

PROYECTO 2: ICO

Declaraciones Iniciales:

En esta sección les dejo imágenes, que indican algunas configuraciones iniciales que he realizado, para que luego el resto de la indicaciones posteriores sean más comprensibles (Algunas de ellas están en el código proporcionado).

1. Configuración del Constructor del Smart Contract

```
//Constructor
constructor(address _escrow) ERC20 ("Avomo's Token","AVT"){
    require(openingTime>block.timestamp, "Incremente el tiempo de Inicio");
    require(openingTime<closingTime, "El tiempo de Inicio debe ser menor al tiempo final");

    admin = msg.sender;
    escrow = _escrow;

    _mint(address(this), tokenSupply);
}
```

2. Inicializaciones y datos que uso para pruebas en la red

```
//Admin = msg.sender : 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
//Escrow: 0xd0870fA1b7C4700F2BD7f44238821C26f7392148
//Investor1: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 - 200 tokens - Luego los venderá todos
//Investor2: 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db - 400 tokens
```

3. Balance inicial de Ethers y de Tokens del Smart Contract antes de las transacciones.

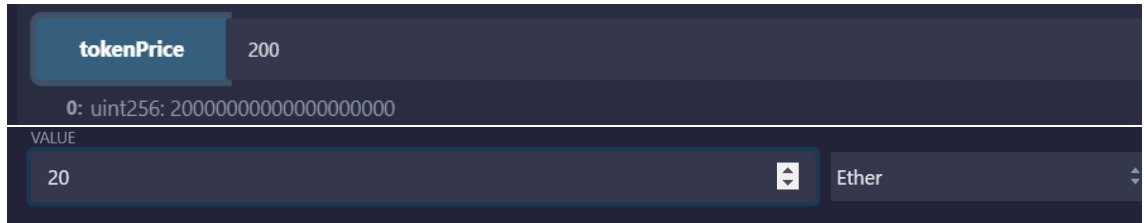
| | |
|-------------------------|--------------------------------|
| balancesEther... | balancesEthersSC - call |
| 0: uint256: 0 | |
| balanceToken... | |
| 0: uint256: 1000000 | |

Compra de tokens del Inversor1:

1. Balance de Tokens y de Ethers del inversor1 antes de las transacciones

| | |
|----------------------|--|
| balanceEthers | 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 |
| 0: uint256: 99 | |
| balanceOf | 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 |
| 0: uint256: 0 | |

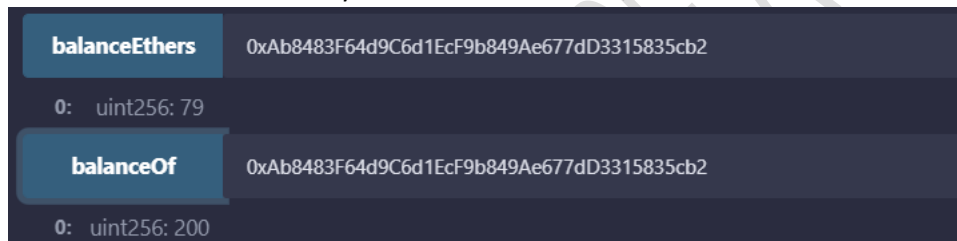
2. El inversor1 quiere comprar 200 Tokens de la ICO, por tanto hace uso de la función 'tokenPrice' para calcular el precio (20 ETH, véase en el código el exchangeRate y la lógica de la función).



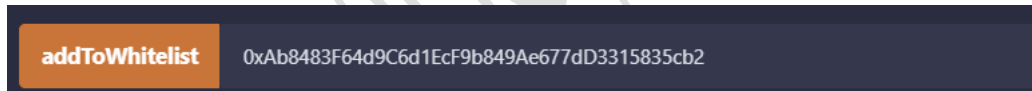
3. El Inversor, procede a comprar los Tokens de la ICO



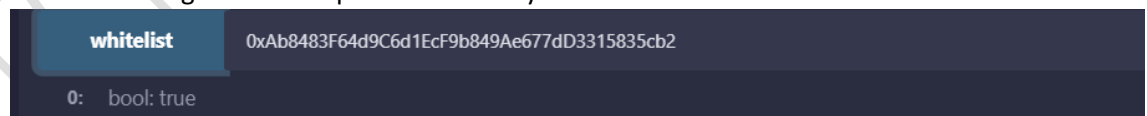
4. Balance de Ethers y de Tokens del Inversor, tras la transacción



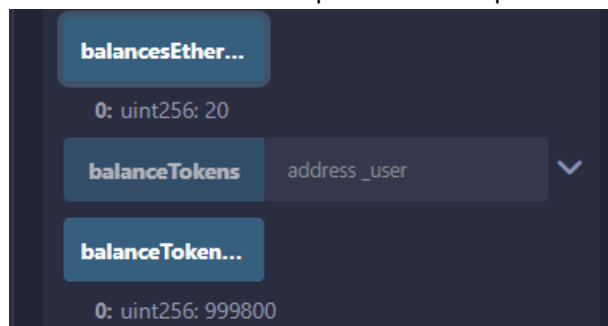
5. Añadimos el inversor a la Whitelist, utilizando la función 'addToWhitelis'



6. Nos aseguramos de que el inversor1 ya está en la Whitelist



7. Balances del Smart Contract después de la compra del Inversor1



Compra de Tokens del Inversor2:

1. Balance de Tokens y de Ethers del inversor1 antes de las transacciones

| | |
|----------------------|--|
| balanceEthers | 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db |
| 0: uint256: 99 | |
| balanceOf | 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db |
| 0: uint256: 0 | |

2. El inversor1 quiere comprar 400 Tokens de la ICO, por tanto hace uso de la función 'tokenPrice' para calcular el precio (40 ETH, véase en el código el exchangeRate y la lógica de la función).

| | |
|----------------------------------|-----|
| tokenPrice | 400 |
| 0: uint256: 40000000000000000000 | |

| | |
|---------------------------------|--------------------------------------|
| VALUE | |
| <input type="text" value="40"/> | <input type="button" value="Ether"/> |

3. El Inversor2, procede a comprar los Tokens de la ICO

| | |
|------------------|-----|
| buyTokens | 400 |
|------------------|-----|

4. Balance de Ethers y de Tokens del Inversor, tras la transacción

| | |
|----------------------|--|
| balanceEthers | 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db |
| 0: uint256: 59 | |
| balanceOf | 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db |
| 0: uint256: 400 | |

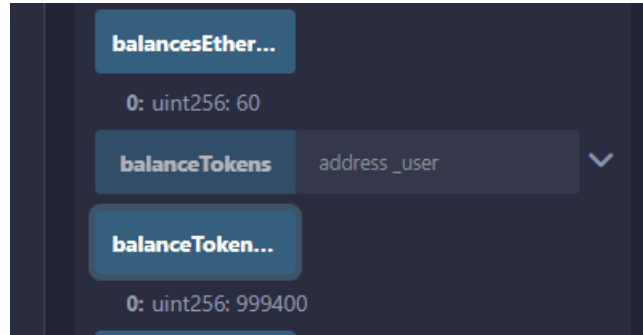
5. Añadimos el inversor a la Whitelist, utilizando la función 'addToWhitelis'

| | |
|-----------------------|--|
| addToWhitelist | 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db |
|-----------------------|--|

6. Nos aseguramos de que el inversor1 ya está en la Whitelist

| | |
|------------------|--|
| whitelist | 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db |
| 0: bool: true | |

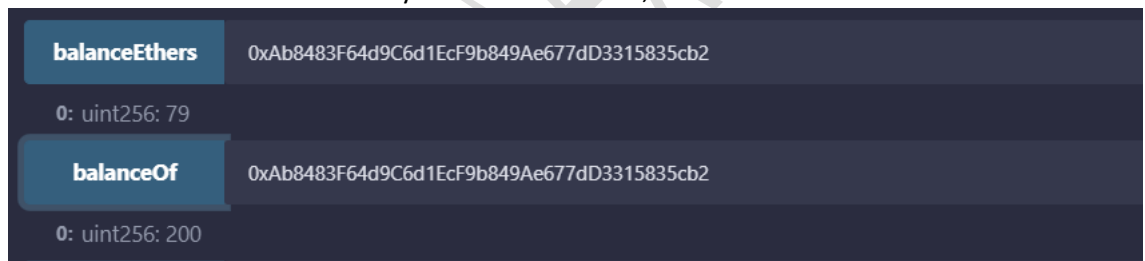
7. Balances del Smart Contract después de la compra del Inversor1



Venta de Tokens del Inversor1:

Partimos del hipotético de que el Inversor1, se ha arrepentido de su inversión y quiere devolver los 200 Tokens previamente adquiridos, al Smart Contract, o lo que es lo mismo, venderlos a la ICO

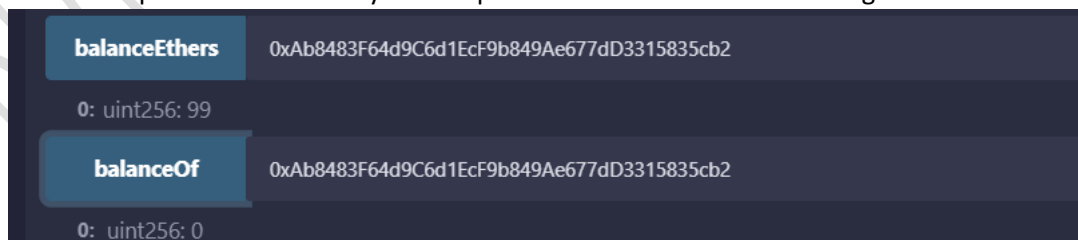
1. Balance de token y Ethers del Inversor1, antes de la venta.



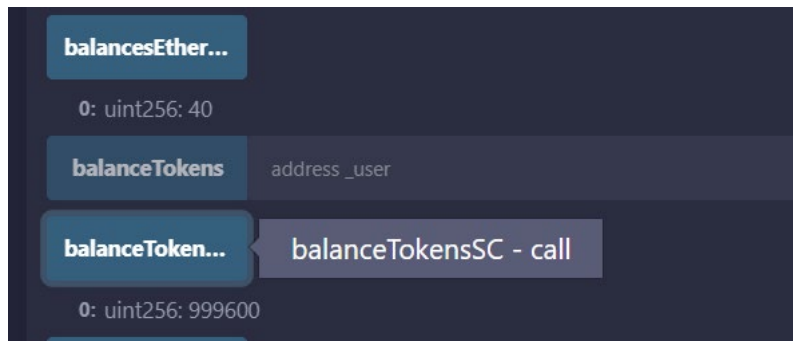
2. El inversor2 procede a la venta/devolución de los tokens a la ICO.



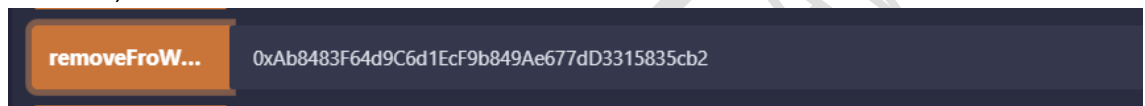
3. Balances de Ethers y Tokens del inversor1 tras la venta. Podemos observar cómo se ha quedado sin tokens y ha recuperado su cantidad de ethers original.



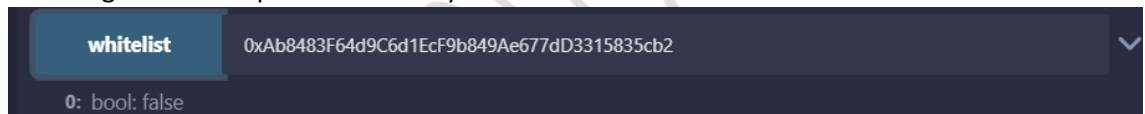
- Ahora, se puede apreciar cómo se han actualizado los balances de Ethers y Tokens del Smart Contract.



- Ya que el inversor1, ha quitado su inversión de la ICO, debemos eliminarlo de la Whitelist, utilizando la función 'removeFromWhitelist'.



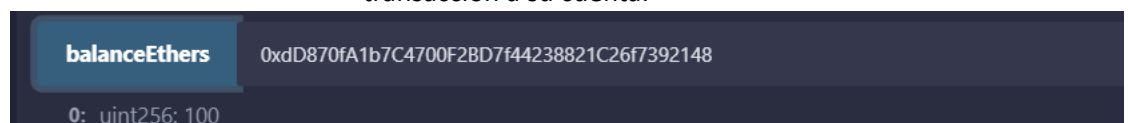
- Nos aseguramos de que el inversor1 ya no está en whitelist.



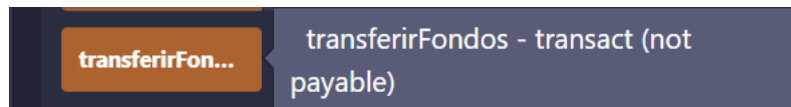
Transferencia de los fondos de la ICO a la cuenta escrow:

Al finalizar todas las transacciones de compra y venta de la ICO, el Admin/Owner debe transferir los fondos a la cuenta de seguridad asignada (escrow):

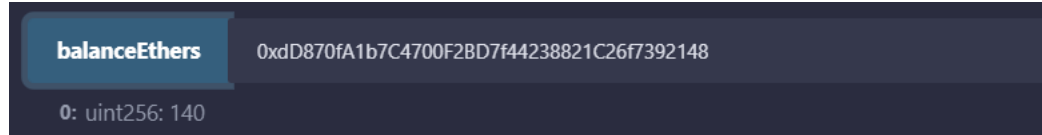
- Primero observemos cuál es el balance inicial del escrow, antes de cualquier transacción a su cuenta.



- Utilizamos la función 'transferirFondos', para transferir los fondos de la ICO a la cuenta escrow.



- Veamos el balance actualizado del escrow tras la transacción.



- Por tanto en la ICO ya no quedan fondos, como se puede apreciar.

