and the amount.

Vemos en Etherscan la Tx approve

Transaction Details

Overview

Logs (1)

State

[This is a Ropsten Testnet transaction only]

Transaction Hash:

0x7d6201ec558b50d5ad8c92dd219dfc657b2bfd86b8a8182051c8f6a1c9b1b1d

Status:

Success

Block:

12637004

458 Block Confirmations

Timestamp:

1 hr 44 mins ago (Jul-21-2022 01:09:00 PM +UTC)

From:

0x6e927980222a26265a96b91d2cbe79e1f4ede0ac

To:

Contract 0x609e912e4755b202f43ef98dec5d3287b21327e6

Value:

0 Ether (\$0.00)

Transaction Fee:

0.00011555000032354 Ether (\$0.00)

Gas Price:

0.000000002500000007 Ether (2.500000007 Gwei)

Etherscan

Ropsten Testnet Network

All Filters

Search by Address / Txn Hash / Block / Token / Ens

Home

Blockchain

Tokens

Misc

Ropsten

Contract 0x609e912E4755B202F43EF98dEC5D3287b21327E6

Contract Overview

Balance: 0 Ether

More Info

My Name Tag: Not Available

Contract Creator: 0x6e927980222a26265a... at txn 0xf6cbbf83f7c80255081...

Token Tracker: FWCToken (WCT)

Transactions

Internal Txns

Erc20 Token Txns

Contract

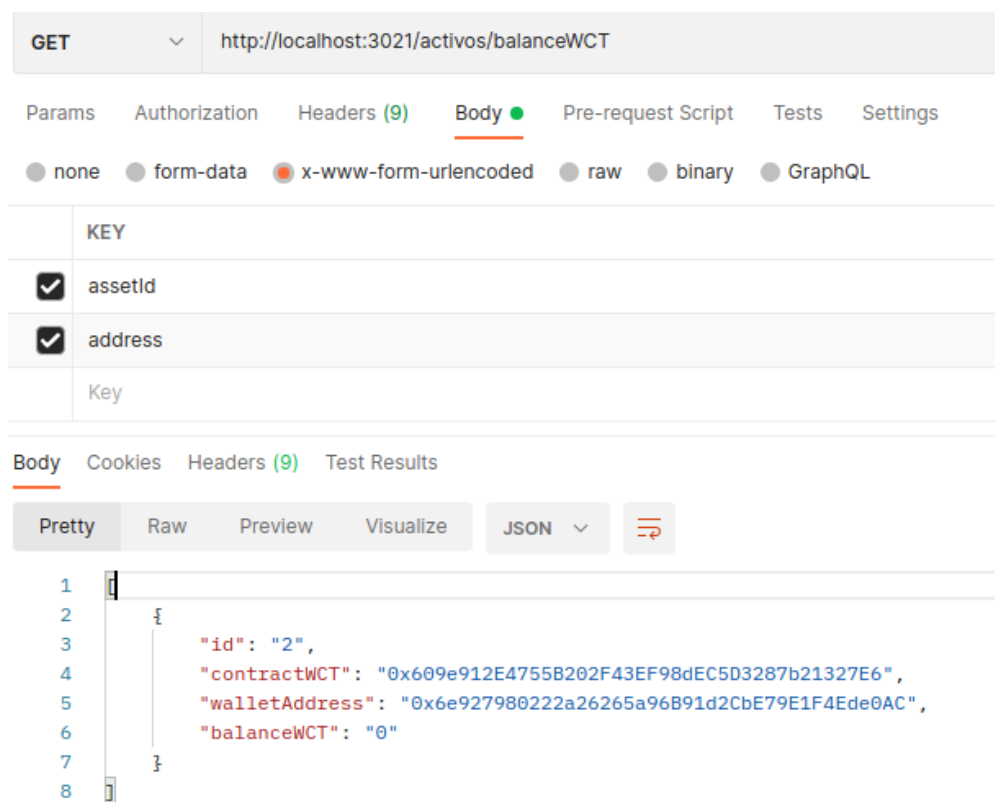
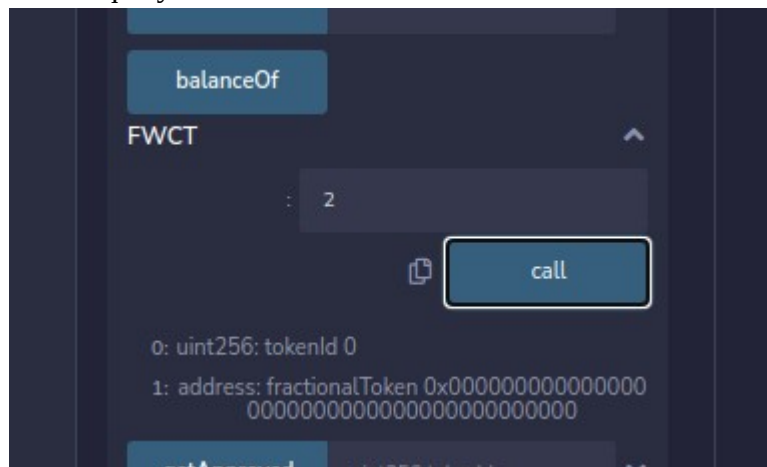
Events

Latest 7 from a total of 7 transactions

| Txn Hash | Method | Block | Age | From | To | Value | Txn Fee |
|-------------------------|---------|----------|------------------|-------------------------|----------------------------|---------|------------|
| 0x7d6201ec558b50d5ad... | Approve | 12637004 | 1 hr 44 mins ago | 0x6e927980222a26265a... | IN 0x609e912e4755b202f4... | 0 Ether | 0.00011555 |

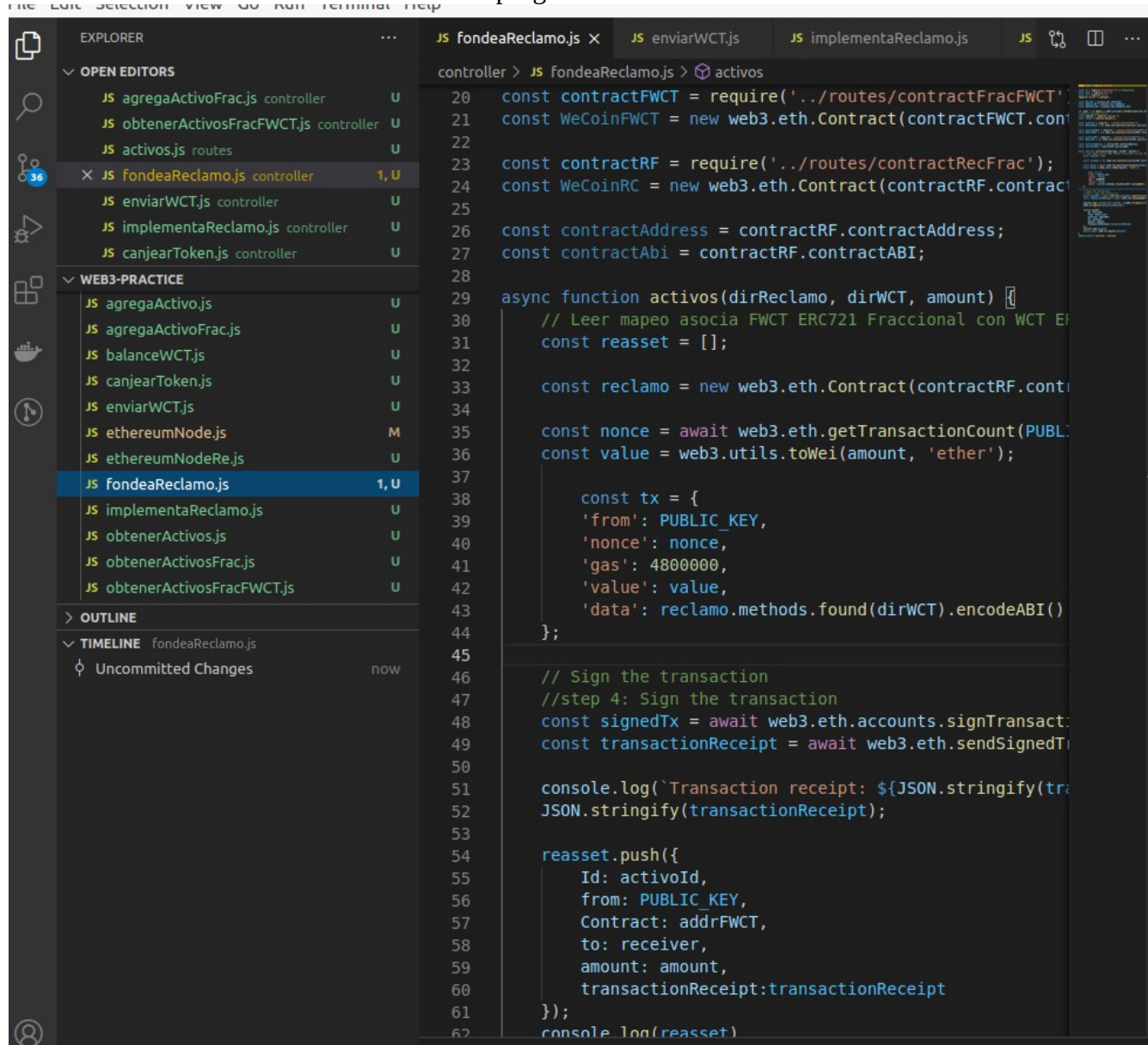
El contrato inteligente el contrato inteligente reclamoFrac toma sus tokens FNT y los quema. Calcula el ETH total al que tiene derecho y se lo envía al inversor.

Podemos verificar en Remix que ya to tenemos asociados token fraccionales asi:

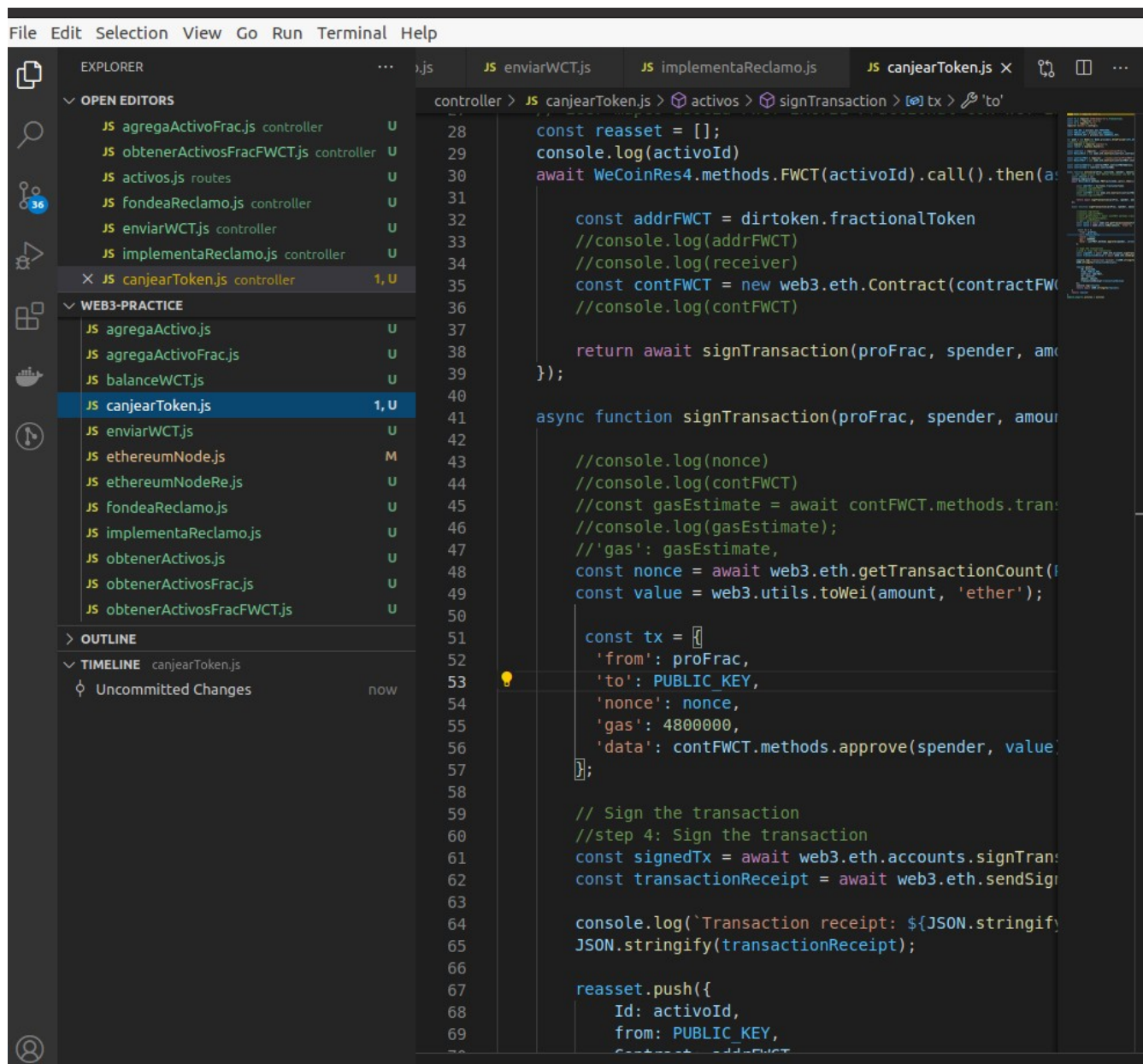


Como vemos el contrato inteligente reclamoFrac funciona totalmente vía Remix, en la espera de que Infura ayude a determinar el error y si no habría que ver la opción de otro acceso al nodo Ethereum.

Los servicios asociados a reclamarFrac esta programados:



```
20 const contractFWCT = require('../routes/contractFracFWCT');
21 const WeCoinFWCT = new web3.eth.Contract(contractFWCT.contractABI);
22
23 const contractRF = require('../routes/contractRecFrac');
24 const WeCoinRC = new web3.eth.Contract(contractRF.contractABI);
25
26 const contractAddress = contractRF.contractAddress;
27 const contractAbi = contractRF.contractABI;
28
29 async function activos(dirReclamo, dirWCT, amount) {
30   // Leer mapeo asocia FWCT ERC721 Fraccional con WCT ERC20
31   const reaset = [];
32
33   const reclamo = new web3.eth.Contract(contractRF.contractABI, contractAddress);
34
35   const nonce = await web3.eth.getTransactionCount(PUBLIC_KEY, 'latest');
36   const value = web3.utils.toWei(amount, 'ether');
37
38   const tx = {
39     'from': PUBLIC_KEY,
40     'nonce': nonce,
41     'gas': 4800000,
42     'value': value,
43     'data': reclamo.methods.found(dirWCT).encodeABI()
44   };
45
46   // Sign the transaction
47   //step 4: Sign the transaction
48   const signedTx = await web3.eth.accounts.signTransaction(tx, PRIVATE_KEY);
49   const transactionReceipt = await web3.eth.sendSignedTransaction(signedTx.rawTransaction);
50
51   console.log(`Transaction receipt: ${JSON.stringify(transactionReceipt)}`);
52
53   reaset.push({
54     Id: activoId,
55     from: PUBLIC_KEY,
56     Contract: addrFWCT,
57     to: receiver,
58     amount: amount,
59     transactionReceipt: transactionReceipt
60   });
61   console.log(reaset);
62 }
```



```
File Edit Selection View Go Run Terminal Help

EXPLORER
OPEN EDITORS
JS agregaActivoFrac.js controller U 28
JS obtenerActivosFracFWCT.js controller U 29
JS activos.js routes U 30
JS fondeaReclamo.js controller U 31
JS enviarWCT.js controller U 32
JS implementaReclamo.js controller U 33
X JS canjeaToken.js controller 1, U 34
WEB3-PRACTICE
JS agregaActivo.js U 37
JS agregaActivoFrac.js U 38
JS balanceWCT.js U 39
JS canjeaToken.js 1, U 40
JS enviarWCT.js U 42
JS ethereumNode.js M 43
JS ethereumNodeRe.js U 44
JS fondeaReclamo.js U 45
JS implementaReclamo.js U 46
JS obtenerActivos.js U 47
JS obtenerActivosFrac.js U 48
JS obtenerActivosFracFWCT.js U 49
OUTLINE
TIMELINE canjeaToken.js
Uncommitted Changes now

controller > JS canjeaToken.js > activos > signTransaction > tx > 'to'

const reasset = [];
console.log(activoId)
await WeCoinRes4.methods.FWCT(activoId).call().then(async () => {

  const addrFWCT = dirtoken.fractionalToken
  //console.log(addrFWCT)
  //console.log(receiver)
  const contFWCT = new web3.eth.Contract(contractFWCT, addrFWCT)
  //console.log(contFWCT)

  return await signTransaction(proFrac, spender, amount)
});

async function signTransaction(proFrac, spender, amount) {

  //console.log(nonce)
  //console.log(contFWCT)
  //const gasEstimate = await contFWCT.methods.transaction({
  //  'from': proFrac,
  //  'to': PUBLIC_KEY,
  //  'nonce': nonce,
  //  'gas': 4800000,
  //  'data': contFWCT.methods.approve(spender, value).encodeABI()
  //}).estimateGas({
  //  'from': proFrac,
  //  'to': PUBLIC_KEY,
  //  'nonce': nonce,
  //  'gas': 4800000,
  //  'data': contFWCT.methods.approve(spender, value).encodeABI()
  //});

  // Sign the transaction
  //step 4: Sign the transaction
  const signedTx = await web3.eth.accounts.signTransaction(transaction, privateKey);
  const transactionReceipt = await web3.eth.sendSignedTransaction(signedTx.rawTransaction);

  console.log(`Transaction receipt: ${JSON.stringify(transactionReceipt)}`);

  reasset.push({
    Id: activoId,
    from: PUBLIC_KEY,
    to: addrFWCT,
    amount: amount,
    nonce: nonce,
    gas: 4800000,
    data: contFWCT.methods.approve(spender, value).encodeABI()
  });
}
```

Como podemos ver a nivel de blockchain vemos que la prueba del concepto de token fraccional funciona y esto nos permite aplicarlo en diversos campos es infinito su uso.

En este ámbito de la fraccionalización el impacto en UK, artículo publicado la semana pasada 14-07-22. Esto en relación a la proyección y hacia donde se puede usar.

¿CAMBIO DE MODELO EN LOS FONDOS DE INVERSIÓN POR BLOCKCHAIN Y LA TOKENIZACIÓN?

https://www.linkedin.com/posts/alfredo-mu%C3%B1oz_investment-association-uk-investing-for-activity-6952524073998405633-tMhA?utm_source=linkedin_share&utm_medium=member_desktop_web

The Investment Association.pdf

La IA (The Investment Association) en UK, que agrupa a los Fondos de Inversión, defiende en este documento la necesidad de modernizar productos y servicios, por la inevitable innovación que supone la tecnología blockchain y los criptoactivos, que traerán beneficios en eficiencia operativa, menores costes con mejoras en prestación de servicios y un incremento y diversidad de los activos disponibles.

El fenómeno de la tokenización, claramente disruptivo, permitiría la propiedad fraccionada y supondría la apertura de mercados inicialmente accesibles con carácter institucional, facilitando el acceso y la liquidez, lo que podría generar inversiones hiperpersonalizadas.

Entienden que habría tres posibles escenarios que pueden producirse:

1. Mejoras sobre los fondos generando eficiencia sobre el funcionamiento actual, generando mejoras en la velocidad, la escala y la eficiencia, lo que exigiría la adopción de tokenización, permitiendo que las acciones de fondos y los activos subyacentes sean negociados de una manera mucho más efectiva.
2. Supone una evolución severa en la innovación con una personalización de la inversión, con bloques temáticos de inversión, ampliando la tokenización a empresas privadas, infraestructuras y activos digitales nativos
3. Y un tercer escenario, que supone un cambio radical, con una hiperpersonalización del inversor que disfrutaría de una experiencia interactiva y participativa, con exposiciones en acciones y valores individuales, en lugar de a través de un fondo. Habría un cambio en la participación del cliente y la relación pasaría de la delegación al fondo a una actuación como gestor de inversiones.

Por todo ello, la IA defiende tres ámbitos de colaboración entre los reguladores y la industria, que permitiera hacer más competitiva la industria de fondos en UK:

1. Innovación: Creando un marco para los fondos tokenizados; análisis de implicaciones de DeFi y consiguientes reformas.
2. Perímetro regulatorio: reglas para los criptoactivos, supervisión, control sobre las stablecoins y sus efectos
3. Cambio regulatorio: mejorar asesoramiento a los consumidores; suministro y divulgación de información digital a los inversores; generar reputación internacional de UK como centro financiero líder.