

# Memoria Laboratorio 2 BC

## Ejercicio 2

María Andrea Ugarte Valencia

Bahdon Barkhad

Marcos Villar Avión

# ÍNDICE

<b>1 - Definición del Escenario.....</b>	<b>3</b>
<b>2 - Diseño.....</b>	<b>3</b>
2.1 - Casos de Uso.....	3
2.2 - Contrato inteligente.....	6
2.2.1 - Variables globales.....	6
2.2.2 - Funciones y modifiers.....	6
2.3 - React App.....	7
2.4 - Usuarios del sistema.....	7
<b>3 - Implementación del smart contract.....</b>	<b>8</b>
<b>4 - Pruebas.....</b>	<b>8</b>
4.1 - Subir un censo.....	8
4.2 - Añadir Administradores.....	9
4.3 - Eliminar Candidatos.....	10
4.4 - Asignar una zona específica a un censo.....	10
4.5 - Mostrar el censo de una zona específica.....	10
4.6 - Observación.....	11

# 1 - Definición del Escenario

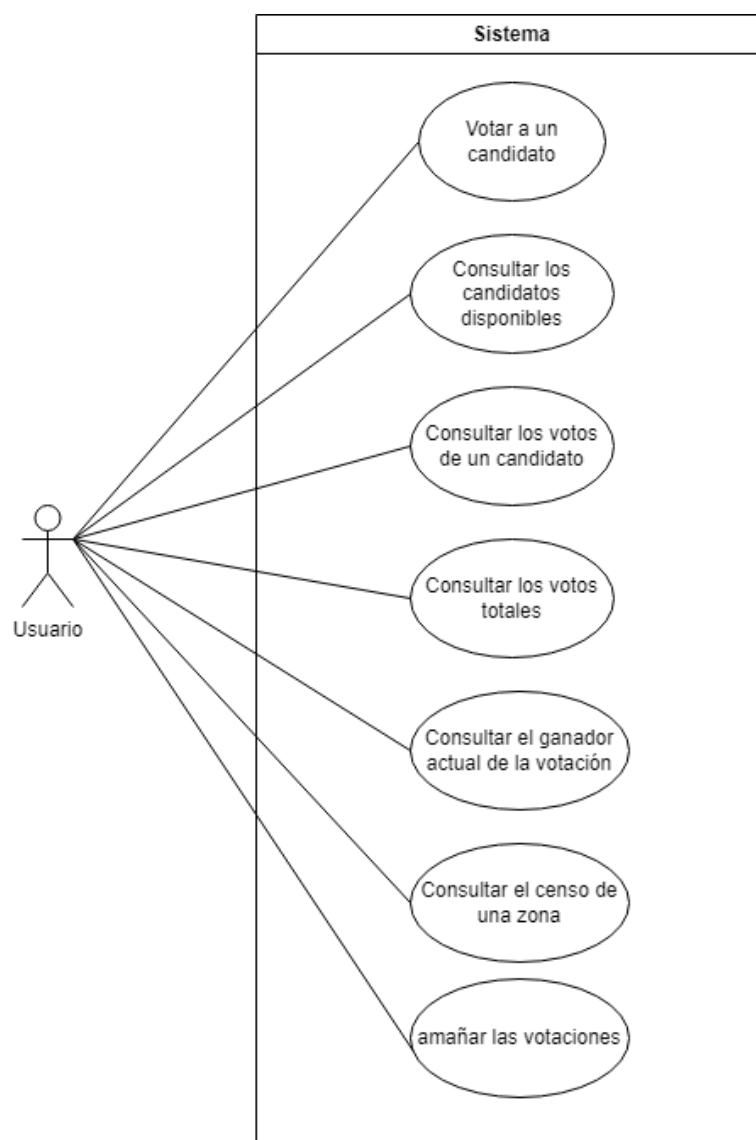
Para esta práctica, hemos creado un sistema de censo en el que un administrador puede cargar un censo en la plataforma con los nombres de los votantes asignados a una zona específica.

## 2 - Diseño

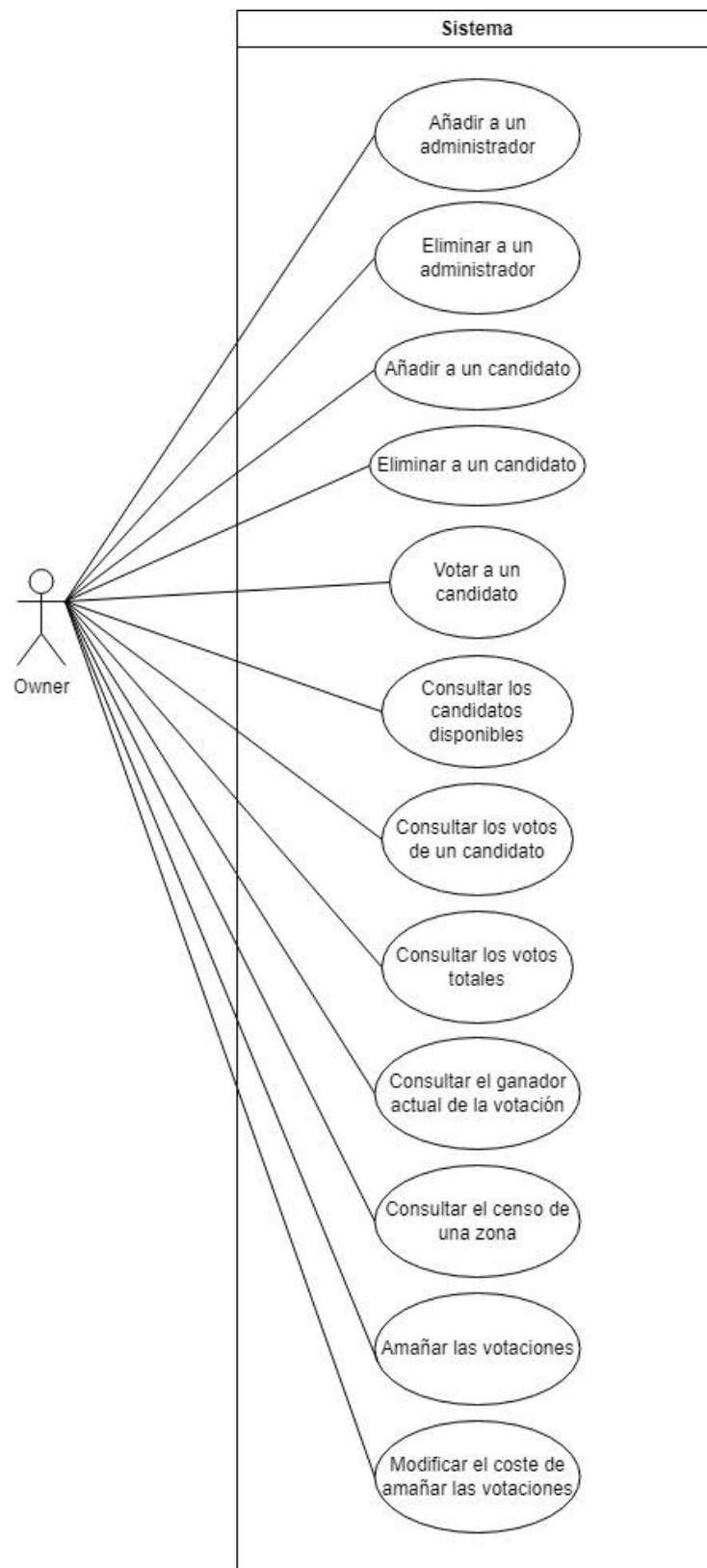
### 2.1 - Casos de Uso

Dado a la implementación de esta nueva funcionalidad, los casos de uso se han actualizado:

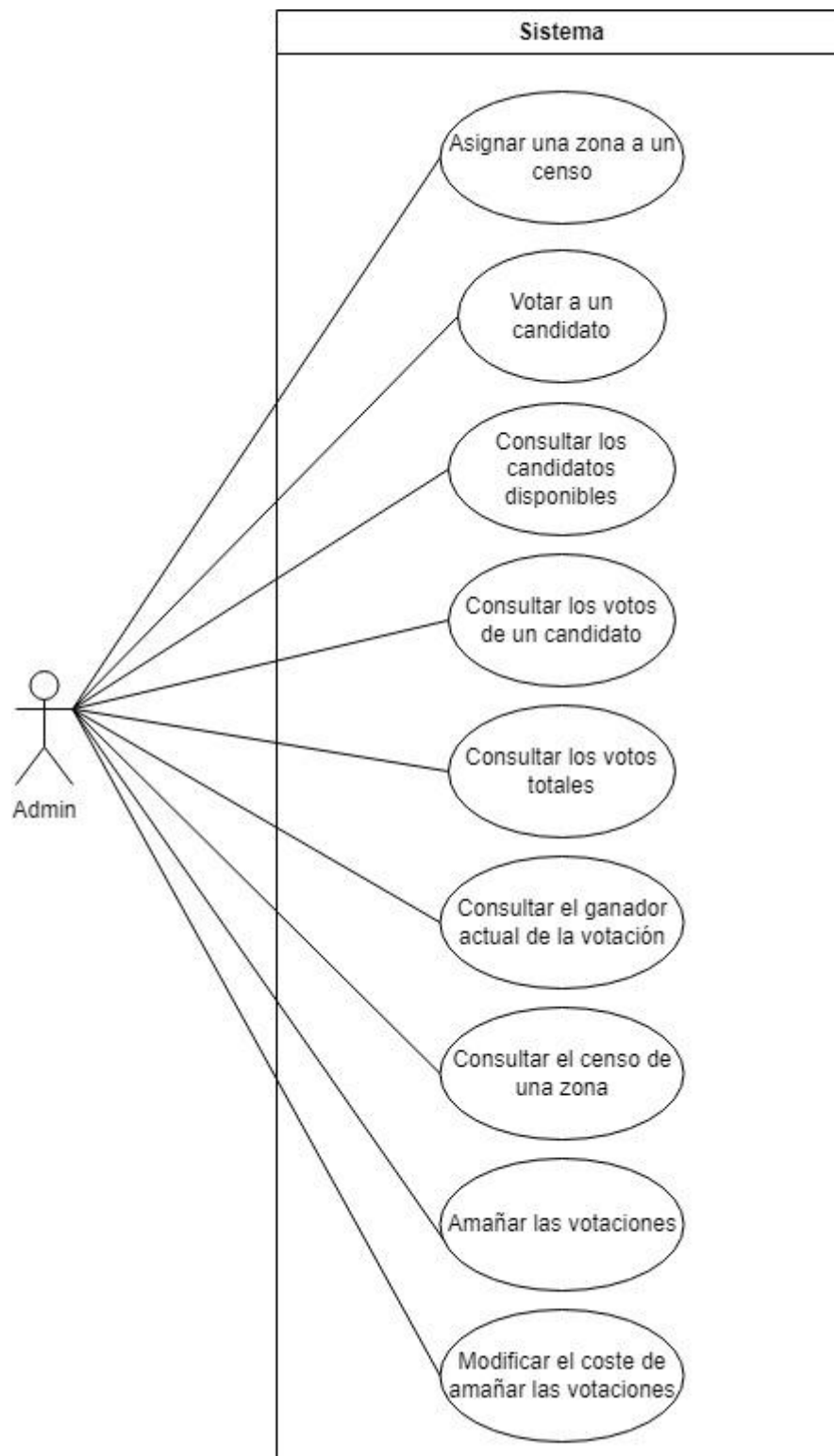
- Votante



- Owner



- Administrador



## 2.2 - Contrato inteligente

### 2.2.1 - Variables globales

Se han creado nuevas variables globales para este contrato:

<u>Variable global</u>	<u>Descripción</u>
isAdmin	Diccionario/Mapping para guardar la información de si un usuario es administrador
censusForZone	Diccionario/Mapping para guardar una relación entre un censo y su zona correspondiente

### 2.2.2 - Funciones y modifiers

Se han creado nuevas funciones para alcanzar los nuevos objetivos, al igual que nuevos modificadores. En la siguiente tabla se especifican los modificadores utilizados:

<u>Modificador</u>	<u>Descripción</u>
onlyAdmin	Para verificar que un usuario es administrador

En la siguiente tabla se especifican las funciones implementadas para esta práctica y una breve descripción de ellas:

<u>Nombre</u>	<u>Funcionalidad</u>	<u>Control de Acceso</u>
addAdmin()	Añade un administrador.	Owner
removeAdmin()	Elimina un administrador.	Owner
addCensusForZone()	Permite a un administrador asignar una zona a un censo.	Admin
getCensusForZone()	Muestra el hash del censo asignado a una zona específica.	

## 2.3 - React App

Para poder adaptar el ejercicio 1 a nuestro smart contract tuvimos que modificar los siguientes ficheros: **dapp\_ipfs/src/contracts/src/addresses.js** y **dapp\_ipfs/src/App.js**

En addresses.js tuvimos que actualizar la dirección con la de nuestro contrato.

```
contract address 0x0b4d3e6827d8b96102e8aaeb37bf26bb6515f0f7
```

```
const addresses = {  
  ipfs: "0x0b4d3e6827d8b96102e8aaeb37bf26bb6515f0f7",  
};  
export default addresses;
```

En App.js hemos modificado el siguiente bloque de código para obtener los hash de los censos:

```
//contract = new ethers.Contract(address, abi, defaultProvider);  
async function readCurrentUserFile() {  
  try {  
    const result = await ipfsContract.getCensusForZone(  
      defaultProvider.getSigner().getAddress()  
    );  
    console.log({ result });  
    return result;  
  } catch (error) {  
    console.error("Error fetching census for zone:", error.message);  
    return null;  
  }  
}
```

También se han hecho modificaciones en el diseño de la interfaz para que resulte más coherente con nuestro smart contract.

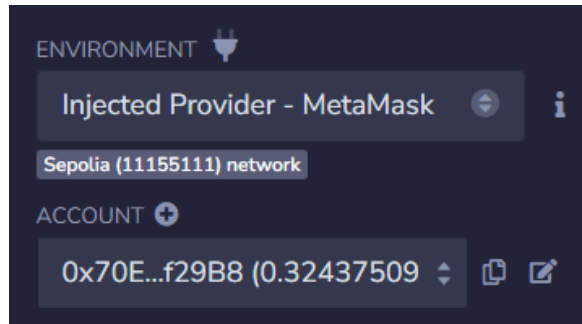
## 2.4 - Usuarios del sistema

En nuestro sistema se ha añadido un nuevo usuario, el **administrador**.

La función del administrador es trabajar en la gestión del sistema de votaciones. En este caso, se encarga de la subida y asignación de los censos a sus respectivas zonas.

## 3 - Implementación del smart contract

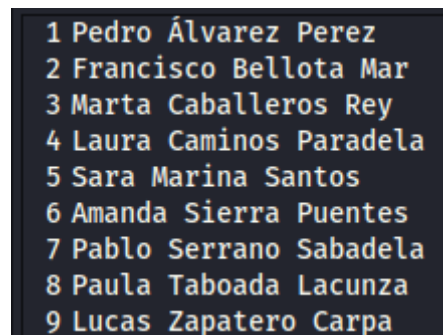
Para ejecutar el *smart contract* desarrollado, necesitaremos desplegarlo con nuestra cuenta de Metamask. Esto nos dará la capacidad de realizar operaciones que conlleven algún tipo de costo.



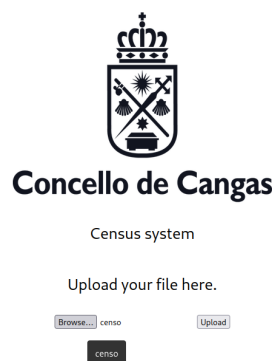
## 4 - Pruebas

### 4.1 - Subir un censo

Contamos con el censo de los votantes:

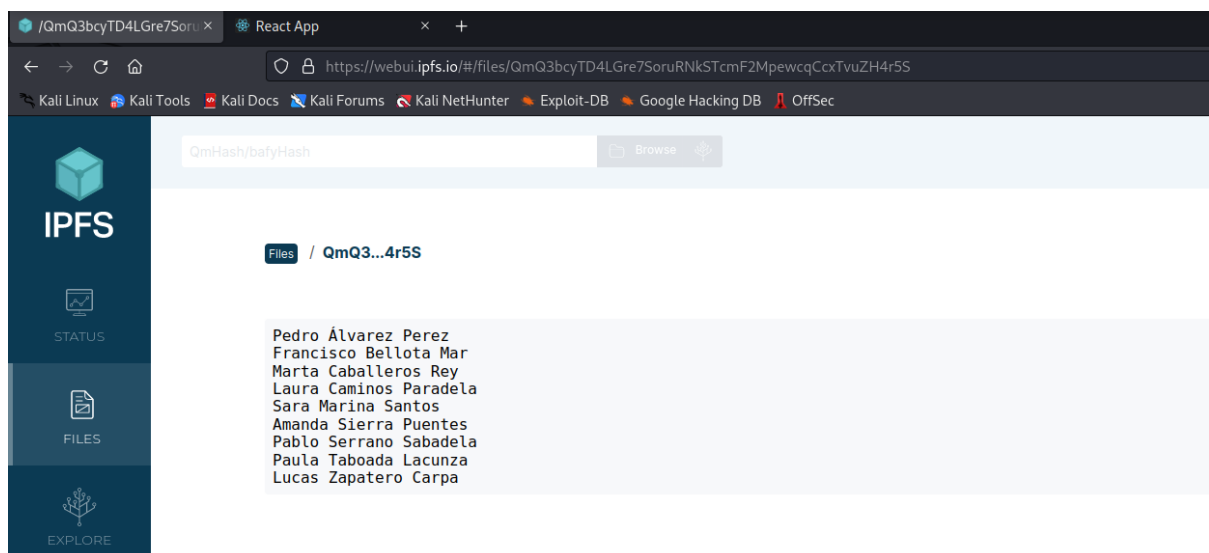
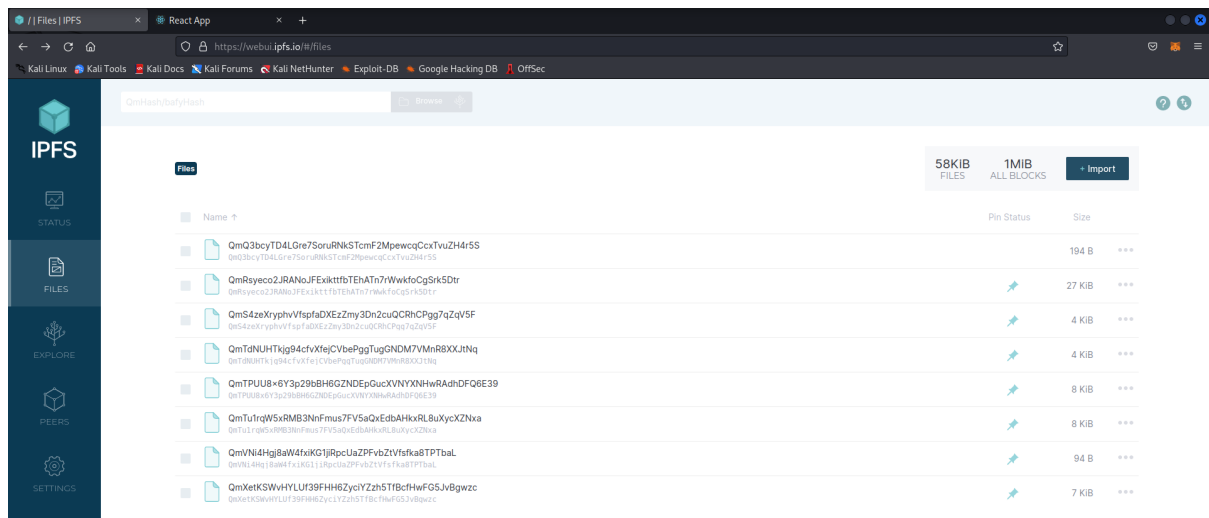


Desde nuestra interfaz, podremos subir el censo que se guardará como archivo IPFS:



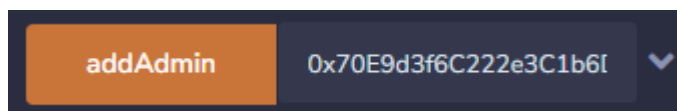


Podemos visualizar el archivo en <https://webui.ipfs.io/#/files>:

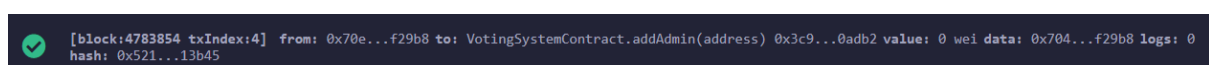


## 4.2 - Añadir Administradores

Añadimos a un administrador para que pueda realizar la gestión de censos:

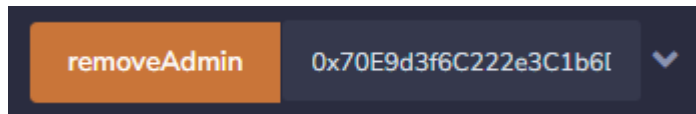


La operación se realiza correctamente:

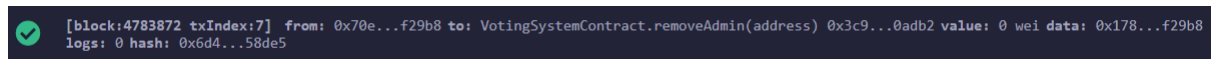


## 4.3 - Eliminar Candidatos

Eliminamos a un administrador:

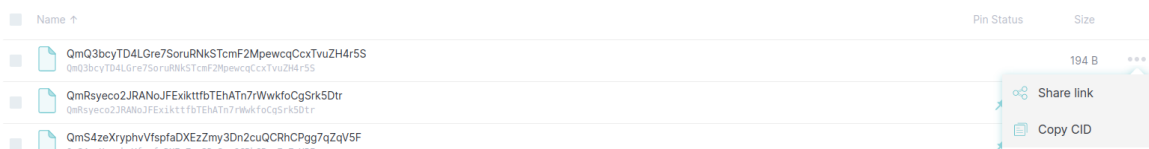


El administrador se elimina con éxito:

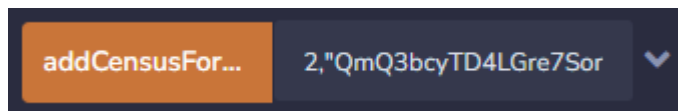


## 4.4 - Asignar una zona específica a un censo

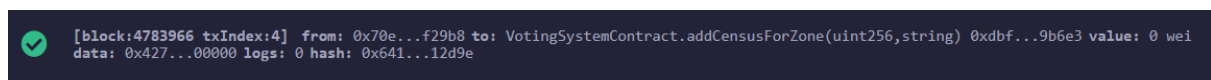
Copiamos el CID del censo:



Y desde el rol de administrador le asignamos una zona específica:

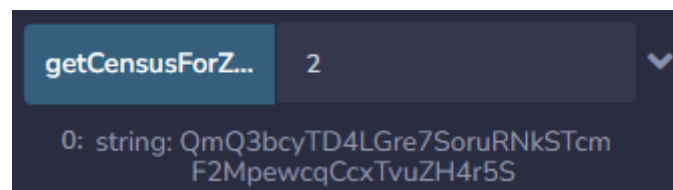


La asignación se ha realizado de manera exitosa.



## 4.5 - Mostrar el censo de una zona específica

Gracias al id de una zona, podemos ver el hash de su censo correspondiente:



## 4.6 - Observación

Aquí podemos observar la notificación de metamask cada vez que realicemos una acción que modifique nuestro contrato:

