Práctica 4

Ejercicio 1: ¡A votar!

Implementación de un sistema de votación:

1.1 Contrato DhondtElectionRegion

```
contract DhontElectionRegion ∢
uint immutable private regionId;
mapping(uint => uint) private weights;
uint[] internal results;
savedRegionInfo();
   regionId = _regionId;
   results = new uint[](_nPartidos);
weights[28] = 1; // Madrid
   weights[8] = 1; // Barcelona
   weights[41] = 1; // Sevilla
   weights[44] = 5; // Teruel
   weights[42] = 5; // Soria
   weights[49] = 4; // Zamora
   weights[9] = 4; // Burgos
   weights[29] = 2; // Malaga
function registerVote(uint partido) internal returns (bool) {  □ infinite gas
   if (partido < 0 || results.length <= partido)</pre>
   results[partido] += weights[regionId];
```

1.2 Contrato PollingStation

```
abstract contract PollingStation {
bool public votingFinished;
bool private votingOpen;
address private president;
president = _president;
   votingOpen = false;
   votingFinished = false;
modifier execPresident {
  require(msg.sender == president, "error (execPresident)");
modifier isVotingOpen {
   require(votingOpen, "error (isVotingOpen)");
   _;
function openVoting() external execPresident {
                                          🖺 - gas
   require(!votingOpen, "error (openVoting)");
   votingOpen = true;
require(!votingFinished, "error (closeVoting)");
   votingOpen = false;
   votingFinished = true;
function castVote(uint id) external virtual {}
function getResults() external view virtual returns(uint[] memory) {}
```

1.3 Contrato DhondtPollingStation

1.4 Contrato Election

Ejercicio 2:

2.1

Role	Address		
Authority	0x5B38Da6a701c568545dCfcB03FcB875f56beddC4		
President 1	0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2		
President 2	0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db		
Voter 1	0x78731D3Ca6b7E34aC0F824c42a7cC18A495cabaB		
Voter 2	0x617F2E2fD72FD9D5503197092aC168c91465E7f2		
Voter 3	0x17F6AD8Ef982297579C203069C1DbfFE4348c372		
Voter 4	0x5c6B0f7Bf3E7ce046039Bd8FABdfD3f9F5021678		
Voter 5	0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7		

Creamos un contrato Election con la dirección de Authority y 3 partidos:

2.3

Creamos las dos sedes electorales, la primera en Madrid y la segunda en Teruel mediante la función **createPollingStation** con las direcciones del president 1 y el president 2.

President1: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2

Sede: Madrid Id: 28

Dirección de la sede creada: 0x18bfF3b399EAcBBcFea6709053ad1709fc073690

President2: 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db

Sede: Teruel Id: 44

Dirección de la sede creada: 0x1Eae946CaB3Cfa4cAf87226bf6886208619C0948

2.4.

Cada presidente llama a la función openVoting:

Transacción del President1 para openVoting:



Transacción del President2 para openVoting:

2.5

Emitimos un voto con cada una de las siguientes direcciones mediante la función castVote:

Voter1: Partido 1 en Madrid.

Voter2: Partido 0 en Teruel.

Voter3: Partido 2 en Madrid.

Voter4: Partido 1 en Madrid.

President1: Partido 1 en Madrid.

Voter5: Partido 0 en Teruel.

President2: Partido 1 en Teruel.

2.6

Llamamos a la función getResults con la dirección de la Authority:

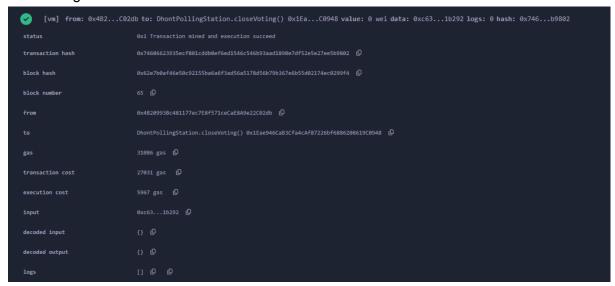
Se puede ver que, como en las sedes no han cerrado las votaciones, se revierte la llamada.

Cerramos las votaciones con ambos presidentes con la función closeVoting:

closeVoting de Presidente 1:



closeVoting de Presidente 2:



Llamamos a la función getResults con la dirección de uno de los presidentes en su sede, llamamos al president1 que muestra los votos que se han obtenido en Madrid.

2.9

Llamamos a la función getResults con la dirección de la autoridad administrativa:

Nos queda que los resultados generales son:

Partido	Resultado	
0	10	
1	8	
2	1	