

Votaciones Claras

Objetivo

Realizar elecciones confidenciales y confiables utilizando la tecnología Blockchain, Por ello se requiere crear un Contrato Inteligente en Solidity que permita generar y administrar una votación entre varios candidatos.

Descripción del proyecto

Debido al mal manejo que se realiza alrededor del mundo en los sistemas de votaciones, se desea crear un sistema de votaciones eficaz y confiable utilizando la tecnología Blockchain.

Utilizando esta tecnología, se garantizará la realización de unas votaciones claras y transparentes a través de la creación de un Contrato Inteligente que administre el sistema de votaciones entre varios candidatos.

Este sistema contempla dos etapas :

- Alta de candidatos
- Votaciones generales

Las votaciones concluirán una vez que todos los votantes hayan emitido su voto por algún candidato. El sistema mostrará el nombre del ganador una vez terminada la contienda.

Alta de Candidatos

Este sistema permitirá el registro de los candidatos interesados en participar en esta elección.

Debido a que el puesto por el que concursarán los candidatos requiere un alto grado de especialización. Cada uno de los candidatos deberá subir la siguiente información personal para ser conocido por los votantes.

- ID
- Nombre

Elección de candidatos

Será decisión de cada uno de los votantes emitir su voto utilizando el ID del candidato de su preferencia en este periodo de votaciones. El voto será anónimo y secreto ya que ninguno de los candidatos podrá conocer la identidad de la persona que votó por él.

El Contrato Inteligente llevará a cabo el registro de las votaciones contabilizando cada uno de los votos emitidos y concluirá las votaciones una vez que todos los votantes hayan ejercido su derecho al voto.

El Contrato Inteligente, realizará el conteo de los votos por candidato y dará la victoria al candidato que tenga más votos acumulados.

Votaciones Claras

Es importante señalar que una vez que el votante emite su voto por el candidato de su preferencia, ya no será posible votar nuevamente o cambiar el voto emitido.

Diagramas

Diagrama General de Votaciones Claras

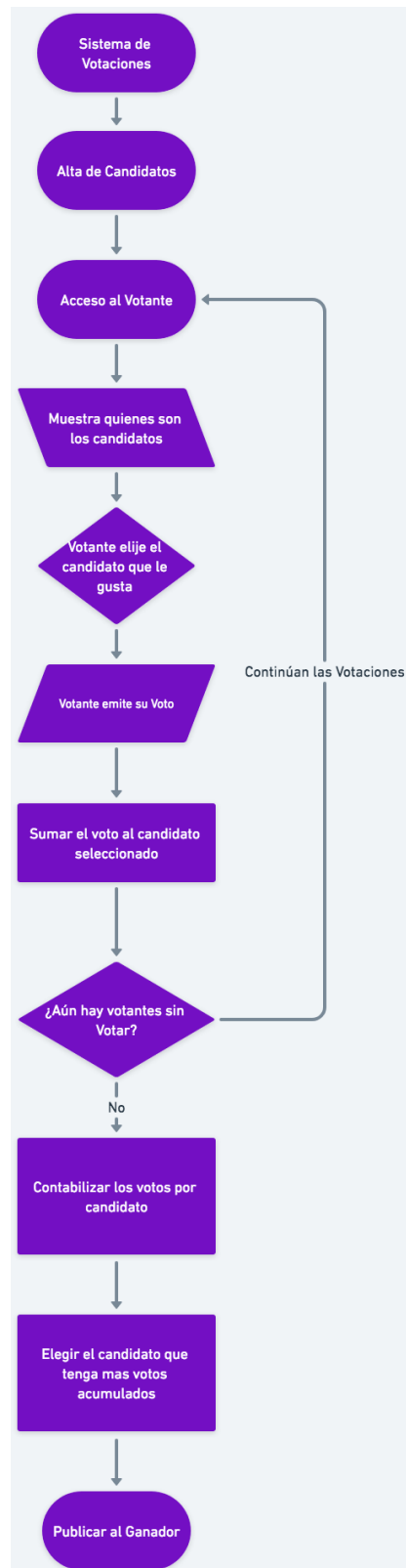


Diagrama de Alta de Candidatos



Votaciones Claras

Diagrama de Realización de las Elecciones



Votaciones Claras

Manual de Usuario

A continuación se detalla el funcionamiento del Contrato Inteligente para la realización de elecciones.

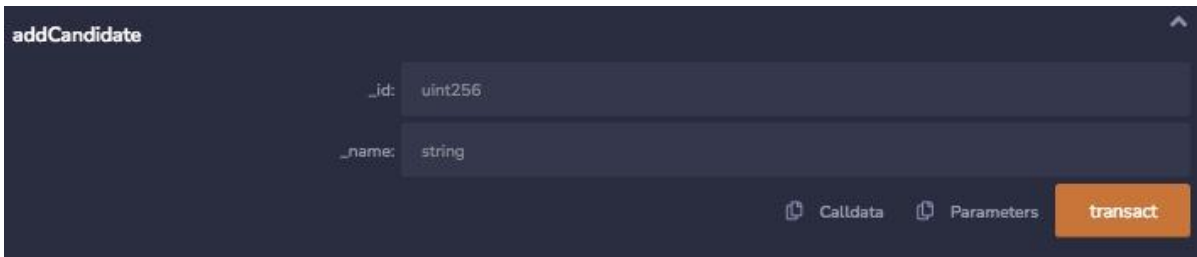
El contrato Inteligente perteneciente al Proyecto Votaciones Claras, deberá ser ejecutado utilizando Remix.

Siendo importante señalar que este contrato se ejecutará utilizando Sepolia como método para el pago de Gas y cada una de las personas interesadas en votar, deberán contar con una wallet activa en Metamask para poder emitir un Voto.

Alta de Candidatos

El Contrato de votaciones Claras permite el alta de los candidatos a ser elegidos. Para lograrlo, el candidato escribe su ID de registro único y su nombre en los campos mostrados.

A continuación, se da click en el botón “transact” para enviar el registro de este candidato a la Blockchain y que el registro quede guardado de forma inalterable.



Ventana Alta de Candidatos

El Botón “transact” hace un llamado a la Wallet de Metamask para solicitar la confirmación de la operación y el cobro de Gas correspondiente para esta operación.

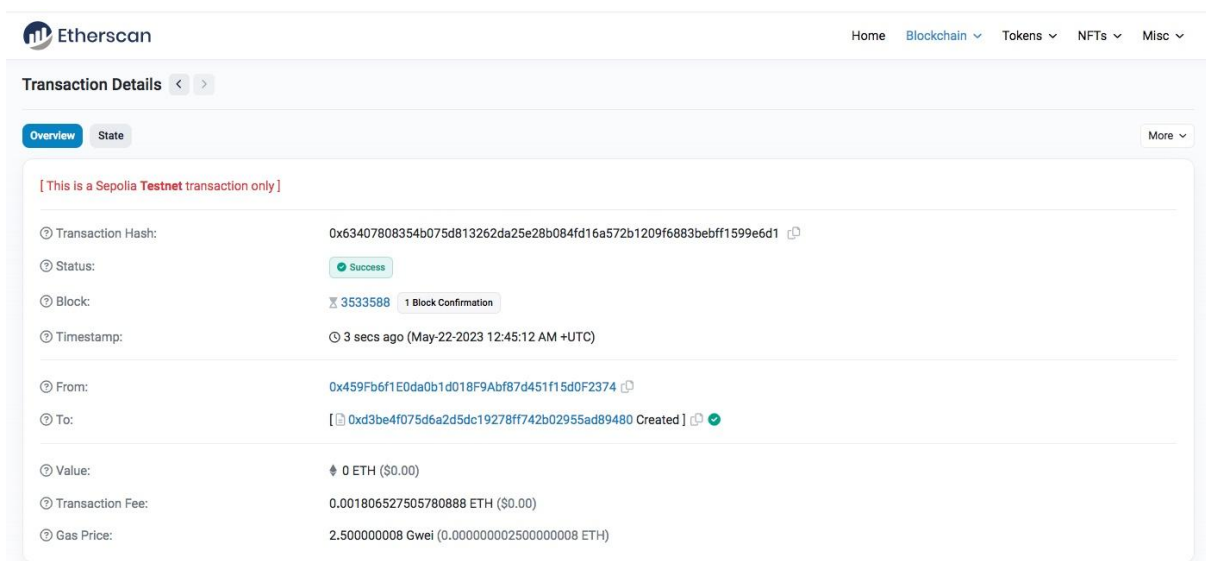
Votaciones Claras



Wallet de MetaMask

Una vez confirmada la operación, se procede a registrar la información dentro de la Blockchain.

Es importante señalar que todas las operaciones efectuadas pueden ser verificadas en etherScan para dar transparencia a la operación.



Ventana en EtherScan

Votaciones Claras

Consulta de Candidatos

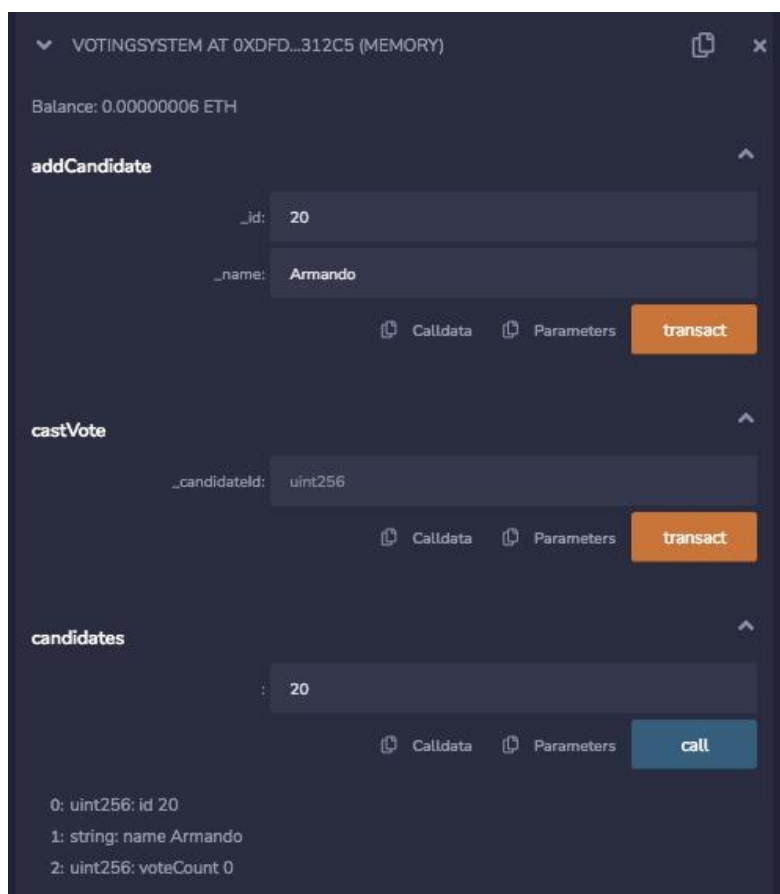
El Contrato permite consultar la información de los candidatos registrados utilizando su ID.

Para realizar esta búsqueda, se escribe en el campo el ID del candidato y el Contrato Inteligente buscará la información almacenada en la Blockchain.



Ventana Consulta de Candidatos

A continuación se muestra como se despliega la información de los candidatos:



Ejemplo de Información de Candidatos

Votaciones Claras

Emitir un Voto

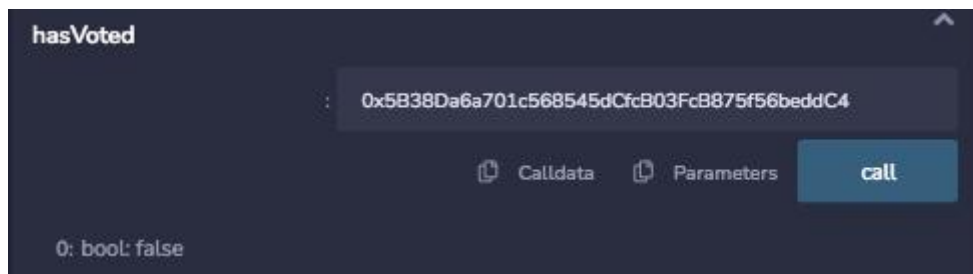
Para Votar dentro de este contrato es necesario contar con una wallet con Sepolias y poner el ID del candidato elegido para que el sistema cuente un voto a su favor. Dar clic en “transact” para la confirmación y el pago del gas correspondiente.



Ventana de Emisión de Voto

Comprobación de votos por persona

Una vez emitido el voto, la wallet utilizada en el proceso quedará registrada para que no pueda hacerse mal uso de ella y su estatus podrá ser consultado en la siguiente ventana:



Ventana de Validación

En el caso de que la persona quiera votar dos veces, el contrato le desplegará el siguiente mensaje:

```
transact to VotingSystem.castVote errored: VM error: revert.  
  
revert  
    The transaction has been reverted to the initial state.  
Reason provided by the contract: "You have already voted."  
Debug the transaction to get more information.
```

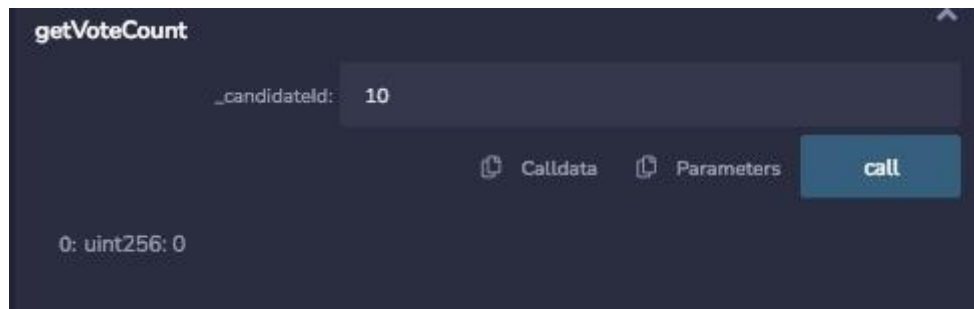
En el caso de que la persona quiera votar por algún candidato no registrado, el contrato le desplegará el siguiente mensaje:

Votaciones Claras

```
transact to VotingSystem.castVote errored: VM error: revert.  
  
revert  
    The transaction has been reverted to the initial state.  
Reason provided by the contract: "Invalid candidate ID."  
Debug the transaction to get more information.
```

Obtener el Conteo de los Votos

El Contrato lleva una sumatoria de los votos dados a cada candidato y para consultar esta información, debemos utilizar la ventana `getVoteCount` en la cual escribimos el ID del candidato para que el contrato nos muestre cuántos votos lleva registrados a su favor hasta el momento.



The screenshot shows the Remix IDE interface for the `getVoteCount` function. The function name is displayed at the top left. Below it, the parameter `_candidateId` is set to the value `10`. To the right of the input field are two icons: a clipboard for 'Calldata' and a document for 'Parameters'. A blue 'call' button is positioned to the right of these icons. At the bottom left, the output is shown as `0: uint256: 0`.

Conteo de Votos

Votaciones Claras

Contrato Inteligente

A continuación se encuentra el Código del contrato Inteligente desarrollado en Solidity para la realización de elecciones.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
contract VotingSystem {
    // Structure to represent a candidate
    struct Candidate {
        uint id;
        string name;
        uint voteCount;
    }

    // Mapping to store and retrieve candidates
    mapping(uint => Candidate) public candidates;

    // Mapping to track if an address has already voted
    mapping(address => bool) public hasVoted;

    // Event to notify when a vote is cast
    event VoteCasted(uint indexed candidateId);

    // Function to add a candidate
    function addCandidate(uint _id, string memory _name) public {
        candidates[_id] = Candidate(_id, _name, 0);
    }

    // Function to cast a vote for a candidate
    function castVote(uint _candidateId) public {
        require(!hasVoted[msg.sender], "You have already voted.");

        Candidate storage candidate = candidates[_candidateId];
        require(candidate.id != 0, "Invalid candidate ID.");

        candidate.voteCount++;
        hasVoted[msg.sender] = true;

        emit VoteCasted(_candidateId);
    }

    // Function to get the vote count of a candidate
    function getVoteCount(uint _candidateId) public view returns (uint) {
        return candidates[_candidateId].voteCount;
    }
}
```

Votaciones Claras

