

TECHFEST 2018

02-03 DE MAYO

ASEGURANDO LA PARTICIPACIÓN CIUDADANA CON BLOCKCHAIN



Herve Falciani



José L. de la Rosa



José M. Calabuig



GENERALITAT
VALENCIANA
Conselleria de Transparència,
Responsabilitat Social,
Participació i Cooperació



CÀTEDRA TRANSPARÈNCIA
I GESTIÓ DE DADES



UNIVERSITAT
POLITÀCNICA
DE VALÈNCIA

MUGI



etsinf

LUGAR:
Sala de reuniones 1H
ETSINF - UPV

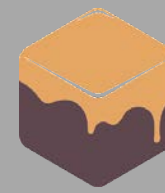
Hervé Falciani

Pep Lluís de la Rosa

Jose M. Calabuig



Ganache sin terminal



<http://truffleframework.com/ganache/>

El puerto
por defecto
es el
7545

Ganache			
ACCOUNTS	BLOCKS	TRANSACTIONS	LOGS
CURRENT BLOCK 0 GAS PRICE 20000000000 GAS LIMIT 314159200000 NETWORK ID 5777 RPC SERVER HTTP://127.0.0.1:7545 MINING STATUS AUTOMINING			
MNEMONIC moment suggest skin know blush toe ladder border avoid federal armor rhythm		HD PATH m/44'/60'/0'/0'/0/account_index	
ADDRESS 0x30fAf164170CD5dd55EAFfAe3984aAA9539D35b7	BALANCE 100.00 ETH	TX COUNT 0	INDEX 0
ADDRESS 0xdC545F47c063F81d81252C01360DA66e40Fe50B4	BALANCE 100.00 ETH	TX COUNT 0	INDEX 1
ADDRESS 0xA64b283DaD66202649F81285415f0b292AB34418	BALANCE 100.00 ETH	TX COUNT 0	INDEX 2
ADDRESS 0x24785777Ee852C54d86a3Eb9647c092632f57657	BALANCE 100.00 ETH	TX COUNT 0	INDEX 3
ADDRESS 0xFd7E798028d6a16db622117c45D29159eA63Fdf9	BALANCE 100.00 ETH	TX COUNT 0	INDEX 4
ADDRESS 0xF4b29007ba37B3a26BFB3dB7594d0B3DA9838fa5	BALANCE 100.00 ETH	TX COUNT 0	INDEX 5
ADDRESS 0xC66D61d56716318053d26b7CC1566F685d598B23	BALANCE 100.00 ETH	TX COUNT 0	INDEX 6

Contratos desde Remix



<https://remix.ethereum.org>

Remix - Solidity IDE

Es seguro <https://remix.ethereum.org/#optimize=false&version=builtin>

Aplicaciones Visualizaciones R Whitepapers <https://www.analyti...> Blockchain Remix - Solidity IDE Practicas UPV

browser/ballot.sol

Compile Run Settings Analysis Debugger Support

Environment Web3 Provider Custom (1524993994747)

Account 0x2c7...b8d64 (99.9999999999987925)

Gas limit 3000000

Value 0 wei

Create

Load contract from Address At Address

0 pending transactions

0 contract instances

```
1 pragma solidity ^0.4.11;
2
3 /// @title Votación con voto delegado
4 contract Ballot {
5     // Declara un nuevo tipo de dato complejo, que será
6     // usado para almacenar variables.
7     // Representará a un único votante.
8     struct Voter {
9         uint weight; // el peso del voto se acumula mediante la delegación de votos
10         bool voted; // true si esa persona ya ha votado
11         address delegate; // persona a la que se delega el voto
12         uint vote; // índice de la propuesta votada
13     }
14
15     // Representa una única propuesta.
16     struct Proposal {
17         bytes32 name; // nombre corto (hasta 32 bytes)
18         uint voteCount; // número de votos acumulados
19     }
20
21     address public chairperson;
22
23     // Declara una variable de estado que
24     // almacena una estructura de datos `Voter` para cada posible dirección.
25     mapping(address => Voter) public voters;
26
27     // Una matriz dinámica de estructuras de datos de tipo `Proposal`.
28     Proposal[] public proposals;
29
30     // Crea una nueva votación para elegir uno de los `proposalNames`.
31     function Ballot(bytes32[] proposalNames) {
32         chairperson = msg.sender;
33         voters[chairperson].weight = 1;
34
35         // Para cada nombre propuesto
36         // crea un nuevo objeto de tipo Proposal y lo añade
37         // al final del array.
38         for (uint i = 0; i < proposalNames.length; i++) {
39             // `Proposal({...})` crea un nuevo objeto de tipo Proposal
40             // de forma temporal y se añade al final de `proposals`
41             // mediante `proposals.push(...)`.
42             proposals.push(Proposal({
43                 name: proposalNames[i],
44                 voteCount: 0
45             }));
46         }
47     }
48 }
```

[2] only remix transactions, script

Search transactions Listen on network

En la pestaña
Run
selecciona
Web3 Provider
con el
puerto

Mejorando el código



```
pragma solidity ^0.4.11;
```

```
// Votación con voto delegado
```

```
contract Ballot {
```

```
// Único votante.
```

```
struct Voter {
```

```
    uint weight; // el peso del voto se acumula mediante la delegación de votos
```

```
    bool voted; // true si esa persona ya ha votado
```

```
    address delegate; // persona a la que se delega el voto
```

```
    uint vote; // índice de la propuesta votada
```

```
}
```

```
// Única propuesta.
```

```
struct Proposal {
```

```
    bytes32 name; // nombre corto (hasta 32 bytes)
```

```
    uint voteCount; // número de votos acumulados
```

```
}
```

Mejorando el código



```
address public chairperson;
```

```
// Variable de estado que almacena una estructura de datos
```

```
// `Voter` para cada posible dirección.
```

```
mapping(address => Voter) public voters;
```

```
// Matriz dinámica de estructuras de datos de tipo `Proposal`.
```

```
Proposal[] public proposals;
```

Mejorando el código



```
// Creamos una nueva votación para elegir uno de los `proposalNames`.
```

```
function Ballot(bytes32[] proposalNames) {
```

```
    chairperson = msg.sender;
```

```
    voters[chairperson].weight = 1;
```

```
// Para cada nombre propuesto crea un nuevo objeto de tipo Proposal
```

```
// y lo añadimos al final del array.
```

```
    for (uint i = 0; i < proposalNames.length; i++) {
```

```
        // `Proposal({...})` crea un nuevo objeto de tipo Proposal
```

```
        // de forma temporal y se añade al final de `proposals`
```

```
        // mediante `proposals.push(...)`.
```

```
        proposals.push(Proposal({
```

```
            name: proposalNames[i],
```

```
            voteCount: 0
```

```
        }));
```

```
    }
```

```
}
```

Mejorando el código



```
// Proporciona a `voter` el derecho a votar en esta votación.
```

```
// Sólo puede ser ejecutado por `chairperson`.
```

```
function giveRightToVote(address voter) {
```

```
    // Si el argumento de `require` da como resultado `false`,
```

```
    // finaliza la ejecución y revierte todos los cambios
```

```
    // producidos en el estado y los balances de Ether.
```

```
    // A veces es buena idea usar esto por si las funciones
```

```
    // están siendo ejecutadas de forma incorrecta.
```

```
    //Pero ten en cuenta
```

```
    // que de esta forma se consumirá todo el gas enviado
```

```
    // (está previsto que esto cambie en el futuro).
```

```
    require((msg.sender == chairperson) && !voters[voter].voted);
```

```
    voters[voter].weight = 1;
```

```
}
```


Mejorando el código



```
// Delega tu voto a `to`.
```

```
function delegate(address to) {
```

```
    // Asignación por referencia
```

```
    Voter sender = voters[msg.sender];
```

```
    require(!sender.voted);
```

```
    // No se permite la delegación a uno mismo.
```

```
    require(to != msg.sender);
```

```
    // Propaga la delegación en tanto que `to` también delegue.
```

```
    // Por norma general, los bucles son muy peligrosos porque si tienen muchas iteraciones
```

```
    // puede darse el caso de que empleen más gas del disponible en un bloque.
```

```
    // En este caso, eso implica que la delegación no será ejecutada pero en otros casos
```

```
    //puede suponer que un contrato se quede completamente bloqueado.
```

```
    while (voters[to].delegate != address(0)) {
```

```
        to = voters[to].delegate;
```

```
        // Encontramos un bucle en la delegación. No está permitido.
```

```
        require(to != msg.sender);
```

```
    }
```


Mejorando el código



```
// Dado que `sender` es una referencia, esto
// modifica `voters[msg.sender].voted`
sender.voted = true;
sender.delegate = to;
Voter delegate = voters[to];
if (delegate.voted) {

    // Si la persona en la que se ha delegado el voto ya ha votado,
    // se añade directamente al número de votos.
    proposals[delegate.vote].voteCount += sender.weight;
} else {

    // Si la persona en la que se ha delegado el voto
    // todavía no ha votado, se añade al peso de su voto.
    delegate.weight += sender.weight;
}
}
```

Mejorando el código



```
// Da tu voto (incluyendo los votos que te han delegado)
// a la propuesta `proposals[proposal].name`.
function vote(uint proposal) {
    Voter sender = voters[msg.sender];
    require(!sender.voted);
    sender.voted = true;
    sender.vote = proposal;

    // Si `proposal` está fuera del rango de la matriz,
    // esto lanzará automáticamente una excepción y
    // se revocarán todos los cambios
    proposals[proposal].voteCount += sender.weight;
}
```

Mejorando el código



```
// Calcula la propuesta ganadora teniendo en cuenta
// todos los votos realizados.
function winningProposal() constant
    returns (uint winningProposal)
{
    uint winningVoteCount = 0;
    for (uint p = 0; p < proposals.length; p++) {
        if (proposals[p].voteCount > winningVoteCount) {
            winningVoteCount = proposals[p].voteCount;
            winningProposal = p;
        }
    }
}
```

Mejorando el código



```
// Llama a la función winningProposal() para obtener
// el índice de la propuesta ganadora y así luego devolver el nombre.
function winnerName() constant
    returns (bytes32 winnerName)
{
    winnerName = proposals[winningProposal()].name;
}
}
```

okay

THANKYOU

