Tecnológico Nacional de México

Instituto Tecnológico de Veracruz

TRANSACCIONES COMPUTACIONALES CON BLOCKCHAIN

Grupo: AD21 9J8A

Alumna:

- Contreras Cruz Andrea - E17021540

Docente: Genaro Méndez López

Investigación:

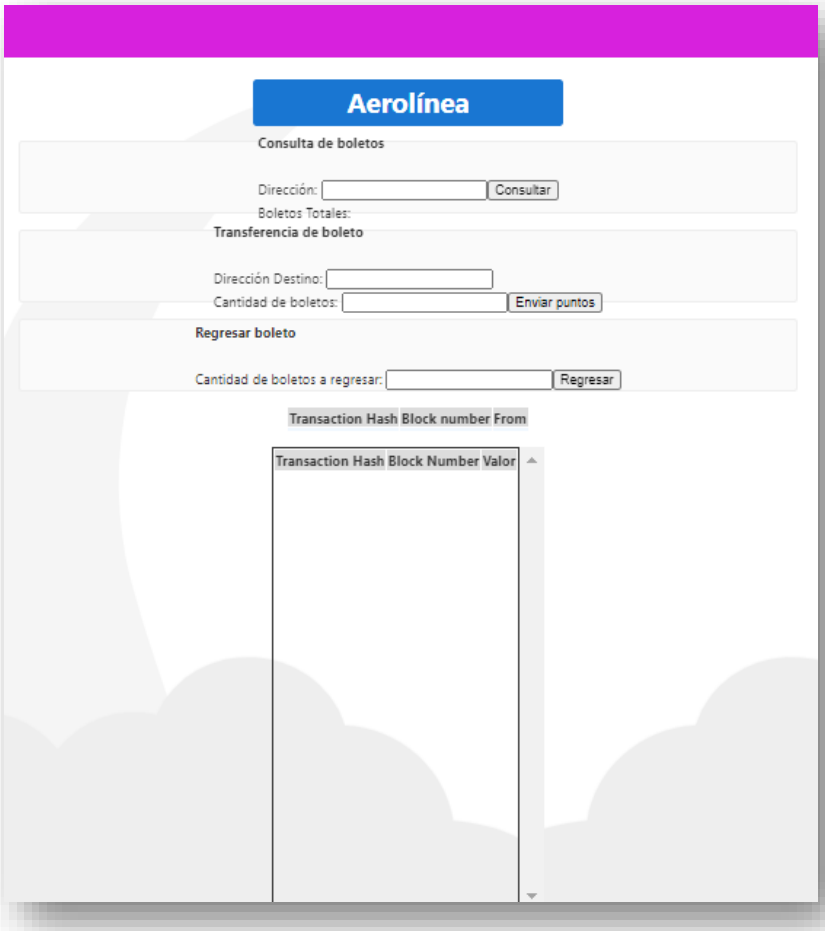
Documentación

H. Veracruz, Ver a 12 de Enero de 2021

Propuesta

El propósito de esta dapp es el de crear un programa que permita la compra de “vuelos” o boletos en una aerolínea, que, al mismo tiempo, permita regresarlos/cancelarlos. Cada que se realiza la venta de un vuelto se genera un token, que cuando cancelan u ocupan se regresa.

Interfaz



Métodos

- Consulta de boletos: Consultar la cantidad boletos disponibles.

```
async Balance(): Promise<void> {
  const addressDapp =  this.cantboleto.get('dircant')?.value;
  console.log(addressDapp);
  const accountBalance = await this.getBalanceByAccount(addressDapp);
```

```
    console.log(`accountBalance => ${accountBalance}`);
    this.balanceOf = accountBalance;
  }
```

Diseño:

```
  <div class="card">
    <form [formGroup]="cantboleto">
        <b><p class="tituloCarta">Consulta  de boletos</p></b>
        <br> Dirección: <input type="text" class="form-control"
formControlName="dircant">
        <button (click)="Balance()" class="button">Consultar</button>
        <br>
        <p>Boletos Totales: {{balanceOf}}</p>
    </form>
  </div>
```

- Transferencia de boleto: se le asigna un token al movimiento o compra del vuelo.

```
  async transferir(): Promise<void> {
    const address1 =  this.transboleto.get('recep')?.value;
    const numsgc =  this.transboleto.get('nump')?.value;

    this.web3s.contrato.methods.transfer(address1, numsgc).send({from:
this.web3s.accounts[0]})

    .then((response:any) => {
      this.resultado = "Transacción exitosa";
      this.blockHash = response.blockHash;
      this.blockNumber = response.blockNumber;
      this.from = response.from;
      this.transactionHash = response.transactionHash;
      this.web3s.contrato.methods.approve(address1, numsgc).send({from:
this.web3s.accounts[0]})
   })

   .catch((error: any) => {
      console.log(error);
      this.resultado = "Error";
   });
  }
```

Diseño:

```
  <div class="card">
    <form [formGroup]="transboleto">
      <b> <p class="tituloCarta">Transferencia de boleto</p></b>
        <br> Dirección Destino: <input type="text" class="form-control"
formControlName="recep">
        <br> Cantidad de boletos: <input type="text" class="form-control"
formControlName="nump">
        <button (click)="transferir()" class="button">Enviar puntos</button>
        <br>
    </form>
  </div>
```

- Regresar boleto: tiene dos funciones, el de regresar el token a la dirección de origen cuando se ocupe el boleto o cuando haya una cancelación.

```
  async regresar(): Promise<void> {
    const cuent1 =  '0x9Da3c0B8cF4774Bda306D3Ee54dfBEF59ba5d8D8';
    const cantboletos =  this.regresarboleto.get('cantboletos')?.value;

    this.web3s.contrato.methods.transfer(cuent1, cantboletos).send({from:
this.web3s.accounts[0]})

    .then((response:any) => {
      this.regresarr = "Voleto regresado con éxito";
      this.blockHash = response.blockHash;
      this.blockNumber = response.blockNumber;
      this.from = response.from;
      this.transactionHash = response.transactionHash;
      this.web3s.contrato.methods.approve(cuent1, cantboletos).send({from:
this.web3s.accounts[0]})
   })

   .catch((error: any) => {
      console.log(error);
      this.regresarr = "Error no se pudo regresar el boleto :(";
   });
  }
```

Diseño:

```
  <div class="card">
```

```
    <form [formGroup]="regresarboleto">
      <b>   <p class="tituloCarta">Regresar boleto</p></b>
        <br> Cantidad de boletos a regresar: <input type="text" class="form-
control" formControlName="cantboletos">
        <button (click)="regresar()" class="button">Regresar</button>
        <br>
        <p>{{regresarr}}</p>

    </form>
  </div>
```

Contrato

```
pragma solidity ^0.4.24;
// Sample token contract
//
// Symbol        : ACC
// Name          : ACC CRIPTOMONEDAS
// Total supply  : 999999
// Decimals      : 0
// Owner Account : 0x9Da3c0B8cF4774Bda306D3Ee54dfBEF59ba5d8D8
//
// Enjoy.
//
// (c) by Juan Cruz Martinez 2020. MIT Licence.
// ----------------------------------------------------------------------
// Lib: Safe Math
// ----------------------------------------------------------------------
---
contract SafeMath {

    function safeAdd(uint a, uint b) public pure returns (uint c) {
        c = a + b;
        require(c >= a);
    }

    function safeSub(uint a, uint b) public pure returns (uint c) {
        require(b <= a);
        c = a - b;
    }

    function safeMul(uint a, uint b) public pure returns (uint c) {
        c = a * b;
        require(a == 0 || c / a == b);
    }
```

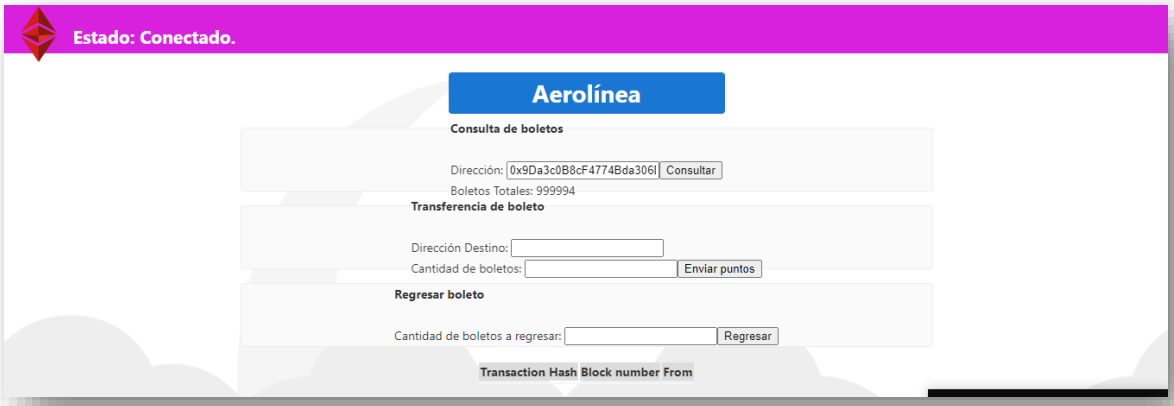
```
    function safeDiv(uint a, uint b) public pure returns (uint c) {
        require(b > 0);
        c = a / b;
    }
}
/**
ERC Token Standard #20 Interface
https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20-token-standard.md
*/
contract ERC20Interface {
    function totalSupply() public constant returns (uint);
    function balanceOf(address tokenOwner) public constant returns (uint
balance);
    function allowance(address tokenOwner, address spender) public constant
returns (uint remaining);
    function transfer(address to, uint tokens) public returns (bool
success);
    function approve(address spender, uint tokens) public returns (bool
success);
    function transferFrom(address from, address to, uint tokens) public
returns (bool success);

    event Transfer(address indexed from, address indexed to, uint tokens);
    event Approval(address indexed tokenOwner, address indexed spender, uint
tokens);
}
/**
Contract function to receive approval and execute function in one call
Borrowed from MiniMeToken
*/
contract ApproveAndCallFallBack {
    function receiveApproval(address from, uint256 tokens, address token,
bytes data) public;
}

/**
ERC20 Token, with the addition of symbol, name and decimals and assisted
token transfers
*/
contract ACCToken is ERC20Interface, SafeMath {
    string public symbol;
    string public  name;
    uint8 public decimals;
    uint public _totalSupply;
```

```
    mapping(address => uint) balances;
    mapping(address => mapping(address => uint)) allowed;
    // ------------------------------------------------------------------------
    // Constructor
    // ------------------------------------------------------------------------
    constructor() public {
        symbol = "ACC";
        name = "ACC CRIPTOMONEDAS";
        decimals = 0;
        _totalSupply = 999999;
        balances[0x9Da3c0B8cF4774Bda306D3Ee54dfBEF59ba5d8D8] = _totalSupply;
        emit Transfer(address(0),
0x9Da3c0B8cF4774Bda306D3Ee54dfBEF59ba5d8D8, _totalSupply);
    }
    // ------------------------------------------------------------------------
    // Total supply
    // ------------------------------------------------------------------------
    function totalSupply() public constant returns (uint) {
        return _totalSupply  - balances[address(0)];
    }
+
    // Get the token balance for account tokenOwner
    // ------------------------------------------------------------------------
---
    function balanceOf(address tokenOwner) public constant returns (uint
balance) {
        return balances[tokenOwner];
    }    // Transfer the balance from token owner's account to to account
    // - Owner's account must have sufficient balance to transfer
    // - 0 value transfers are allowed
    // ------------------------------------------------------------------------
---
    function transfer(address to, uint tokens) public returns (bool success)
{
        balances[msg.sender] = safeSub(balances[msg.sender], tokens);
        balances[to] = safeAdd(balances[to], tokens);
        emit Transfer(msg.sender, to, tokens);
        return true;
    }
    // Token owner can approve for spender to transferFrom(...) tokens
    // from the token owner's account
    //
    // https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20-token-
standard.md
```

```
    // recommends that there are no checks for the approval double-spend
attack
    // as this should be implemented in user interfaces
    // ------------------------------------------------------------------------
---
    function approve(address spender, uint tokens) public returns (bool
success) {
        allowed[msg.sender][spender] = tokens;
        emit Approval(msg.sender, spender, tokens);
        return true;
    }
    // Transfer tokens from the from account to the to account
    function transferFrom(address from, address to, uint tokens) public
returns (bool success) {
        balances[from] = safeSub(balances[from], tokens);
        allowed[from][msg.sender] = safeSub(allowed[from][msg.sender],
tokens);
        balances[to] = safeAdd(balances[to], tokens);
        emit Transfer(from, to, tokens);
        return true;
    }
    function allowance(address tokenOwner, address spender) public constant
returns (uint remaining) {
        return allowed[tokenOwner][spender];
    }
    function approveAndCall(address spender, uint tokens, bytes data) public
returns (bool success) {
        allowed[msg.sender][spender] = tokens;
        emit Approval(msg.sender, spender, tokens);
        ApproveAndCallFallBack(spender).receiveApproval(msg.sender, tokens,
this, data);
        return true;
    }
    function () public payable {
        revert();
    }
}
```

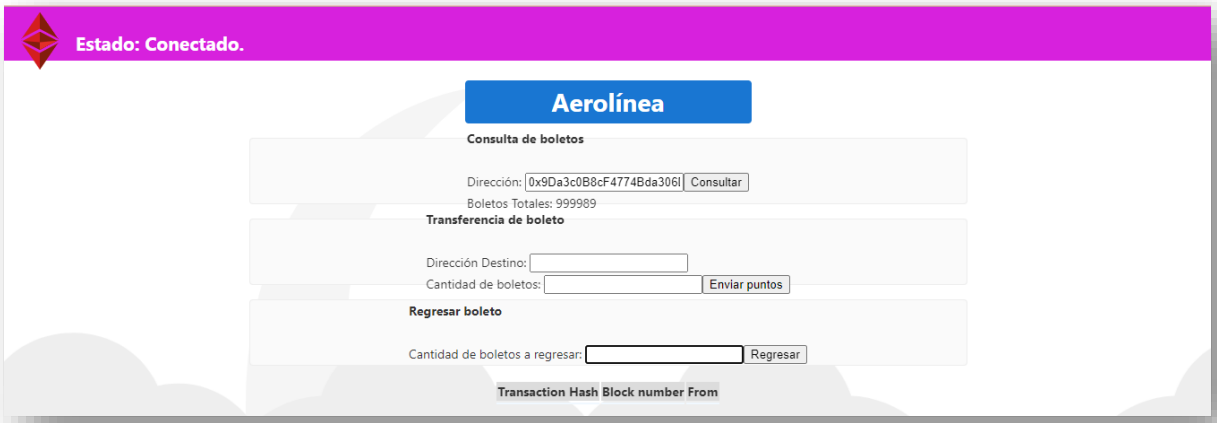
Pruebas

- Método 1



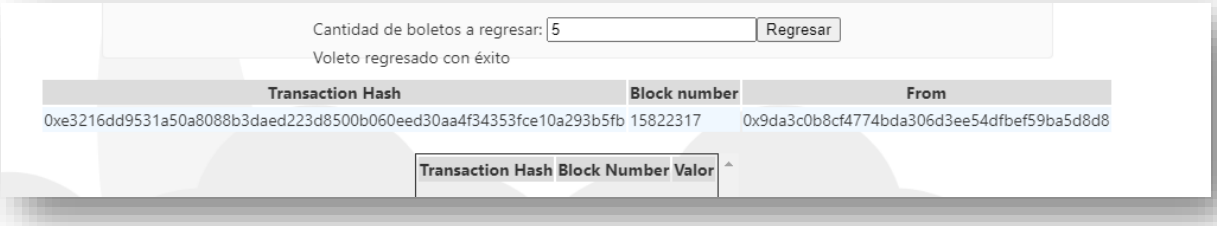
- Método 2

Comprobamos:


Estado: Conectado.
Aerolínea
Consulta de boletos
Dirección: 0x9Da3c0B8cF4774Bda306I
Consultar
Boletos Totales: 999989
Transferencia de boleto
Dirección Destino:
Cantidad de boletos:
Enviar puntos
Regresar boleto
Cantidad de boletos a regresar:
Regresar
Transaction Hash Block number From

- Método 3


Estado: Conectado.
Aerolínea
Consulta de boletos
Dirección: 0x9Da3c0B8cF4774Bda306I
Consultar
Boletos Totales: 999989
Transferencia de boleto
Dirección Destino:
Cantidad de boletos:
Enviar puntos
Regresar boleto
Cantidad de boletos a regresar: 5
Regresar
Transaction Hash Block number From
Transaction Hash Block Number Valor
TRANSFERIR
5 ACC
DETALLES
DATA
HEX
EDITAR
Estimated gas fee
$0.23 0.000478 BNB
Site suggested
Max fee: 0.00047799 BNB
$0.23
Total
5 ACC + 0.000478 BNB
Amount + gas fee
Max amount: 5 ACC + 0.000478 BNB
Rechazar
Confirmar


Cantidad de boletos a regresar: 5
Regresar
Voleto regresado con éxito
Transaction Hash
Block number
From
0xe3216dd9531a50a8088b3daed223d8500b060eed30aa4f34353fce10a293b5fb
15822317
0x9da3c0b8cf4774bda306d3ee54dfbef59ba5d8d8
Transaction Hash Block Number Valor