GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

RECUENTO DE VOTOS ELECTRÓNICO UTILIZANDO LA RED DE ETHEREUM, UNA APLICACIÓN BLOCKCHAIN

Autor: Daniel Ortega Núñez

Director: Atilano Fernández-Pacheco Sánchez-Migallón

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

"Recuento electrónico de votos utilizando la red de Ethereum, una aplicación Blockchain"

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2020/21 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.

Fdo.: Daniel Ortega Núñez Fecha: 10/07/2021

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Atilano Fernández-Pacheco Sánchez-Migallón

Digitally signed by Atilano Fernández-Pacheco Sánchez-Migallón
Date: 2021.07.13 16:33:09 +02'00'

Fdo.: Atilano Fernández-Pacheco Sánchez-Migallón Fecha: 10/07/2021

GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

RECUENTO DE VOTOS ELECTRÓNICO UTILIZANDO LA RED DE ETHEREUM, UNA APLICACIÓN BLOCKCHAIN

Autor: Daniel Ortega Núñez

Director: Atilano Fernández-Pacheco Sánchez-Migallón

Madrid

Agradecimientos

En este apartado se supone que debemos agradecer a la gente que nos ha apoyado, y pudiendo escribir en primera persona, entrar en sentimentalismos. Pero por si lees esto y no me conoces, soy una persona de poco escribir, así que voy a hacerlo rápido y sin muchos rodeos.

En primer lugar, y punto más importante, quería agradecer a mi padre y a mi madre por todo el apoyo, sobre todo emocional, y por la cantidad incontable de ánimos pese a "mis caras de estudio" en las comidas, a mi primo tanto por toda la ayuda técnica como por ayudarme a gestionar algún agobio repentino, a mi director del TFG por acogerme en este proyecto en el que no encontraba a nadie, y por último a todos mis compañeros, y en especial, a los de este último curso que ha sido cuarto de carrera, que la mayoría, acabando juntos en este momento la carrera, hemos estado siempre acompañándonos y ayudándonos unos a otros.

RECUENTO ELECTRONICO DE VOTOS UTILIZANDO LA RED DE ETHEREUM, UNA APLICACIÓN BLOCKCHAIN

Autor: Ortega Núñez, Daniel.
Director: Fernández-Pacheco Sánchez-Migallón, Atilano.
Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

Los grandes cambios sociales causador por la digitalización han demostrado que simplifican innumerables procesos, y con ello, facilitan la vida de muchas personas. Este proyecto propone una digitalización de unas elecciones electorales, aportando un modelo, que utiliza una tecnología en auge, la tecnología Blockchain.

Palabras clave: Blockchain, Votaciones, Smart Contract.

1. Introducción

La tecnología Blockchain, es una tecnología que está revolucionando sectores como el económico, siendo sustento para ello la conocida criptomoneda Bitcoin. El principal atractivo de esta tecnología es la posibilidad que ofrece a los usuarios de realizar intercambios sin necesidad de intermediarios y protegiendo su privacidad.

Pero estas no son las únicas características de esta tecnología, y por tanto no es únicamente aplicable a temas relacionados con la economía. Blockchain puede ser de gran utilidad en numerosos ámbitos, como para crear una nube de almacenamiento descentralizada, o generar registros de datos distribuidos por toda la red.

Aplicando esta tecnología se intentarán aprovechar todas las ventajas que pueda ofrecer la misma para solventar los fallos y mejorar los sistemas actuales de recuento de votos, y a su vez comprobar si tiene alguna característica que impida una justa y correcta votación

2. Definición del proyecto

Este proyecto consiste en investigar sobre la posible aplicación de la tecnología Blockchain en un proceso de recuento de votos de unas elecciones de forma digital, y la implementación de ello en la red, con lo que se podrá comprobar tanto su viabilidad, como su seguridad en cuanto a posibles ataques a la red o intentos de modificar los votos, ya que uno de sus puntos fuertes es la seguridad que ofrece esta tecnología a nivel de almacenamiento de datos en la red, además del anonimato de la votación. En este caso se utilizará la red de Ethereum[1] para desarrollar el proyecto.

3. Descripción del sistema

Para la implementación de este sistema, se diseñará una aplicación que interactuará de forma simultánea con la red Blockchain y con una base de datos de MySQL12[2]. La encargada de toda la lógica y las funcionalidades, así como del almacenamiento del recuento de votos será la Blockchain, la base de datos únicamente se utilizará para facilitar la interacción de los usuarios con la aplicación.

Dentro de la red Blockchain se implementan dos Smart Contract, uno que aporta la lógica interna de unas votaciones, y otro que aporta la generación de un token ERC-20[3]. Este último contrato se ha utilizado para la representación de los votos mediante un objeto digital, es decir, cada token generado en ese contrato representa un voto. Este token es administrado desde el primer contrato comentado, que despliega las lógicas de las votaciones, desde el que se realizan las transacciones de estos tokens llamando a las funciones de ERC-20. Se utilizará la librería de web3 para interactuar con la blockchain desde una aplicación web.

Por otro lado, MySQL se utiliza para crear tablas sencillas, para la implementación de la aplicación web, otorgando una interactividad más visual y accesible al usuario.

Con lo que el sistema a desarrollar en este proyecto está compuesto por numerosas herramientas, pero la arquitectura a rasgos generales que muestra la estructura del proyecto se muestra a continuación en la *Figura 1*:

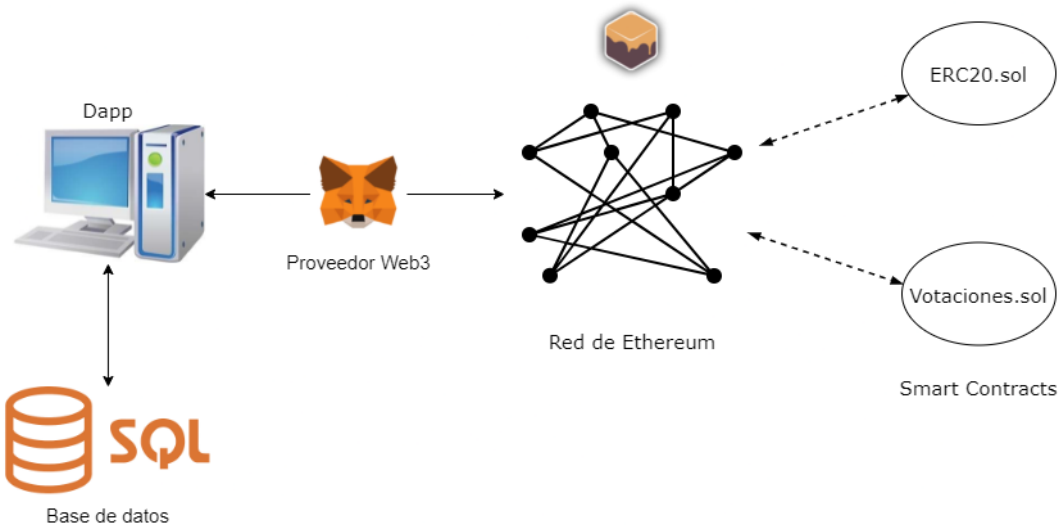

Figura 1: Diagrama de la arquitectura del sistema.

El proyecto se divide principalmente en tres partes:

- La más característica del proyecto, la red de Ethereum. En esta parte es donde se encuentra definida toda la lógica de las votaciones y del token utilizado para representar los votos, contenidas dentro de los Smart Contracts.
- La base de datos, que como se ha comentado contendrá únicamente tablas para mejorar la interfaz de la página web.
- La página web, que constará de numerosos archivos escritos en PHP que generaran una web con diferentes pestañas con diferentes funcionalidades, así como una interfaz de inicio de sesión controlada a través de la base de datos.

4. Resultados

Una vez desplegada y configurada la aplicación se realiza una prueba a pequeña escala de unas votaciones, en este caso para 3 votantes y dos partidos, en la que únicamente votan dos de los tres usuarios antes de cerrar las votaciones al mismo partido.

Mediante estas pruebas se han podido comprobar el correcto funcionamiento de los limitantes impuestos a ciertas acciones para un correcto funcionamiento lógico de estas votaciones, y además se ha podido comprobar la información en las transacciones que corresponderían con la votación de un usuario. Esta información se puede observar a continuación en la *Figura 2* gracias a Ganache[4].

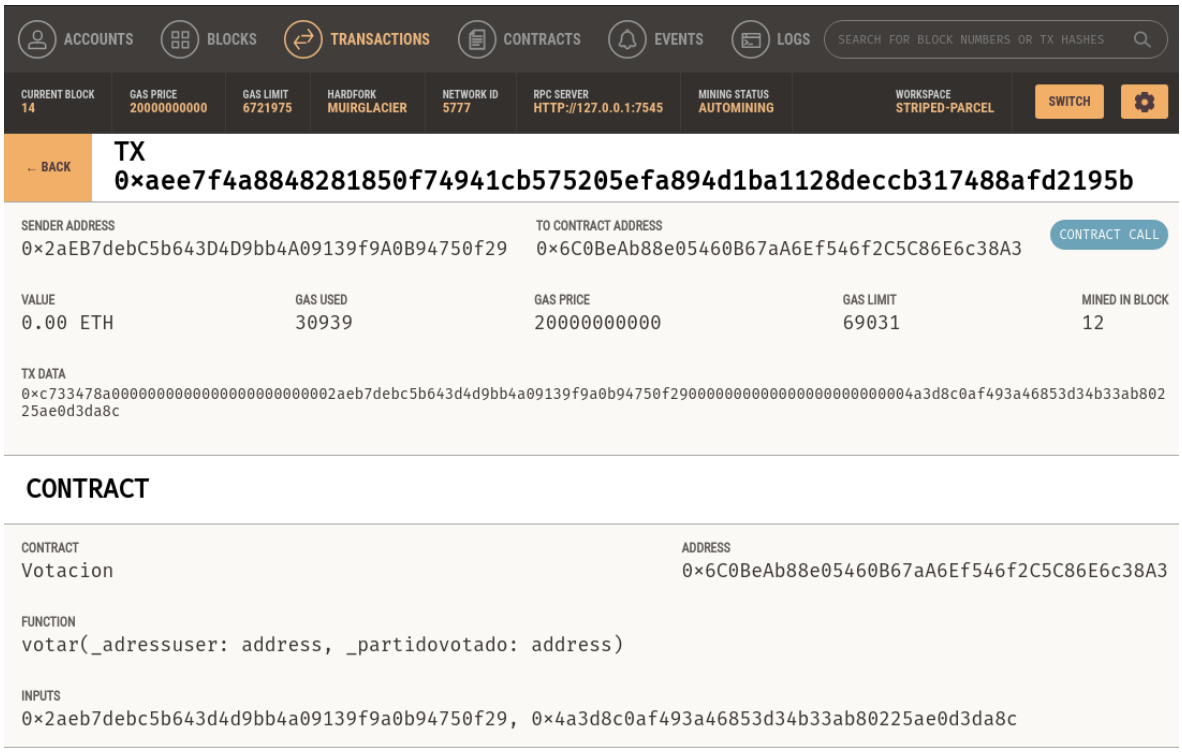

Figura 2: Información de la transacción.

En la imagen se puede observar la gran transparencia en el proceso de votación a un partido, y el proceso de almacenamiento de los votos en un mapeo de variables.

Gracias a las pruebas realizadas se ha comprobado el correcto funcionamiento de la lógica dentro de un recuento de votos en unas elecciones, durante todo el proceso, desde el reparto del token representante del voto, pasando por la elección del usuario al partido y su correspondiente voto, hasta la función de llamada de recuento de tokens almacenados por cada partido.

Como punto importante no conseguido, se observa que, en el sistema diseñado, existe un requerimiento relacionado con las características que debería tener una votación, que no está presente. Este requerimiento se trata de la existencia de un anonimato en las votaciones, ya que, aunque a priori pueda parecer así, las características de trazabilidad de la Blockchain hacen posible ver entre que cuentas se ha realizado una transferencia como se puede ver en la *Figura 2*. Y en el diseño realizado, al necesitar vincular una cuenta a cada usuario para realizar el reparto de tokens, la cuenta perteneciente a cada usuario es conocida, y trazando las transferencias de esa cuenta se podría saber a qué partido ha votado cada usuario.

5. Conclusiones

Tras la finalización del proyecto, se puede concluir el correcto funcionamiento lógico del sistema mediante esta tecnología. Pero la necesidad de una posible solución al diseño realizado, o un nuevo diseño para la aplicación de esta lógica. Esta necesidad es debida a la falta de cumplimiento de todos los derechos necesarios en unas elecciones, en este caso el del anonimato. Y con este rediseño, poder evitar que, mediante la característica trazabilidad de la Blockchain, sea posible obtener quien ha realizado cada voto, o no identificar a cada usuario con una cuenta, pero sin con ello perder las características de seguridad que podría ofrecer este sistema por su naturaleza distribuida.

6. Referencias

[1] "Ethereum", ethereum.org, 2021. [Online]. Available: https://ethereum.org/en/. [Accessed: 05- Jul- 2021].

[2] "MySQL", Mysql.com, 2021. [Online]. Available: https://www.mysql.com/. [Accessed: 05- Jul- 2021].

[3] "¿Qué es un token ERC-20? | Bit2Me Academy", *Bit2Me Academy*, 2021. [Online]. Available: https://academy.bit2me.com/que-es-erc-20-token/. [Accessed: 05- Jul- 2021].

[4] "Ganache | Truffle Suite", *Truffle Suite*, 2021. [Online]. Available: https://www.trufflesuite.com/ganache. [Accessed: 05- Jul- 2021].

ELECTRONIC VOTE COUNTING USING ETHEREUM NET, A BLOCKCHAIN APPLICATION

Author: Ortega Núñez, Daniel.
Supervisor: Fernández-Pacheco Sánchez-Migallón, Atilano.
Collaborating Entity: ICAI – Universidad Pontificia Comillas

ABSTRACT

The social changes brought about by digitalization have proven to simplify countless processes, thus making life easier for many people. This project proposes a digitization of an electoral election, providing a model that uses a booming technology, Blockchain technology.

Keywords: Blockchain, Voting, Smart Contract.

1. Introduction

Blockchain technology is a technology that is revolutionizing sectors such as the economic sector, being the basis for it the well-known cryptocurrency Bitcoin. The main attraction of this technology is the possibility it offers users to make exchanges without the need for intermediaries and protecting their privacy. But these are not the only characteristics of this technology, and therefore it is not only applicable to issues related to the economy. Blockchain can be very useful in many areas, such as creating a decentralized storage cloud, or generating data records distributed throughout the network. Applying this technology will try to take advantage of all the benefits it can offer to solve the failures and improve the current systems of vote counting, and in turn check if it has any feature that prevents a fair and correct voting.

2. Project definition

This project consists of investigating the possible application of Blockchain technology in a digital vote counting process of an election, and the implementation of it in the network, with which it will be possible to check both its viability and its security in terms of possible attacks on the network or attempts to modify the votes, since one of its strengths is the security offered by this technology at the level of data storage in the network, in addition to the anonymity of the vote. In this case, the Ethereum[1] network will be used to develop the project.

3. Description of the system

For the implementation of this system, an application will be designed that will interact simultaneously with the Blockchain network and with a MySQL[2] database. The Blockchain will be in charge of all the logic and functionalities, as well as the storage of the vote count, the database will only be used to facilitate the interaction of the users with the application.

Two Smart Contracts are implemented within the Blockchain network, one that provides the internal logic of voting, and another that provides the generation of an ERC-20 token[3]. The latter contract has been used for the representation of votes by means of a digital object, which means that each token generated in this contract represents a vote. This token is managed from the first commented contract, which deploys the voting logics, from which the transactions of these tokens are performed by calling the ERC-20 functions. The web3 library will be used to interact with the blockchain from a web application.

On the other hand, MySQL is used to create simple tables, for the implementation of the web application, giving a more visual and accessible interactivity to the user.

The system to be developed in this project is composed of numerous tools, but the general architecture that shows the structure of the project is shown below in *Figure 1*:

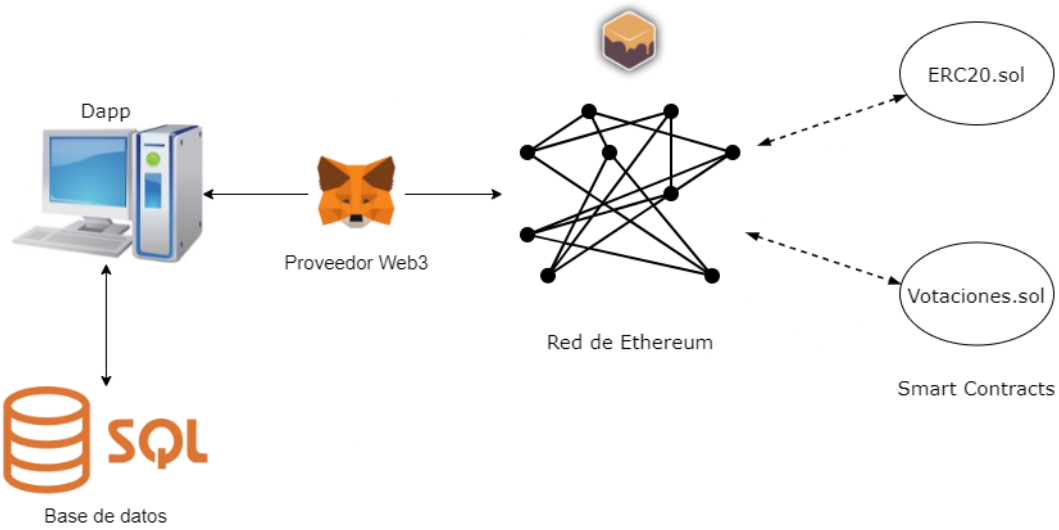

Figure 1: System architecture diagram.

The project is mainly divided into three parts:

- The most characteristic of the project, the Ethereum network. This part is where all the voting logic and the token used to represent the votes, contained within the Smart Contracts, are defined.
- The database, which as mentioned above will contain only tables to improve the interface of the website.
- The web page, which will consist of numerous files written in PHP that will generate a web page with different tabs with different functionalities, as well as a login interface controlled through the database.

4. Resultados

Once the application has been deployed and configured, a small-scale test of some votes is carried out, in this case for 3 voters and two parties, in which only two of the three users vote before closing the votes to the same party.

By means of these tests it has been possible to verify the correct functioning of the limitations imposed on certain actions for a correct logical functioning of these votes, and it has also been possible to verify the information in the transactions that would correspond to the vote of a user. This information can be seen in *Figure 2* below thanks to Ganache[4].

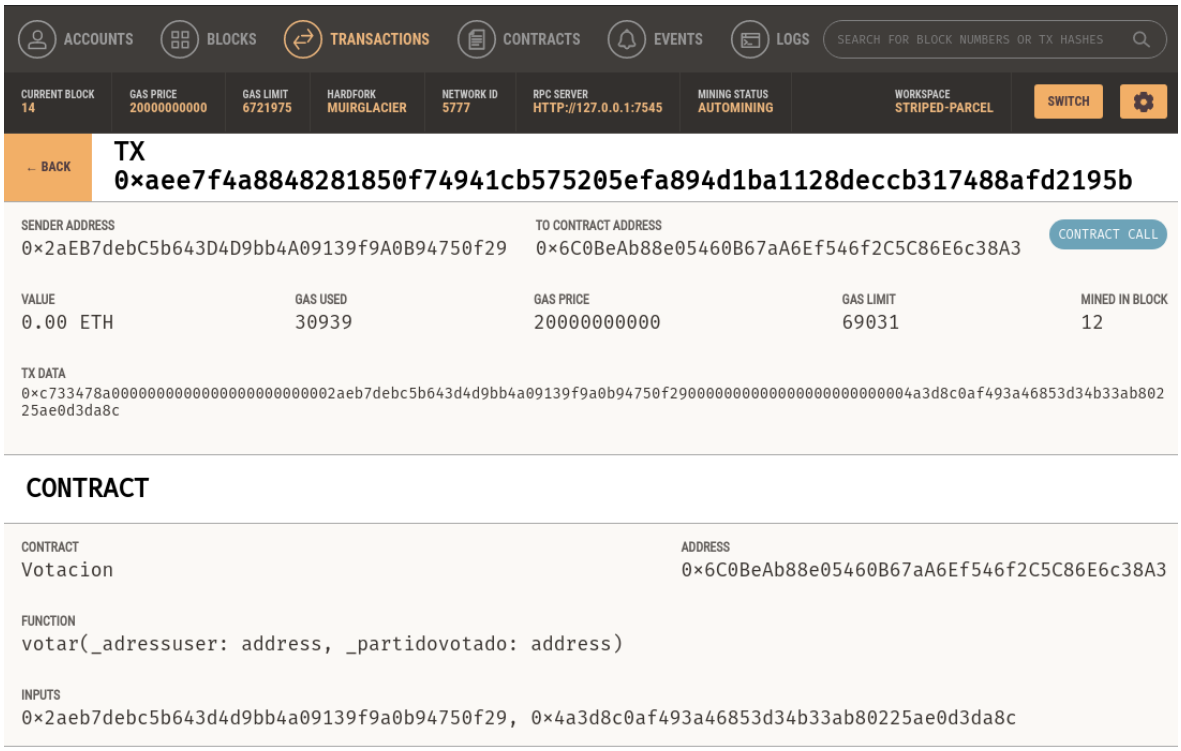

Figure 2: Transaction information.

In the image it is possible to observe the great transparency in the process of voting for a party, and the process of storing the votes in a mapping of variables.

Thanks to the tests carried out, the correct functioning of the logic within a vote count in an election has been verified, during the whole process, from the distribution of the token representing the vote, through the user's election to the party and its corresponding vote, to the function of calling the counting of tokens stored by each party.

As an important point not achieved, it is observed that, in the designed system, there is a requirement related to the characteristics that a vote should have, which is not present. This requirement is about the existence of anonymity in voting, since, although a priori it may seem so, the traceability features of the Blockchain make it possible to see between which accounts a transfer has been made as can be seen in *Figure 2*. And in the design made, by needing to link an account to each user to perform the distribution of tokens, the account belonging to each user is known, and by tracing the transfers from that account it would be possible to know which party each user has voted for.

5. Conclusions

After the completion of the project, it can be concluded the correct logical operation of the system using this technology. But the need for a possible solution to the design made, or a new design for the application of this logic. This need is due to the lack of compliance with all the necessary rights in an election, in this case that of anonymity. And with this redesign, to be able to avoid that, through the characteristic traceability of the Blockchain, it is possible to obtain who has made each vote, or not to identify each user with an account, but without thereby losing the security features that this system could offer because of its distributed nature.

6. Referencias

[1] "Ethereum", ethereum.org, 2021. [Online]. Available: https://ethereum.org/en/. [Accessed: 05- Jul- 2021].

[2] "MySQL", Mysql.com, 2021. [Online]. Available: https://www.mysql.com/. [Accessed: 05- Jul- 2021].

[3] "¿Qué es un token ERC-20? | Bit2Me Academy", *Bit2Me Academy*, 2021. [Online]. Available: https://academy.bit2me.com/que-es-erc-20-token/. [Accessed: 05- Jul- 2021].

[4] "Ganache | Truffle Suite", *Truffle Suite*, 2021. [Online]. Available: https://www.trufflesuite.com/ganache. [Accessed: 05- Jul- 2021].

Índice de la memoria

Índice de Ilustraciones

Índice de tablas

Capítulo 1. INTRODUCCIÓN

Este proyecto consiste en investigar sobre la posible aplicación de la tecnología Blockchain en un proceso de recuento de votos de unas elecciones de forma digital, y la implementación de ello en la red, con lo que se podrá comprobar tanto su viabilidad como su seguridad en cuanto a posibles ataques a la red o intentos de modificar los votos, además del anonimato de la votación.

En cuanto a la tecnología Blockchain se refiere, es una tecnología que está revolucionando sectores como el económico, siendo sustento para ello la conocida criptomoneda Bitcoin. El principal atractivo de esta tecnología es la posibilidad que ofrece a los usuarios de realizar intercambios sin necesidad de intermediarios y protegiendo su privacidad.

Estas características no hacen que esta tecnología sea únicamente aplicable a temas relacionados con la economía. Blockchain puede ser de gran utilidad en numerosos ámbitos, tanto como para crear una nube de almacenamiento descentralizada, como para gestionar identidades o generar registros de datos distribuidos por toda la red.

Aplicando esta tecnología se intentarán aprovechar todas las ventajas que pueda ofrecer la misma para solventar los fallos y mejorar los sistemas actuales de recuento de votos, y a su vez comprobar si tiene alguna característica que impida una justa y correcta votación

1.1 MOTIVACIÓN DEL PROYECTO

Hace unos años, sucedió un boom sin precedentes de las criptomonedas que llamó mucho la atención por todo internet y culminó en el primer pico de Bitcoin, e hizo que mucha gente se interesase por el tema. Cuando realmente se investiga sobre ello, su funcionamiento a nivel tecnológico llama por su complejidad, y entendiéndolo, la cantidad de aplicaciones que tiene la tecnología Blockchain se hace visible.

Entre esta cantidad de aplicaciones, existen muchas que pueden tener un gran impacto en la sociedad actual, ayudando a mejorarla y creando mayor comodidad para las personas.

Este proyecto, desde un punto de vista social podría facilitar la votación de mucha gente y aumentar su confianza en el sistema de recuento de votos, previniendo un pensamiento negativo de si se está cometiendo fraude.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

Este proyecto se realizará sobre una blockchain ya existente y conocida como es Ethereum, siendo la blockchain, una tecnología basada en una cadena de bloques, la cual se explicará brevemente a continuación, junto con todas las herramientas utilizadas y de interés para este proyecto.

2.1 BLOCKCHAIN

Blockchain es una tecnología en la que se van registrando las transacciones que se realizan en una red de forma segura, las cuales quedan públicas, es decir, libremente accesible por cualquier usuario, en una base de datos distribuida. Esta base de datos solo puede ser actualizada con el consenso de la mayoría de los participantes de la red, por tanto, es un registro verificable de las operaciones realizadas desde el nacimiento de la red.

Su propio nombre indica su funcionamiento, se trata de una cadena de bloques, en la que la información y transacciones se va añadiendo en un bloque, que se vincula con el anterior. Dicho funcionamiento se puede ver resumido y esquematizado en la *Ilustración 1* donde se puede apreciar que cada bloque tiene, entre otras cosas: la información de sus transacciones, un hash para el bloque, y el hash del bloque anterior, lo que determina el orden de la cadena. Estos hash son creados mediante un algoritmo que transforma cualquier bloque de datos en una serie de caracteres con longitud fija.

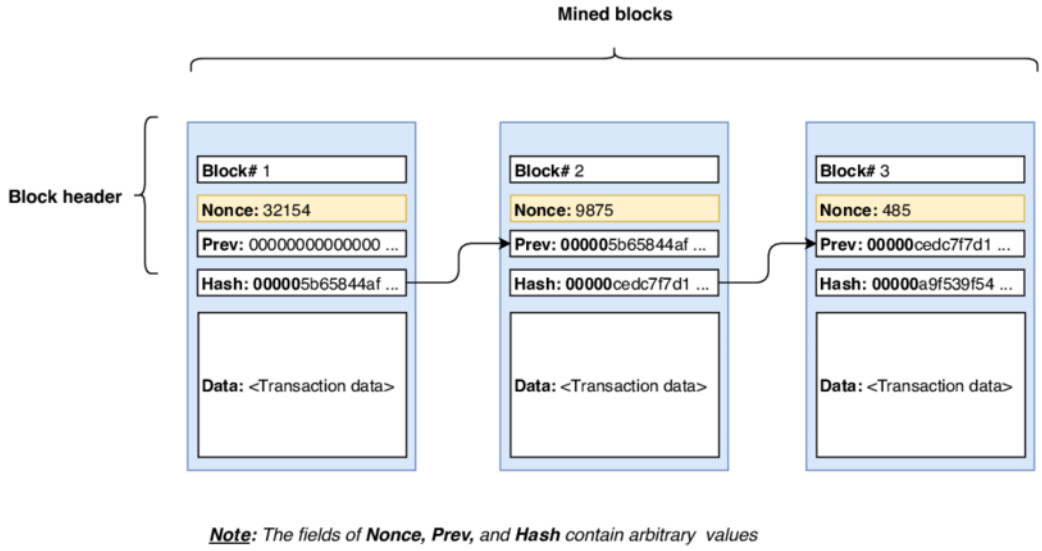

Ilustración 1: Esquema de los bloques y su información.[1]

De esta serie de caracteres llamada Hash, es de la que depende la dificultad de minado de los bloques, ya que tiene que minarse un determinado número de bloques por tiempo, y por lo que la red ajusta la dificultad de minado dependiendo del número de ceros requerido en el hash, para ajustarse a ese tiempo de bloque.

Pero toda la información almacenada en los bloques tiene que almacenarse, y para ello están los nodos de la red, que son los encargados de almacenar y distribuir una copia de cada bloque nuevo, de forma que esa información no se pierda, aunque se desconecte uno de los nodos, es decir, toda la información queda almacenada en los nodos, hasta que se añade un bloque, que queda registrado en la red para siempre.

El principal propósito de esta tecnología es lograr una red descentralizada formada por múltiples nodos distribuidos, de forma que los datos almacenados en ellos no sean modificados de ninguna forma y no estén controlados por ningún núcleo o entidad, y con ello ofrecer una fiabilidad a los usuarios.

Para entender correctamente estos conceptos, a continuación, en la *Ilustración 2*, podemos ver el funcionamiento de cómo se actualizaría una cadena de bloques al solicitar una transacción.

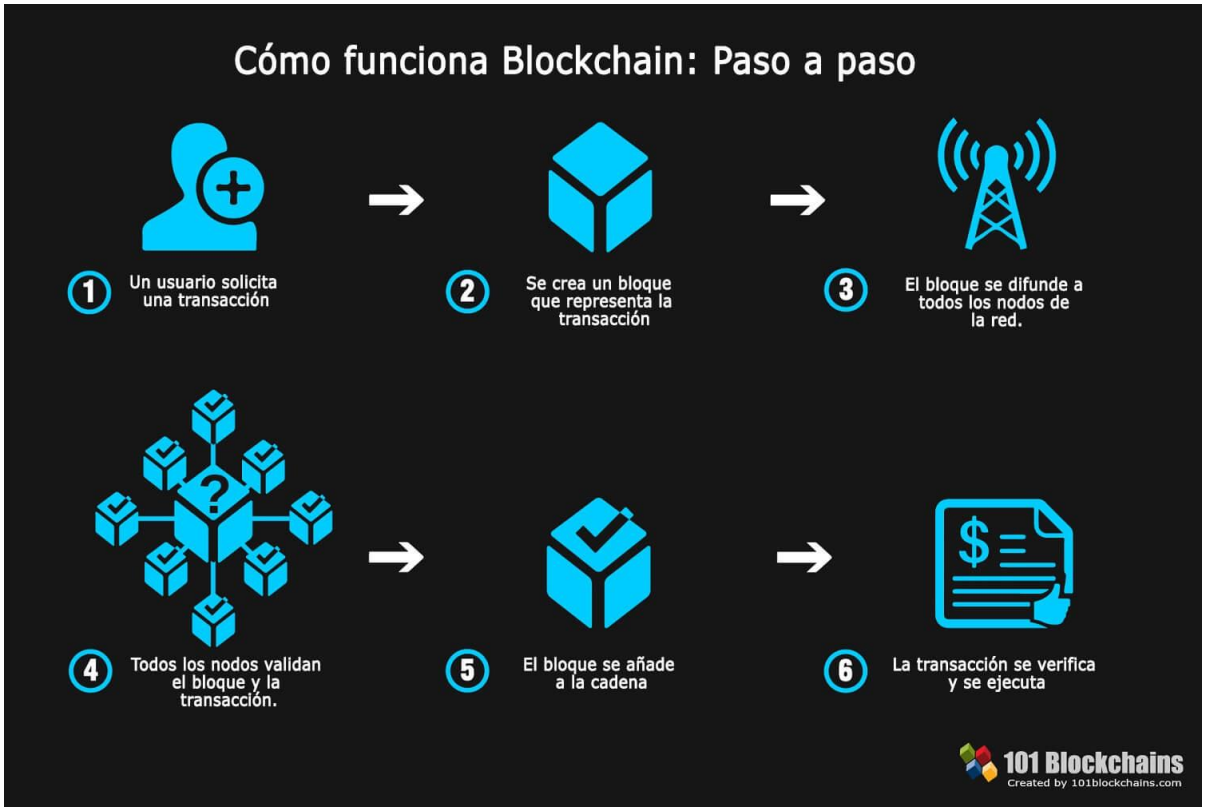

Ilustración 2: Funcionamiento Blockchain.[3]

Esta transacción generaría un bloque, que se distribuiría a todos los nodos, los cuales al validarlo por mayoría se añade a la cadena, la cual queda actualizada e inalterable.

2.1.1 BLOCKCHAIN ETHEREUM

La red blockchain que se usará en este proyecto será Ethereum. Ethereum es una plataforma global de código abierto para aplicaciones descentralizadas, una tecnología creada por la comunidad tras la criptomoneda ether (ETH), la cual es actualmente la segunda más grande del mercado de criptomonedas, y por múltiples aplicaciones[2]. El propósito inicial de esta red cuando fue creada en 2015 por el programados Vitalik Buterin, fue el de crear un instrumento para aplicaciones descentralizadas o colaborativas, las cuales se conocen por el nombre de DApps.

2.2 *SMART CONTRACTS*

La utilidad extra que aporta esta red entre otras cosas, son los smart contract o contratos inteligentes, lo que aporta mucha facilidad a la hora de programar ciertas lógicas que den instrucciones a la red. Es decir, un smart contract es un programa que realiza ciertas lógicas que se le hayan indicado dependiendo de diferentes condiciones, como por ejemplo acuerdos entre personas. Estos programas se pueden ejecutar debido a que los nodos disponen de una máquina virtual conocida como EMV (Ethereum virtual machine).

Estos contratos se programan en su propio lenguaje de programación, Solidity[4], y corren sobre la EMV que es un entorno de ejecución de los mismos. Este código de programación es un código de alto nivel orientado a objetos, muy útil para controlar la lógica y el comportamiento de las cuentas de Ethereum dentro de un smart contract.

2.3 *DAPPS*

Las DApps[5] son aplicaciones descentralizadas, que utilizan esta tecnología para la comunicación directa entre usuarios sin ningún intermediario y cuyo back-end está ejecutándose en una red descentralizada y no en un servidor centralizado. En el caso de Ethereum, cualquier usuario puede crear su propia DApp, e implementarla en la red Blockchain. Estas aplicaciones tienen infinidad de funcionalidades, ya que pueden, desde interactuar con las criptomonedas, hasta almacenar complejas lógicas de programas. Estas aplicaciones se aprovechan de las características de la Blockchain como en el caso de la seguridad de sus datos.

2.4 *PRINCIPALES HERRAMIENTAS UTILIZADAS*

Para este proyecto han sido necesarias ciertas herramientas y programas que ayudan y facilitan la realización del mismo:

2.4.1 NODEJS

NodeJS[6] es un entorno de tiempo de JavaScript que dispone de lo necesario para ejecutar un programa en dicho lenguaje, y que nos permite ejecutar en el servidor una arquitectura orientada a eventos. La finalidad de NodeJS es la creación de aplicaciones de red rápidas gracias a su alta escalabilidad.

2.4.2 TRUFFLE

Truffle[7] es el entorno de trabajo o framework más utilizado por los desarrolladores para Ethereum, ya que proporciona un conjunto de herramientas para ayudar al desarrollo de proyectos más fácilmente. Provee con una cartera de activos utilizando la EVM y ofrece funciones como: compilación, despliegue o incluso escribir los propios smart contract y hacerles test. También proporciona una serie de plantillas para estos proyectos las cuales son fácilmente editables para que tome forma y posteriormente sea probado.

Lo que hace a este framework una herramienta ideal es que permite pasar de la idea conceptual de smart contract, a un desarrollo del producto final[8].

2.4.3 GANACHE

Ganache[9] es una herramienta que pertenece al conjunto de aplicaciones de Truffle y que proporciona una red Blockchain personal para hacer simulaciones de prueba sin costes por tasas de la red. Está disponible tanto como aplicación de escritorio como una herramienta de comandos, en este proyecto se utilizará la primera.

Desde esta herramienta se pueden realizar tanto transacciones de prueba, como el despliegue de contratos y como se ha comentado anteriormente sin que esto suponga costes reales, gracias a su disposición de diferentes cuentas en esta Blockchain personal.

2.4.4 METAMASK

Metamask[10] es una cartera de criptomonedas, que es instalada como extensión de un navegador, y que permite a estos navegadores interactuar con la red Blockchain de Ethereum.

Permite a los usuarios las transacciones de criptomonedas y tokens basados en Ethereum y conectarse a aplicaciones descentralizadas a través del navegador web en el que esté instalada la extensión.

2.4.5 REMIX IDE

La manera más sencilla, y la utilizada en este proyecto para programar en Solidity, lenguaje de los Smart Contract, es usando Remix IDE[11], un entorno integrado de desarrollo basado en un navegador que aporta un compilador y un entorno en tiempo de ejecución.

2.4.6 VS CODE

Para facilitar la creación y modificación de archivos, es necesario un editor de texto, como editor de texto para este proyecto se ha utilizado principalmente VS Code[12].

Se decidió utilizar este editor de texto por su gran compatibilidad con los diferentes lenguajes utilizados además de por su compatibilidad para diferentes plataformas, tanto Windows como Linux.

2.4.7 LAMP

Se trata de una infraestructura de aplicaciones de software de código abierto, que continente ciertas herramientas para el sistema operativo de Linux. Estas herramientas son Principalmente: Apache que se trata del servidor web, MySQL como gestor de la base de datos, y PHP el lenguaje de programación utilizado.

Estas aplicaciones instaladas juntas, permiten que un servidor pueda alojar aplicaciones y sitios web dinámicos escritos en PHP.

2.4.7.1 phpMyAdmin

PhpMyAdmin[13] es una herramienta utilizada para facilitar la administración de MySQL utilizando el navegador web. De esta forma la interacción con la base de datos se facilita.

Capítulo 3. ESTADO DE LA CUESTIÓN

Actualmente, muchos procesos han sido transformados completamente debido a la constante digitalización que estamos viviendo, lo cual ha creado numerosas ventajas tanto sociales como económicas aumentando el bienestar de las personas. Pero muchos otros siguen realizándose mediante métodos tradicionales, quizás porque aún no se confía plenamente en la digitalización de estos.

Es el caso de las votaciones que se realizan en la mayoría de los países, en las que aún se sigue utilizando principalmente un método convencional de votación por medio de papeletas. Realizar estas de forma digital facilitaría en gran medida el recuento de los votos y agilizaría el proceso. Para ello deberían asegurarse dos aspectos fundamentales a la hora de diseñar el sistema: en primer lugar, que no sea posible la falsificación de una identidad y por tanto, que no se introdujesen en la base de datos más votos de los que debería haber, además de intentar proporcionar la máxima seguridad posible frente a ciberataques; y en segundo lugar, garantizar el anonimato total del votante, es decir, asegurar el sufragio secreto de cada individuo y no permitir que ningún voto sea relacionado con una persona sin su consentimiento.

El blockchain, donde más impacto tiene, y por tanto su aplicación principal, es para medios digitales de intercambio de valor, que se relacionan con las conocidas criptomonedas, puesto que el uso de Bitcoin o Ethereum como métodos de pago alternativos es una realidad. Pero como se ha comentado anteriormente, no tiene únicamente esta aplicación.

Esto es debido a que su funcionamiento de, como su nombre indica, ir colocando bloques de información uno detrás de otro, se puede aplicar a cualquier tipo de registro, de forma que en cada bloque se almacena una cantidad de información validada por la red y una relación del bloque con el anterior haciendo que cada bloque tenga su lugar específico en la cadena. Además, esta cadena de bloques se distribuye en varios nodos de la red, por lo que se garantiza la disponibilidad de la información en todo momento.

La idea de utilizar esta tecnología para crear un sistema de recuento de votos es aprovecharse de su característica como una base de datos más segura que las habituales. Esto es debido a que esta tecnología al crear los registros entrelazados y cifrados, protege la seguridad y la privacidad de esta información que además, al guardarlos en todos los nodos de la red, sería muy complicado que la información se perdiese o fuese atacada por ciberdelincuentes, ya que aunque un nodo se desconecte de la red, el resto de nodos seguirán disponiendo de la información, o aunque un nodo fuese atacado y modificada su información, el resto de nodos verificarían la veracidad de dicha información. Esto hace que para poder cambiar parte de la información almacenada se debería atacar a más del 50% de la red, de otra forma sería imposible cambiar la información de cualquiera de los bloques de la cadena. Esto solucionaría la posible falta de confianza de muchas personas que no confían en otros métodos de recuento de votos digitales, por miedo a hackeos y posibles fraudes electorales, pudiendo aportar hasta sus ordenadores personales como parte de la red, funcionando como nodos.

El hacking en el ámbito político es perfectamente posible, como se ha podido ver por ejemplo en elecciones de EEUU, donde un grupo de hackers bloquearon el acceso a ciertas bases de datos que verificaban las firmas de los votantes, y modificaron o eliminaron el registro de ciertos votantes atacando algunos sistemas electorales que se utilizaban por parte de los trabajadores de las mesas electorales. Esto deja a la vista lo importante que es actualizar los planes de ciberseguridad de las elecciones.

Otra de las características y una gran aplicación de esta tecnología, es la gestión de identidades, ya que los usuarios podrían crear su propia identidad digital a prueba de manipulación. Con estas identidades se podría realizar hasta firmas de documentos digitales, pero la idea en este proyecto es utilizarlas para registrarte como votante y que nadie pueda suplantar una identidad.

Por último, habría que garantizar el completo anonimato del voto. En el caso de transacciones de Bitcoin, por ejemplo, se realizan de forma anónima, el sistema únicamente conoce que desde una cartera se quiere enviar una cierta cantidad de la criptomoneda a otra cartera, pero no se conocen los datos de la persona. En el caso de este

proyecto habrá que ver como verificar la identidad de cada persona para que no se pueda enviar más de un voto por usuario y a su vez el voto de este sea totalmente anónimo.

Si se consigue lograr esto, se podrán evitar problemas como en las pasadas elecciones de EEUU (elecciones 2020) de un presunto fraude electoral, por la fiabilidad y seguridad del sistema, permitiendo a la gente votar de forma electrónica ayudando también a impedir la propagación del Covid-19.

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 JUSTIFICACIÓN

El desarrollo de un sistema de recuento de votos digital y fiable resultaría en una mejora considerable del sistema, ya que actualmente las votaciones se realizan de forma convencional o por correo, pero no existe ningún sistema digital en el que la sociedad confíe.

En este proyecto se va a implementar un sistema de recuento de votos digital, a través de tecnología blockchain por las características que puede aportar de fiabilidad, ya que las votaciones actuales, demandarán un sistema digital en un futuro. La realización de las votaciones de forma digital facilitaría mucho el proceso, evitando a la gente tener que desplazarse de sus casas para votar, ayudando con ello, por ejemplo, a muchas personas de elevada edad que les cueste desplazarse, o debido a la época que vivimos actualmente de Covid, evitar multitudes. Además, está implementación reduciría muchos costes de papel en todos los millones de sobres y en las papeletas que se imprimen para votar con ello, un cambio que producirá reducción de costes y un aumento en la comodidad delos votantes.

Con este proyecto se propone estudiar una opción a esta necesidad que se demandará de un sistema de recuento de votos de forma digital y fiable.

4.2 OBJETIVOS

- Crear un sistema de recuento de votos, para que existan alternativas a los métodos tradicionales que puedan sustituir, o funcionar de forma simultánea como un sistema complementario.
- Agilizar el proceso actual y crear una dificultad añadida para evitar que puedan suceder fraudes electorales.

- Comprobar la viabilidad de esta tecnología aplicada al recuento de votos, viendo, además, si esta respeta todos los derechos que ofrece el sistema tradicional y la seguridad a los votantes.

4.3 METODOLOGÍA

Para desarrollar el proyecto se definen las siguientes tereas:

1. Análisis/Búsqueda de información: se estudiarán otros proyectos que empleen tecnología blockchain para entender el funcionamiento en mayor profundidad a nivel técnico y buscar cómo sería una aplicación a este proyecto.
2. Diseño: se diseñará el sistema a nivel lógico.
3. Desarrollo: creación del proyecto.
4. Prueba/Simulación del sistema: el proyecto se simulará en una red de pruebas para comprobar su correcto funcionamiento.
5. Revisión de errores: ver los fallos de la simulación y comprobar los posibles fallos.
6. Desarrollo de la memoria: redacción del documento correspondiente a la memoria del proyecto.

4.4 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA

Una vez analizadas todas las tareas necesarias, es necesaria una planificación y un reparto de tareas, para ello se muestra a continuación una tabla con un cronograma, en la que se indican las tareas a realizar en cada ocasión.

Tareas	Febrero	Marzo	Abril	Mayo	Junio
Análisis/Búsqueda de información					
Diseño					
Desarrollo					
Prueba/Simulación del sistema					
Revisión de errores					
Desarrollo memoria					

Tabla 1: Cronograma.

Además, para este proyecto, se ha realizado una estimación económica de los costes que han supuesto este proyecto:

- Equipo para el desarrollo: para el desarrollo de este sistema se ha utilizado como equipo el ordenador de uso personal dentro de una máquina virtual, por lo que no ha habido gastos materiales en este ámbito.

Ordenador de sobremesa personalizado	
Procesador	Intel Core i7-4790 3.6 GHz
Sistema operativo (Máquina virtual)	Windows 10 Pro (Ubuntu)
Tarjeta gráfica	GeForce GTX 970 4GB
Memoria RAM	24 GB DDR3-1600MHz
Discos duros	2x 120 GB SSD / 1TB HDD

Tabla 2: Especificaciones del equipo utilizado.

- Coste de software: para este proyecto ha sido utilizadas múltiples herramientas, pero todas ellas gratuitas de código abierto y tampoco han supuesto costes

adicionales. Las herramientas utilizadas se muestran en *Principales herramientas utilizadas*.

- Costes de despliegue: toda la implementación y las pruebas realizadas se hicieron de forma local en una máquina virtual, simulando una Blockchain, por lo que no se dieron costes adicionales relacionados con la implementación.

Capítulo 5. SISTEMA/MODELO DESARROLLADO

En este capítulo se va a desarrollar de forma más extensa el sistema diseñado para este proyecto, en el que se pretende utilizar la tecnología Blockchain para facilitar la lógica de un sistema de votaciones digitalizado y comprobar si es un sistema fiable que permita el anonimato en unas votaciones.

Para ello se realizará en primer lugar un análisis del sistema actual en el que se expondrán los diferentes tipos de elecciones en España y su funcionamiento, para a continuación, con estas pautas, crear un diseño semejante utilizando la tecnología Blockchain y los Smart Contracts, y finalizar mostrando su implementación.

5.1 ANÁLISIS DEL SISTEMA

Para realizar un diseño digital de unas votaciones hace falta conocer el funcionamiento del sistema de votaciones actual y con ello cuáles serán los requisitos de nuestro diseño:

5.1.1 SISTEMA ELECTORAL TRADICIONAL

En el caso de España existen cuatro tipos de elecciones: las elecciones generales que deciden quien formara parte del Congreso de los Diputados y del Senado, las elecciones europeas en las que se elige a los europarlamentarios españoles, las elecciones autonómicas en las que los ciudadanos de cada comunidad autónoma deciden quien formará parte del parlamento autonómico de su respectiva comunidad, por último, las elecciones municipales que deciden los miembros de los ayuntamientos[16].

En las elecciones generales, existen ciertas características que afectaran al diseño de nuestros sistema. Para este proyecto lo que nos interesa es su funcionamiento, y para ello habría que tratar de analizar el sistema D'Hondt, además de su división por circunscripción[17].

El número de circunscripciones depende de si la votación es al senado o al congreso de los diputados. En el caso de la elección de los miembros del congreso, las circunscripciones corresponden a cada una de las provincias además de las ciudades autónomas, a cada una de estas circunscripciones le corresponde un número diferente de diputados. A causa de estas variables, el sistema se diseñará para aplicarse y recoger las votaciones de una circunscripción, es decir, se deberá implementar el sistema para cada circunscripción por separado, ya que este sistema no recogerá la división de circunscripciones.

Para repartir los escaños en cada circunscripción se utiliza el sistema D'Hondt, es un método matemático para dividir estos escaños de forma aproximadamente proporcional a su porcentaje en votos. Esta aproximación es necesaria debido a que no es posible asignar un número decimal de escaños. Este método matemático no se tendrá en cuenta para el desarrollo de este proyecto dentro de la blockchain, sino que será un proceso que deberá realizarse posteriormente, una vez obtenida la cantidad exacta de votos obtenidos por cada partido.

Por tanto, como se ha comentado anteriormente, para el desarrollo de este proyecto se ha decidido enfocar el recuento de votos únicamente a una circunscripción, y no añadir la lógica del reparto de escaños dentro de los Smart Contracts, tanto por las características del token a utilizar para el recuento de votos, como por la posible implementación de este sistema como adicional al sistema actual de recuento de votos, por lo que habría que realizar el reparto de escaños una vez sumados todos los votos de los diferentes sistemas. Se podría tratar de explicar esta aplicación poniendo como ejemplo que este sistema se podría utilizar para el recuento de votos de la Comunidad de Madrid ya que es una circunscripción, y el reparto de escaños se realizaría posteriormente con los votos obtenidos del sistema.

5.1.2 TOKENS

Dentro de las Blockchain, pueden existir tokens, que se generan a partir de lógicas de programación en Smart Contracts, y es esta lógica la que describe cómo funciona el token, por lo que existen diferentes tipos de token. Los tokens pueden representar cualquier cosa,

desde litros de agua, hasta watios de electricidad, por lo que se planteó para este proyecto desarrollar un token que representase el voto de los usuarios, y por tanto la gestión de estos votos será a través de transferencia de tokens.

Existen ya multitud de códigos de token creados con una gran protección de fallos de seguridad en su lógica, en el caso de este proyecto se ha analizado la estructura ya preestablecida de un token ERC-20, pero por simplificar la estructura, se decidió crear uno propio con una lógica más sencilla, aun a costa de posibles fallos de seguridad, para la implementación de un sistema de prueba.

5.1.3 REQUERIMIENTOS

Para cumplir con las cualidades de un sistema de votación, sin que las votaciones o el votante se vean afectados en ningún ámbito, deberán cumplirse ciertos requisitos a la hora de diseñar el sistema:

- Todos los usuarios deben tener la posibilidad de votar utilizando este sistema
- Un usuario no debe poder enviar más de un voto, por lo que es necesario un registro con un documento de identificación (DNI) previo a la votación, o alguna base de datos que identifique y relacione a cada ciudadano con una cuenta de blockchain.
- La Junta Electoral Central debe poder incluir todos los partidos que se presenten a las elecciones.
- El sistema debe comenzar la votación, y no permitir más votos tras el cierre de la misma, en sus momentos correspondientes.
- Las cuentas de los partidos únicamente pueden recibir votos, no enviar ni votar.
- Los votos de los usuarios deben ser anónimos.

5.2 DISEÑO

Junto con el análisis realizado y los requerimientos obtenidos en el apartado anterior, se ha diseñado el sistema por partes, por un lado, el diseño de la Blockchain, es decir, la lógica

de los Smart contract, y por otro lado una web que interactúa con la Blockchain apoyada en una base de datos de SQL. A continuación, se mostraran estos diseños de forma esquematizada para facilitar su comprensión y como resultaría la arquitectura completa del proyecto, que para su previa visualización se muestra a continuación en la *Ilustración 3*:

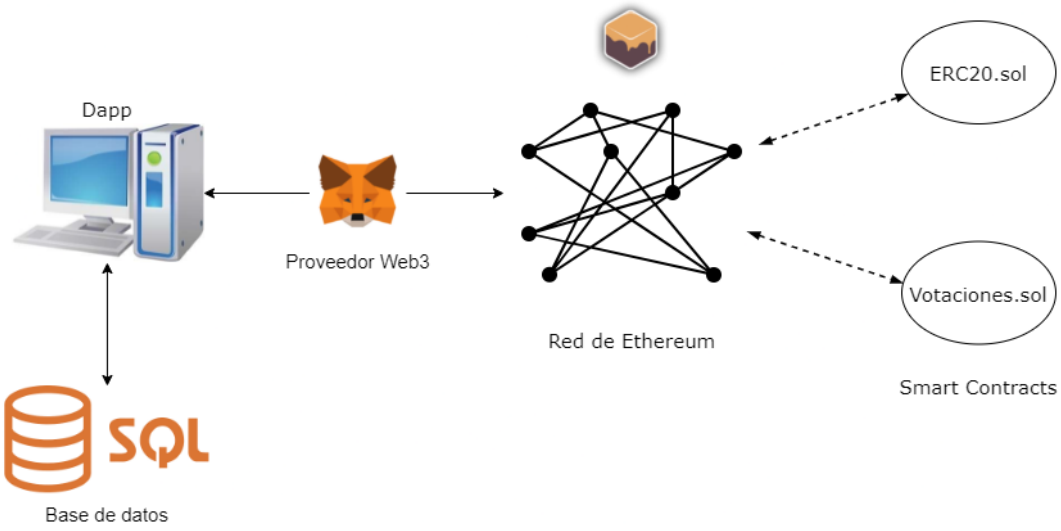

Ilustración 3: Arquitectura completa del proyecto.

5.2.1 DISEÑO DE SMART CONTRACTS

Para realizar la lógica de los Smart Contracts, se utilizarán dos contratos:

El primero se trata de un token ERC-20, que es un contrato con un diseño ya preestablecido, y que aporta ciertas funcionalidades. Para este contrato, se ha realizado un estudio a partir del contrato ERC-20 aportado por OpenZeppelin[14] como se ha comentado en el apartado *Tokens*, de forma que se ha editado para simplificarlo, y darle las características necesarias para representar las funcionalidades de nuestro sistema, a este contrato se le ha denominado "ERC20.sol". Teniendo en cuenta el propósito de representar un voto con un token, se ha modificado el contrato para que disponga de una variable de entrada a la hora de realizar su despliegue en la Blockchain, de forma que el contrato se inicia conociendo en número de votantes que van a realizar las votaciones a través de este

sistema, y genera esa misma cantidad de votantes en tokens. Este contrato engloba todas las funciones de transferencia de estos tokens.

El segundo contrato es el que engloba toda la lógica y las funciones de las votaciones, haciendo uso de los tokens ERC-20 comentados anteriormente, para ello al desplegarlo sobre la Blockchain debe tener como variable de entrada la dirección de memoria en la que se ha desplegado anteriormente el contrato de ERC-20, de esta forma el contrato de "Votaciones" puede hacer uso de las funciones de transferencia del primer contrato desplegado.

A continuación, en la *Ilustración 4*, se muestra un esquema de ambos contratos con sus respectivas variables de entrada, y su despliegue sobre la blockchain:

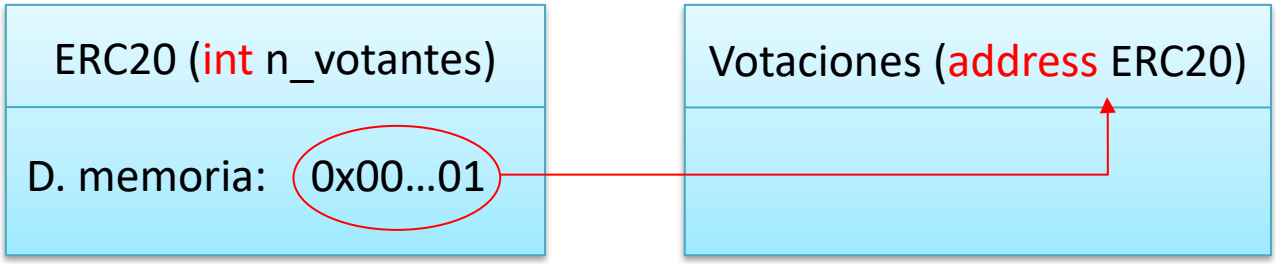

Ilustración 4: Esquema Smart Contracts.

Como se puede observar, es siempre necesario desplegar antes el Smart Contract del Token ERC-20, ya que el contrato "Votaciones" necesita una de sus variables para desplegarse.

Estos contratos deben desplegarse en la blockchain desde la cuenta de Blockchain de la JEC (Junta Electoral Central), la cual tendrá permisos para realizar acciones sobre estos contratos que los usuarios que no han desplegado el contrato no tendrán. Además de esto, tras desplegarlo, todos los tokens están en posesión de la JEC, y esta es la encargada de distribuir todos entre los usuarios antes de que se comience la votación. Por lo que, este contrato, como se comentará más detalladamente en *Casos de Uso*, diferencia usuarios de quien ha desplegado el contrato, en este caso la JEC.

5.2.2 BASE DE DATOS

Para la creación de un sistema más sencillo e intuitivo para el usuario, a este proyecto se le ha añadido una base de datos muy sencilla. Esta base de datos dispone de dos tablas:

La primera tabla se utilizará para realizar un registro de los votantes preparada con antelación, en la que estarán los usuarios identificados con su DNI, y tendrán su cuenta de Blockchain asignada, a las que se repartirán los tokens antes de comenzar la votación. Además, junto con una contraseña, podrán iniciar sesión en la página de votaciones a través de Blockchain que se comentará en el apartado de *Diseño de la Interfaz*. A continuación, en la *Ilustración 5* se muestra un ejemplo de la tabla utilizada:

+ Options

	id	username	dni	password	cuenta	admin
Edit Copy Delete	1	JEC	00000000A	cnt1	0xdCDD754cd10A6647573bf5f06Fc5dfb01CcaaAe3	1
Edit Copy Delete	2	Juan	00000001A	cnt2	0x3F4C121cA9B6eF1D61dE61a8640F4377f0c61041	0
Edit Copy Delete	3	Maria	00000002A	cnt3	0x4dECaF3Ea4EDD5dfE4f4f551367d7940f13E5fa7	0
Edit Copy Delete	4	Pablo	00000003A	cnt4	0x17281Cec883138Bf00fD0aE627F32d0a97d7e8CD	0

Check all *With selected:* Edit Copy Delete Export

Ilustración 5: Tabla de usuarios.

En resumen, esta tabla es utilizada para:

- Realizar un inicio de sesión tanto de los usuarios como del administrador(JEC) en la página web.
- Identificar a cada usuario con una cuenta, y con ello evitar que un usuario vote dos veces con cuentas distintas.
- Localizar las cuentas de los usuarios a las que hay que repartir los tokens.

La segunda tabla, se trata de una sencilla asociación del nombre de los partidos presentados con su correspondiente cuenta, a la que hay que enviar los votos. Esta tabla está pensada únicamente para facilitar el voto al usuario, evitando que tenga que conocer la cuenta del partido para realizar la votación, y pueda votar conociendo únicamente el

nombre del partido. Un ejemplo de esta tabla se muestra a continuación, en la *Ilustración 6*:

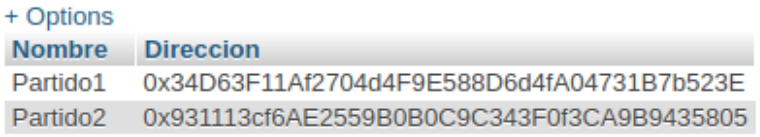

Ilustración 6: Tabla de Partidos.

Esta última tabla es rellenada por la JEC a través de la interfaz web que se trata en el apartado de *Diseño de la Interfaz*.

5.2.3 DISEÑO DE LA INTERFAZ

El diseño del front-end de esta aplicación se ha desarrollado teniendo en cuenta las diferentes necesidades de interacción con la Blockchain, de forma que se han dividido en diferentes paginas para facilitar la comprensión tanto del usuario como de la JEC, organizando las funcionalidades dependiendo del punto de desarrollo en el que se encuentren las votaciones, el flujo de las paginas para los usuarios sería el siguiente mostrado en la *Ilustración 7*.

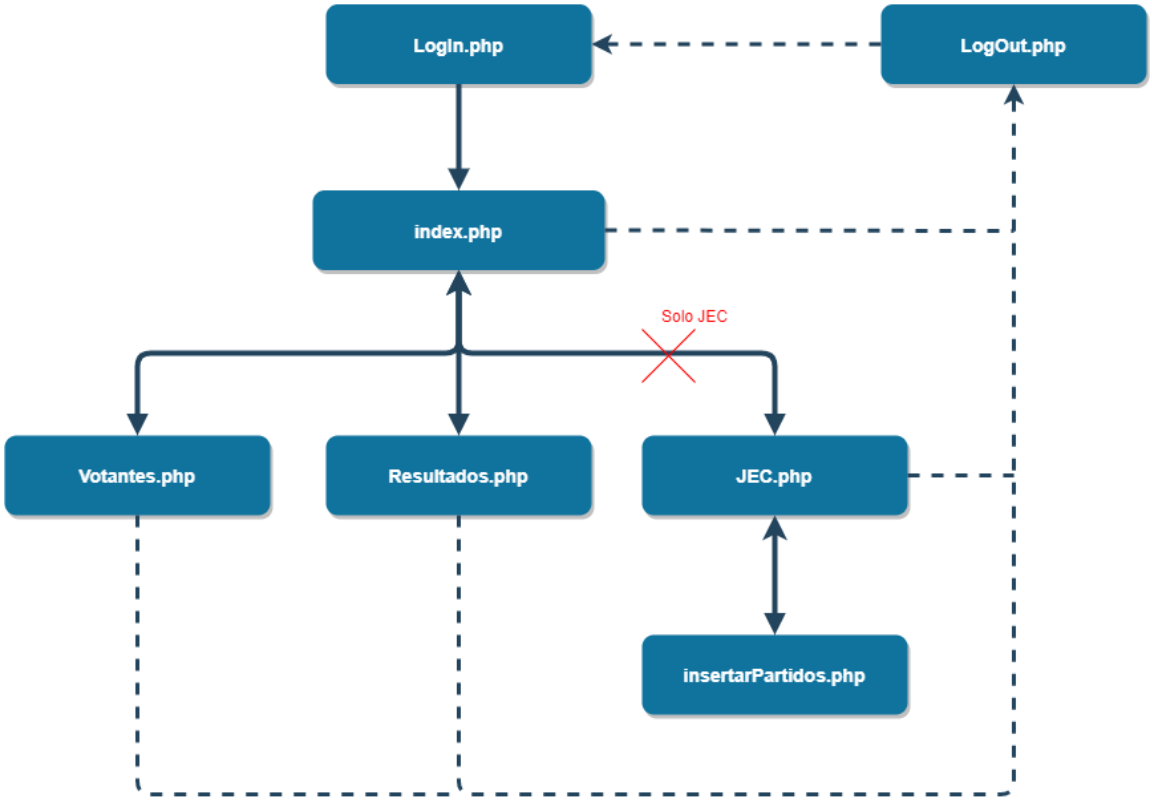

Ilustración 7: Diagrama de flujo de la web.

A continuación, se muestra la interfaz de la página principal de la aplicación en la *Ilustración 8*, lo que corresponde a "index.php".

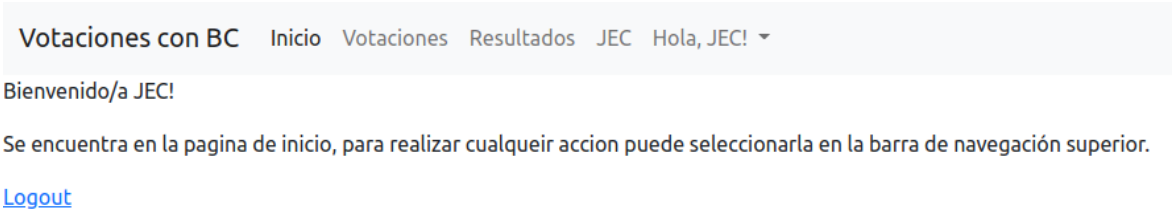

Ilustración 8: Interfaz "index.php".

Antes de comenzar estas votaciones, son necesarias interacciones con la Blockchain por parte de la JEC, tanto para organizar, como para añadir los votantes y los partidos. Estas funcionalidades se ofrecen dentro de "JEC.php"(*Ilustración 9*).

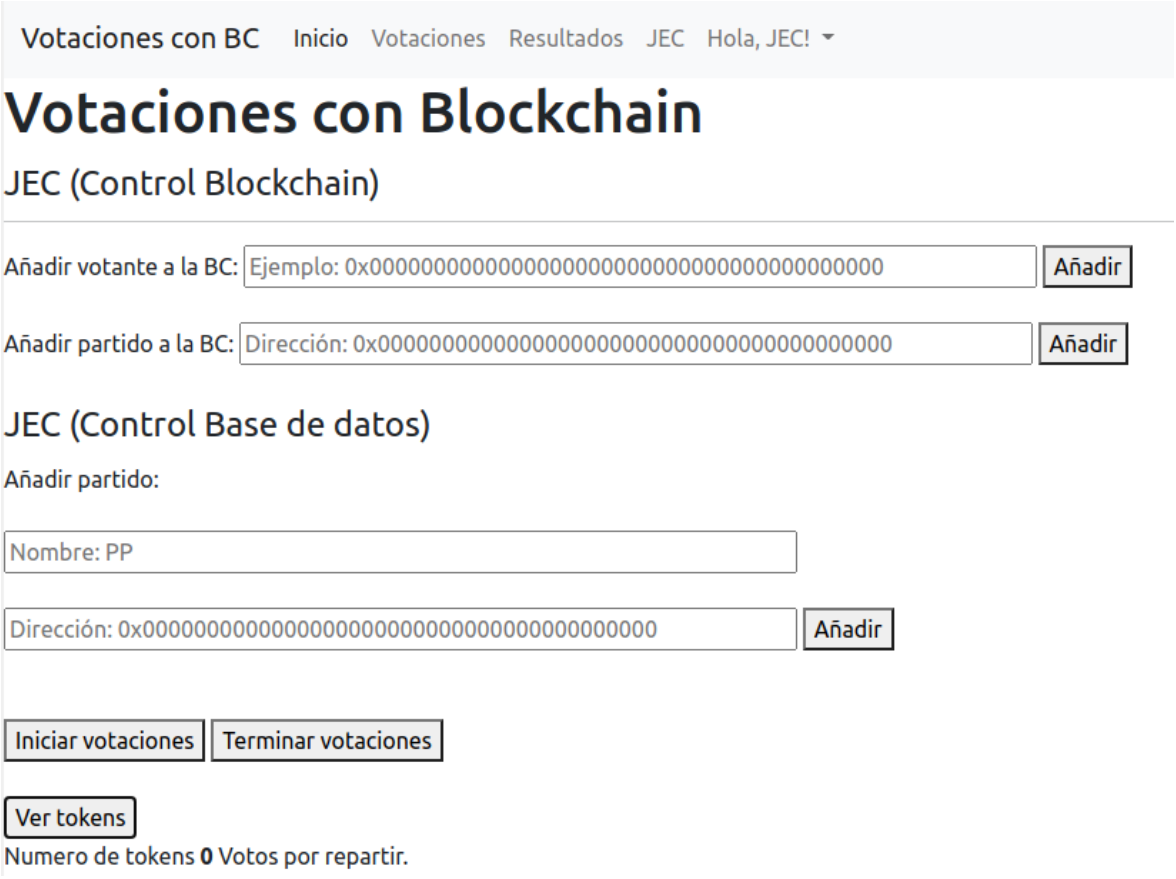

Ilustración 9: Interfaz "JEC.php".

Además, a esta página, únicamente puede acceder la cuenta registrada como administrador en la base de datos, como se ha mostrado en la *Ilustración 5*, para el resto de los usuarios no es visible como se puede comparar entre los diferentes menús mostrados a continuación en la *Ilustración 10* e *Ilustración 11*.

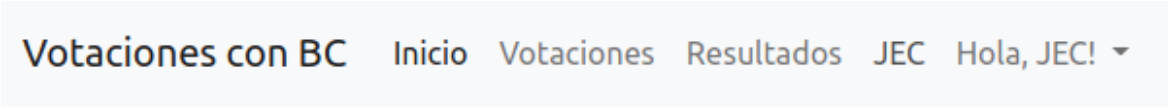

Ilustración 10: Barra de navegación de Admin.

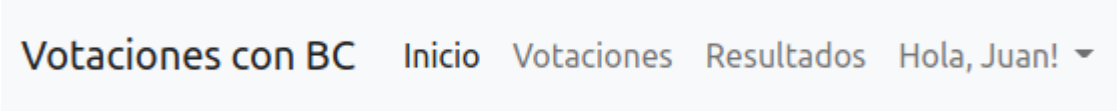

Ilustración 11: Barra de navegación de Usuario.

Cabe mencionar que si alguien no está autorizado a entrar en "JEC.php", aunque intente entrar de forma manual "localhost/JEC.php" será redirigido automáticamente a "index.php" sin ver el contenido de dicha página.

Una vez iniciadas las votaciones, los usuarios pueden emplear la página de "Votantes.php"(*Ilustración 12*) para realizar su elección de voto, y una vez realizada, revisar los resultados desde "Resultados.php"(*Ilustración 13*).

Ilustración 12: Interfaz "Votantes.php".

Ilustración 13: Interfaz "Resultados.php"

Por último, todas estas opciones no serían posibles sin haber realizado antes el inicio de sesión, cuya interfaz podemos observar en la *Ilustración 14*.

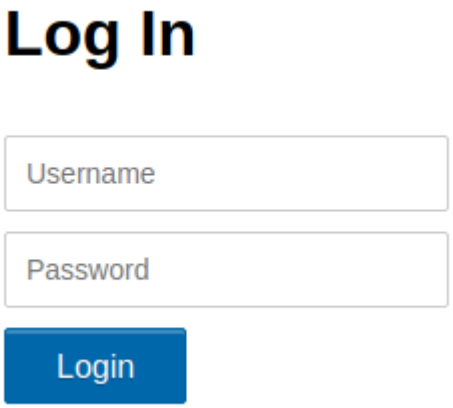

Ilustración 14: Interfaz del Log In.

5.2.4 CASOS DE USO

Para comprender mejor el comportamiento de este proyecto y sus funcionalidades, delimitadas por sus respectivos requerimientos, se muestran a continuación dos diagramas de casos de uso para mostrar de forma esquematizada las interacciones de los usuarios y el administrador.

Además de las acciones de iniciar sesión y cerrar sesión dentro de la página, que son acciones comunes y obligatorias para todos, a continuación, en la Ilustración 15: Diagrama de casos de uso (Votantes).*Ilustración 15* se muestran las acciones disponibles para los usuarios, es decir, los votantes:

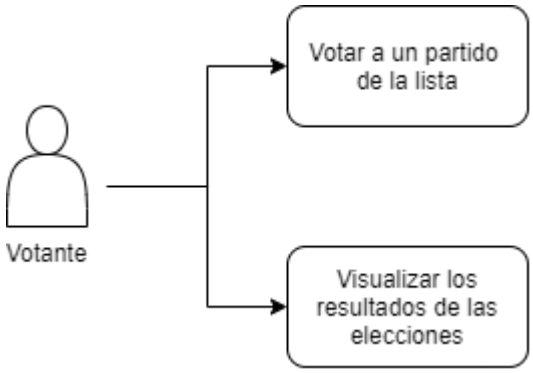

Ilustración 15: Diagrama de casos de uso (Votantes).

Como se puede observar, las interacciones del votante con la página son:

- Votar: Esta opción se representa en la página de "Votantes.php" y es la funcionalidad principal del proyecto, tiene una comunicación con la Blockchain, que realiza el envío del token en la cuenta del votante, a la cuenta del partido seleccionado.
- Visualizar los resultados: Esta opción se realiza desde "Resultados.php" y es la funcionalidad que permite al usuario comunicarse con la Blockchain y esta le devuelva y muestre el número de votos (tokens) del partido seleccionado.

Para el administrador de la página resultan muchas más funcionalidades, ayudando a la configuración e interacción de la lógica del Smart Contract. En la *Ilustración 16*, se puede observar estas funciones extra de administración del sistema.

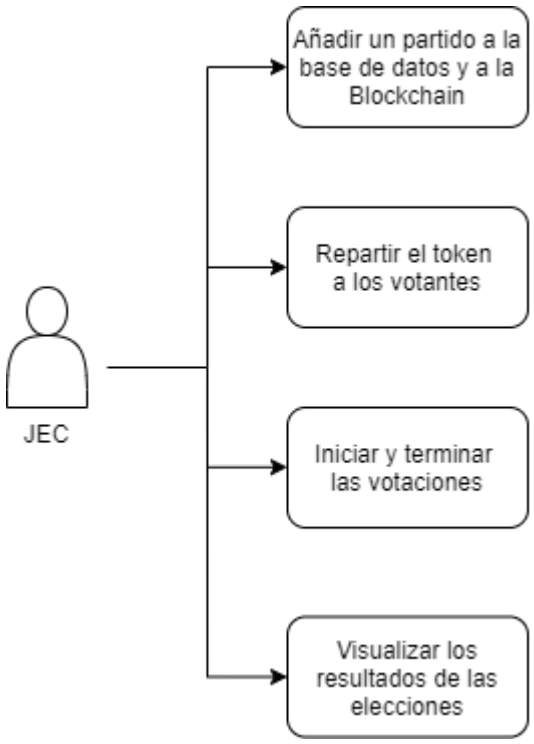

Ilustración 16: Diagrama de casos de uso (JEC/Admin).

Como se puede observar, los casos de uso para el administrador del sistema y sus interacciones se encuentran todas en la página de "JEC.php", a la cual únicamente puede acceder el administrador como se ha comentado en el apartado anterior *Diseño de la Interfaz*, y se encuentran como:

- Añadir partido: Esta opción permite añadir los partidos a la base de datos con su nombre y cuenta de Blockchain, además de declarar sus cuentas en la Blockchain como partidos.
- Añadir votante: Interacción que únicamente se comunica con la Blockchain para repartir el token al votante introducido.

- Iniciar y terminar las votaciones: Opción que permite a los votantes comenzar a votar, o todo lo contrario, no permitir la entrada de más votos por cierre de urnas. La opción de iniciar votaciones no se puede ejecutar si no se han repartido todos los votos, y el balance de la JEC es de 0 tokens.
- Visualizar los resultados: Al igual que el resto de los usuarios, los administradores también pueden visualizar los resultados de las votaciones. Esta es la única funcionalidad de los administradores que no se encuentra en "JEC.php", y como se ha comentado en los casos de uso del usuario, devuelve la cantidad de votos de un partido.

5.2.5 ARQUITECTURA Y ESTRUCTURA DEL PROYECTO

Para explicar el proyecto al completo, se analiza a partir de la *Ilustración 3* mostrada anteriormente, donde se observaba un diagrama con la arquitectura general del proyecto realizado:

Este proyecto se estructura en un directorio creado a partir del framework Truffle, que crea una carpeta de proyecto que engloba todos los archivos utilizados. Este directorio se estructura como se muestra en la *Ilustración 17*:

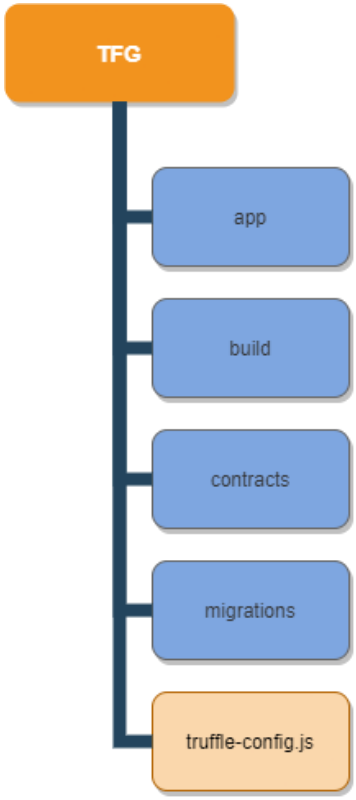

Ilustración 17: Estructura del proyecto.

Dentro de esta estructura encontramos varias carpetas que permiten una mayor organización del proyecto:

- "App" donde se encuentran todos los archivos utilizados para realizar el front-end de la aplicación, de los cuales se ha tratado anteriormente en *Diseño de la Interfaz*. Además, algunos archivos incluidos en esta carpeta también son los que permiten la comunicación con la base de datos de MySQL. A continuación, en la *Ilustración 18* se muestra un diagrama con todos los archivos incluidos en esta carpeta:

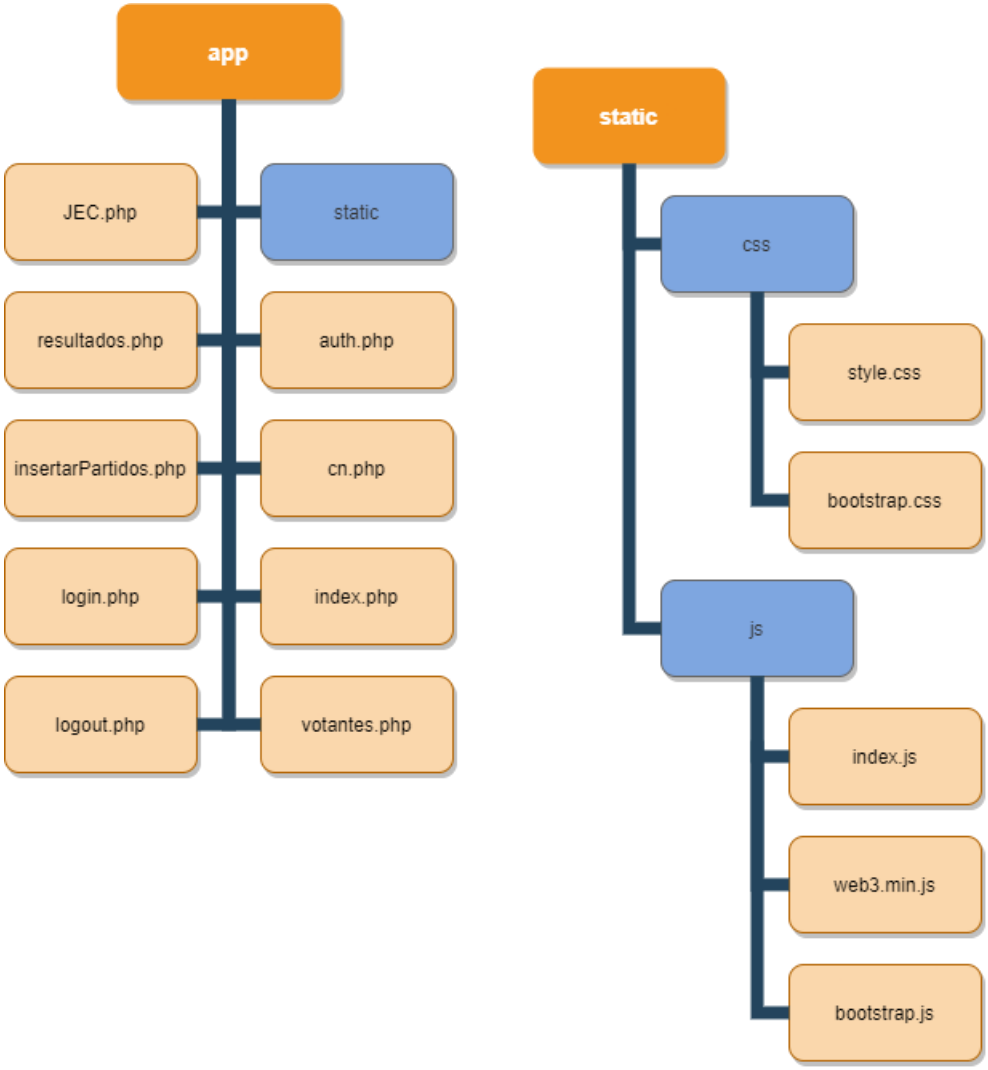

Ilustración 18: Diagrama de los archivos "app".

- "Build" es la carpeta que almacena los archivos de compilación de los Smart Contract. Son archivos JSON, que posteriormente se importan en los archivos de JavaScript de la web para transmitirle los datos. Para esta construcción únicamente se utiliza un archivo denominado "Votaciones.json" para obtener la ABI y la dirección en la red donde se ha desplegado el Smart Contract.
- "Contracts" es la carpeta que almacena los archivos de la lógica del back-end, es decir, el código de los Smart Contract escritos en Solidity.

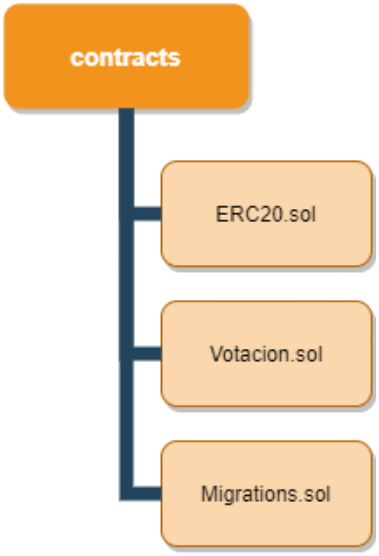

Ilustración 19: Diagrama de Smart Contracts.

- "Migrations" es una carpeta que contiene unos archivos, que como se explica en [15], están escritos en lenguaje JavaScript, y ayudan a desplegar los Smart Contract en la red de Ethereum, y que además son responsables de la organización y el orden de despliegue de estos contratos.

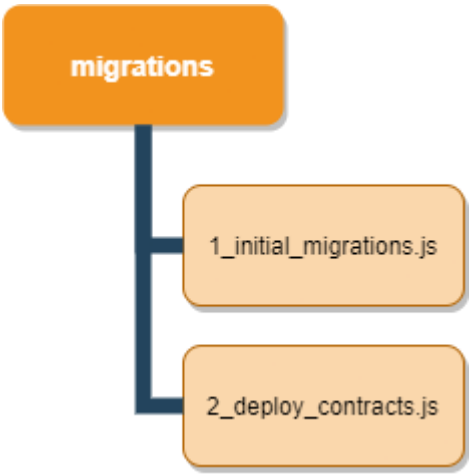

Ilustración 20: Diagrama "Migrations".

- Por último, como se muestra en la *Ilustración 17*, al proyecto lo acompaña un archivo "truffle-config.js", en el que se configura mediante su puerto la red de Blockchain que se va a utilizar además de otras opciones.

5.3 IMPLEMENTACIÓN

En este apartado se detallará de forma más completa como se ha llevado a cabo la implementación del sistema desde el comienzo del proyecto, hasta su finalización.

5.3.1 IMPLEMENTACIÓN SMART CONTRACTS

Para el desarrollo de este proyecto, fue necesaria la programación de Smart Contracts para implementar la lógica del sistema de votación sobre una red Blockchain, en este caso la red de Ethereum.

El lenguaje utilizado para programar estos Smart Contract es Solidity[4], y como se ha comentado anteriormente en el apartado *Principales herramientas utilizadas*, se utilizará Remix IDE, por la ayuda que proporcionan sus módulos de prueba y su facilidad para desplegar los contratos programados y probar sus funciones. A continuación, en la *Ilustración 21* se muestra un diagrama de clases que mostrará las funciones de cada contrato:

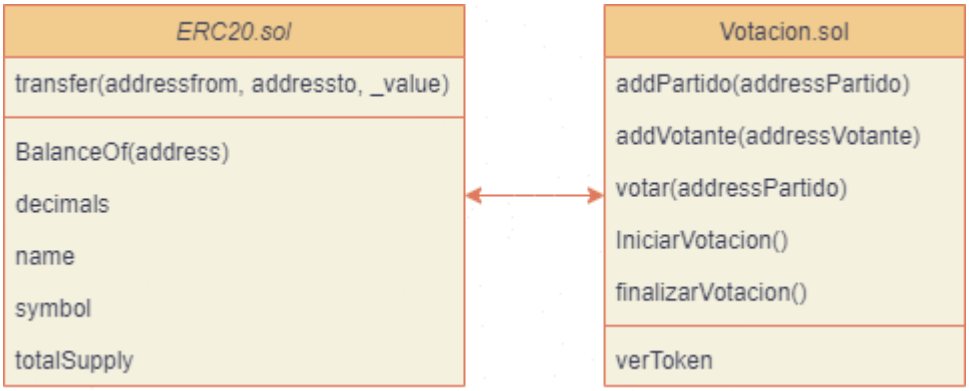

Ilustración 21: Diagrama de clases Smart Contracts.

Gracias a los módulos de prueba de Remix, se pudo comprobar el correcto funcionamiento de estas funciones en diferentes situaciones sin necesidad de desplegar los contratos en ninguna Blockchain.

5.3.2 DEPENDENCIAS

Para desplegar estos contratos en una blockchain son necesarias ciertas dependencias, de las cuales ya se ha hablado anteriormente en *Principales herramientas utilizadas*, por lo que, una vez desarrollados los contratos se instalará: NodeJS[6] que nos permitirá instalar ciertos paquetes necesarios en nuestro ordenador además de implementar la aplicación desarrollada, y Truffle[8] que es un framework que facilita mucho la implementación de proyectos de Blockchain, y a partir del cual se creará el proyecto con el comando: "truffle init" lo que facilitará un directorio como el mostrado y tratado en la *Ilustración 17*.

Una vez completo y configurado el proyecto el en directorio creado, y con sus respectivos archivos, es necesario simular una Blockchain, y para ello que se utilizará Ganache[9] una red Blockchain local que permite realizar pruebas sobre ella sin costes reales, y que acelera el proceso por no tener que esperar a la confirmación de nodos de una Blockchain real.

En la *Ilustración 22* se muestra la interfaz de Ganache, y es sobre donde se implementarán los contratos gracias a las migraciones de Truffle:

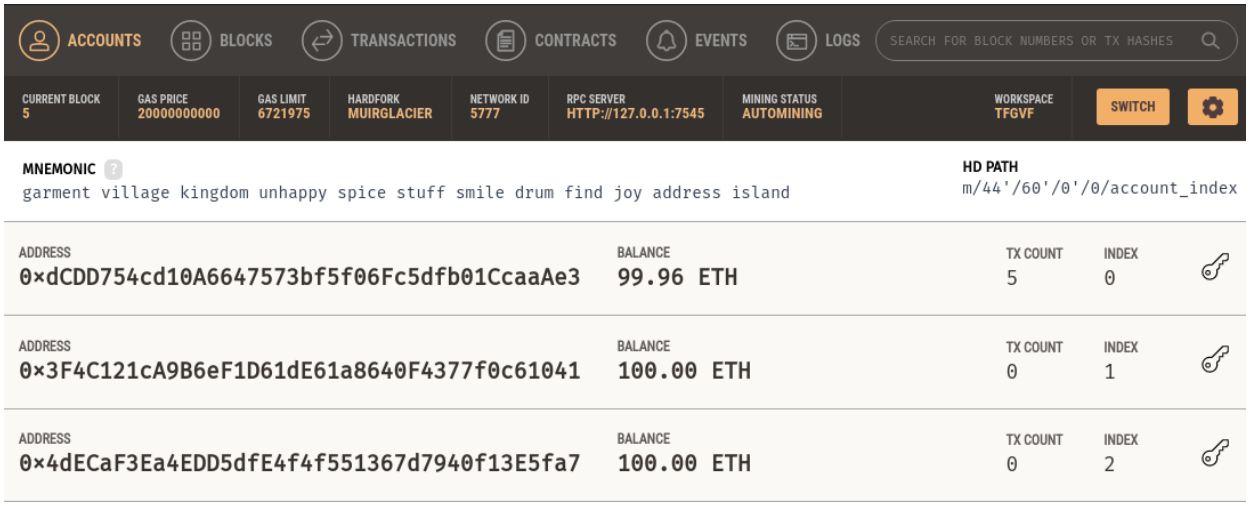

Ilustración 22: Interfaz Ganache.

Además, Ganache facilita un número de cuentas con ETH para que realicemos nuestras transacciones.

Para terminar la configuración, únicamente quedaría migrar los contratos y desplegarlos en la red de Ganache. En la *Ilustración 22* se puede observar como ya se han realizado estas implementaciones de los contratos sobre esta red, ya que en la cuenta 0 se han gastado ETH como tarifa de despliegue al migrar los contratos. El gasto exacto se puede observar al realizar esta operación con el comando "truffle migrate" como se muestra en la *Ilustración 23*.

```
2_deploy_contracts.js
=====================

   Replacing 'ERC20'
   -----------------
   > transaction hash:    0x91bd072e4f42bd374cb817b8b9ec40a024091746d82f1243f38b2b37550670f1
   > Blocks: 0            Seconds: 0
   > contract address:    0xDEaf418FD7341229f229af806667277d34b40313
   > block number:        3
   > block timestamp:     1625081933
   > account:             0xdCDD754cd10A6647573bf5f06Fc5dfb01CcaaAe3
   > balance:             99.98105168
   > gas used:            713135 (0xae1af)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.0142627 ETH


   Replacing 'Votacion'
   --------------------
   > transaction hash:    0x36c7ec47a803827f69ded0db204a5347383e1192dce5ae2e27471167ccc28b22
   > Blocks: 0            Seconds: 0
   > contract address:    0x590DBB169fa9187a73FaD0c9281c1442E7D67082
   > block number:        4
   > block timestamp:     1625081934
   > account:             0xdCDD754cd10A6647573bf5f06Fc5dfb01CcaaAe3
   > balance:             99.95911532
   > gas used:            1096818 (0x10bc72)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.02193636 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.03619906 ETH
```

Ilustración 23: Consumo de implementación de los contratos en Ganache.

5.3.3 APLICACIÓN WEB

Por último, la implementación de la interfaz que se comunicará con esta Blockchain se realizará con la ayuda de Apache, en este caso no será un sitio estático, y se utilizará PHP

junto con la base de datos, por lo que se ha optado por instalar *LAMP*(Apache, MySQL y PHP para Linux).

La implementación de esta web se realiza en el localhost de la máquina virtual, y la comunicación con la base de datos se realiza con phpMyAdmin(*Ilustración 24*) a través de la configuración mostrada en la *Ilustración 25*.

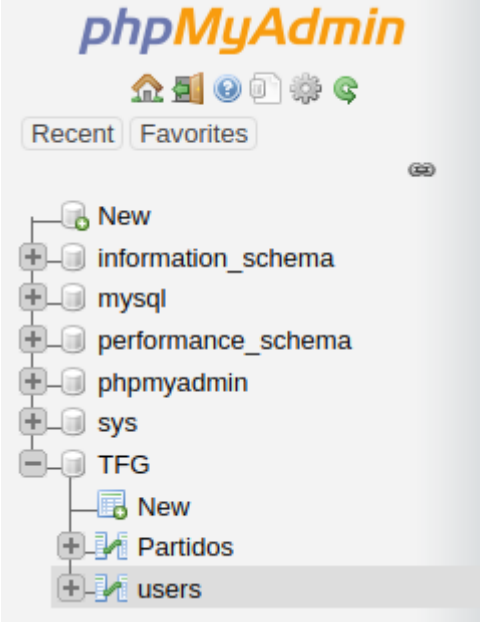

Ilustración 24: Servicio phpMyAdmin.

```php
<?php
    $con =mysqli_connect("localhost", "root", "1234", "TFG");
    mysqli_set_charset($con, "utf8");

    if (mysqli_connect_errno())
  {
  echo "Failed to connect to MySQL: " . mysqli_connect_error();
  }

?>
```

Ilustración 25: Configuración MySQL.

Capítulo 6. ANÁLISIS DE RESULTADOS

En este apartado se analizarán los resultados obtenidos durante la realización del proceso de votación con el sistema diseñado.

Para la realización de este proceso se ha implementado el proyecto simulando una votación sencilla, de 3 votantes y dos partidos. Para realizar esta implementación, hay que seguir los pasos indicados en el *Capítulo 5.* comenzando por desplegar los Smart Contract en la red de Ganache. Una vez desplegados se procede a abrir la aplicación en el navegador, configurando las cuentas (teniendo en cuenta que la red está configurada para Ganache) en Metamask como se muestra en la *Ilustración 26*: Configuración cunetas Metamask..

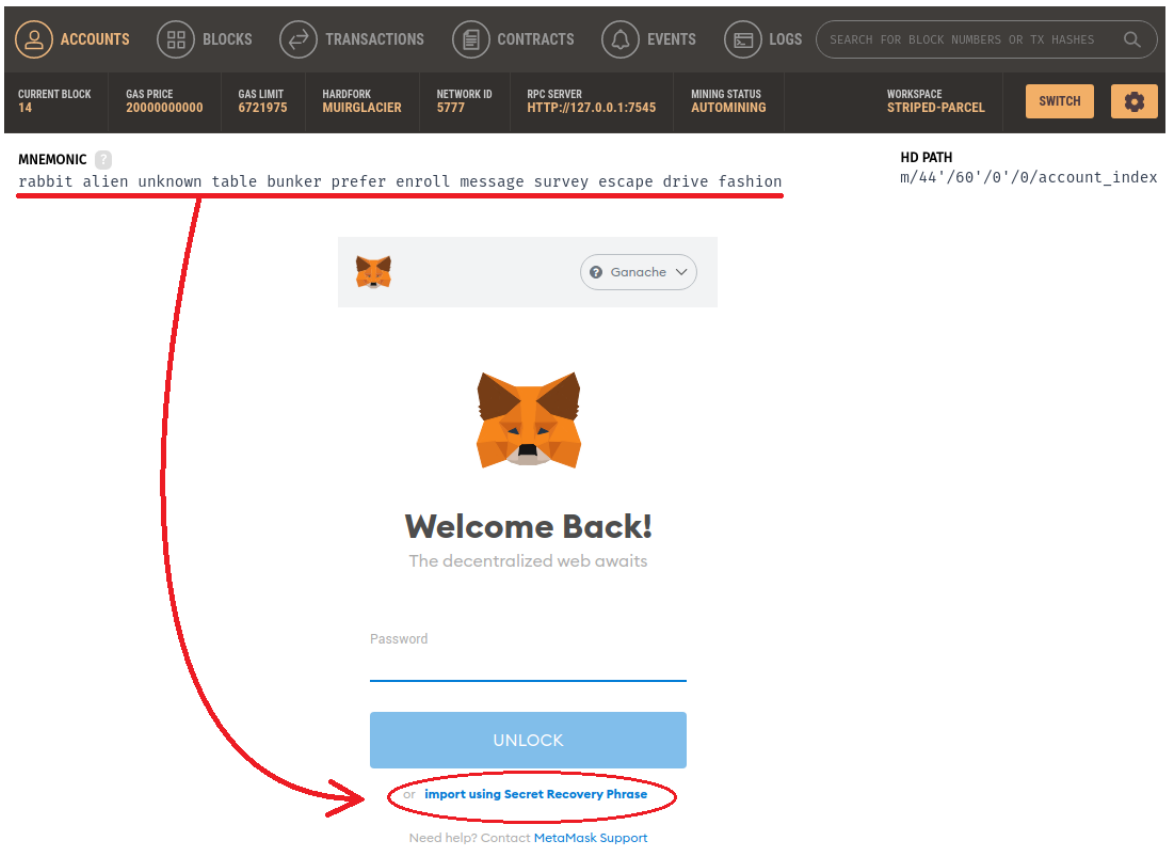

Ilustración 26: Configuración cunetas Metamask.

Al utilizar está red de Ganache, podremos observar fácilmente las transacciones realizadas a cada contrato. A continuación, se muestra en la *Ilustración 27*: Transacción de voto de un usuario. una transacción, que corresponde con la votación de un usuario, y en la que se pueden observar todos los datos de la misma.

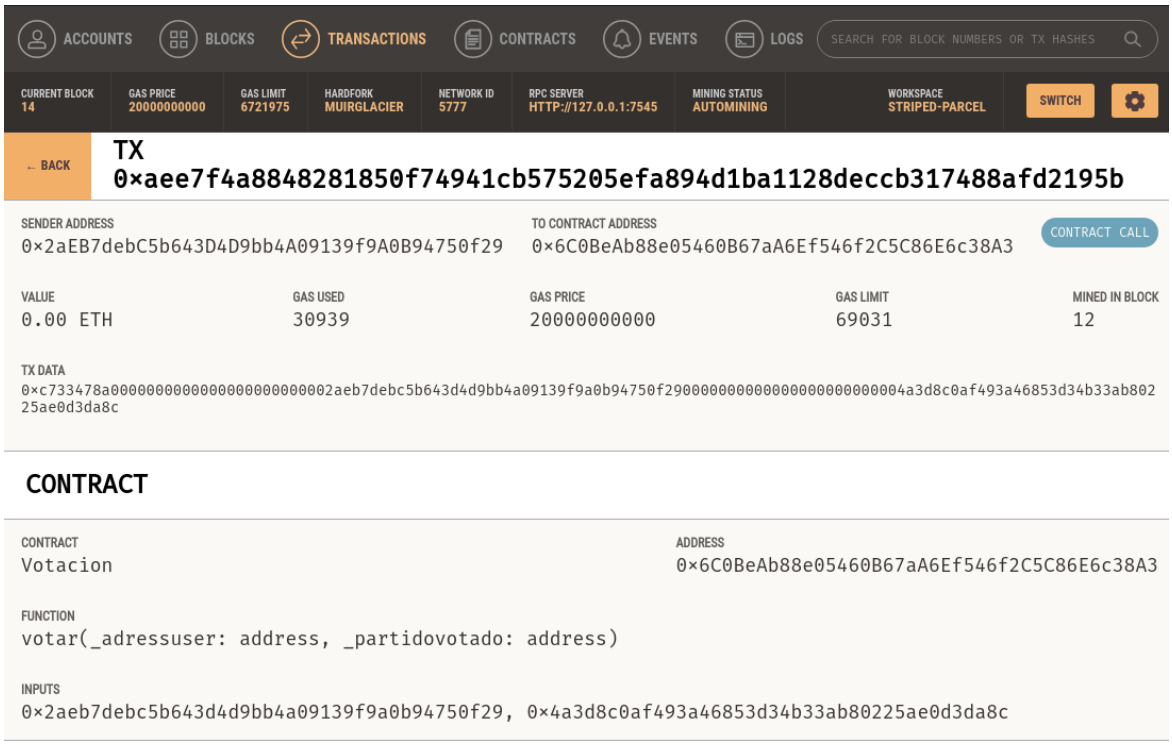

Ilustración 27: Transacción de voto de un usuario.

Para comprobar el cumplimiento de forma satisfactoria de todos los *Requerimientos y Objetivos* propuestos hay que analizar las transacciones realizadas, y comprobar las limitaciones propuestas por el sistema:

- Uno de los requerimientos trataba de que todos los usuarios pudiesen votar, y esto se cumple satisfactoriamente por el diseño del sistema, en el que el Smart Contract ERC-20, se inicializa con un input que decide el número de tokens(variable "TotalSupply") que se van a repartir. Esta cantidad, corresponde al número de gente que va a realizar las votaciones a través de este sistema, las cuales deberán estar registradas en la base de datos para poder utilizar la aplicación. Se muestra en

la *Ilustración 28* el constructor del Smart Contract que administra los tokens, donde se puede ver que necesita una variable de entrada "_censo", donde se introduce el número de votantes.

```
constructor(uint256 _censo) public{
    JEC = msg.sender;
    name = "Estoa";
    symbol = "STA";
    decimals = 0;
    totalSupply = _censo * (uint256(10)**decimals);
    balanceOf[msg.sender] = totalSupply;
}
```

Ilustración 28: Constructor del Smart Contract ERC20.

- La adición de partidos en el sistema no resulta ningún inconveniente, ya que a partir de la página diseñada para la configuración de las elecciones por la JEC(*Ilustración 9*), se encuentran las funcionalidades para añadir tantos partidos como se necesite, tanto a la Blockchain, como a la base de datos, aunque de forma individual y separada, por lo que habría que realizar la acción dos veces por cada partido.
- El sistema está diseñado para no permitir la realización de acciones por parte de los votantes antes de comenzar, o después de finalizar las elecciones. Esto se ha realizado mediante la creación de una variable booleana "estVotacion", que determina el estado de las votaciones, siendo verdadera una vez han comenzado las votaciones, y falsa antes de empezar, al igual que una vez terminadas. En la *Ilustración 29* se muestra un fragmento del código utilizado en el que se puede observar las funciones que modifican el valor de la variable booleana, y la función principal del sistema "votar", en la que se requiere un valor para esa misma variable para permitirla, por lo que no se podrá realizar dicha función si la variable no tiene el valor adecuado.

```
function votar(address _partidovotado)public {
    require(estVotacion==true);
    for(uint i=0; i < partidos.length; i++){
        require(msg.sender!=partidos[i], 'los partidos no pueden votar');
    }
    require(votoSC.balanceOf(msg.sender)==1,'el votante ya ha votado o no está registrado para votar');
    address sender = msg.sender;
    votoSC.transfer(sender , _partidovotado, 1);
}

function iniciarVotacion() public OnlyJEC{
    require(votoSC.balanceOf(JEC)==0, 'No están repartidos todos los tokens a los votantes');
    for(uint i=0; i < partidos.length; i++){
        require(votoSC.balanceOf(partidos[i])==0, 'Los partidos no estan inicializados correctamente');
    }
    estVotacion = true;
}

function finalizarVotacion() public OnlyJEC{
    estVotacion=false;
}
```

Ilustración 29: Muestra de la variable "estVotacion".

- Las cuentas de los partidos únicamente pueden recibir votos, no enviar ni votar, y esta lógica se puede observar en la recién mostrada *Ilustración 29*, en la función "votar", donde se requiere que el usuario que esté mandando la transacción no sea una cuenta de un partido, por lo que los tokens que son enviados a una cuenta de un partido no pueden volver a enviarse.
- Los últimos dos requerimientos restantes están relacionados, resulta que en este caso tenemos una base de datos que relaciona cada persona con una cuenta de Ethereum a través de su DNI, para poder iniciar sesión en la aplicación y saber a qué cuenta se tienen que repartir los tokens. Esto identifica a cada persona con su cuenta, y por tanto esas cuentas no son anónimas, ya que como se puede observar en la *Ilustración 27* la Blockchain almacena la transacción donde se indica entre que cuentas se ha realizado la transacción, por lo que conociendo de quien es la cuenta, como es el caso, se sabría a quien ha votado. Esto incumpliría uno de los requerimientos del sistema, el anonimato del voto.

Una posible solución sería no almacenar la cuenta de cada votante, pero eso llevaría a no poder comprobar si un usuario tiene más de una cuenta, y a no poder repartir los tokens a esa cuenta, lo que incumpliría otro requerimiento.

Esto ha llevado al planteamiento de que, dado que todas las transacciones quedan registradas en la red, y tanto su trazabilidad como su transparencia son completas, la mejor forma de que estas votaciones resulten anónimas sería repartir cuentas de forma aleatoria sin saber a quién pertenece cada cuenta. Resultando en una aplicación, en la que el usuario entra iniciando sesión con su DNI, y siendo propietario de una cuenta que únicamente él conoce, y una vez realice su votación, en la base de datos se actualice una variable booleana que indique que ya ha votado. El inconveniente de esta solución es que la creación de una cuenta nueva no dispondría de un token, por lo que sería necesario buscar una forma de repartir cuentas con tokens, o crear un registro para cuentas en el Smart Contract para que se envíe el token a esa cuenta, y fiarse de que al realizar esa acción no se asocie tu cuenta de Blockchain a tu DNI desde la aplicación web en alguna base de datos.

Capítulo 7. CONCLUSIONES Y TRABAJOS FUTUROS

Con este proyecto, por tanto, se ha diseñado un sistema de votaciones pensado para elecciones electorales, el cual a nivel lógico es totalmente funcional. Con ello este proyecto aporta la existencia de una alternativa más digitalizada al sistema actual, tanto como posible sustituyente, como sistema simultaneo, pudiendo agilizar el proceso a muchas personas y aportando las características de seguridad de una red Blockchain.

El problema que se ha detectado ha sido un problema por la trazabilidad de transacciones, ya que, la red Blockchain al ser tan transparente, permite rastrear las transacciones realizadas, y con ello, saber que cuentas han votado a un partido político. Lo que deriva en que, conociendo al propietario de la cuenta, se conocería su voto, y desaparecería el anonimato de las elecciones.

Como trabajos futuros que mejorarían el proceso y la aplicación, además de buscar un método viable para solucionar la cuestión mencionada anteriormente del anonimato, podría entrar en el modelo una importante mejora de automatización, ya que las pruebas realizadas, se han implementado con 3 usuarios como votantes, y dos partidos. Este proceso no era muy costoso manualmente debido a estos números de valor tan reducido, pero en el caso de unas elecciones reales habría que diseñar e implementar una automatización para que, por ejemplo, el reparto de tokens no deba ser manual, ya que sería muy costoso para el administrador. Esta automatización debería ser aplicable también al registro de partidos, aunque para estos habría también otro añadido previo, y es que se pueda registrar de forma simultánea un partido tanto en la Blockchain como en la base de datos.

Capítulo 8. BIBLIOGRAFÍA

[1] Blockchain And The Future of the Internet: A Comprehensive Review - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Mined-blocks-in-a-blockchain-Hash-in-each-block-now-starts-with-five-zeros_fig2_332139853 [accessed 31 May, 2021]

[2] "¿Qué es Ethereum? | ethereum.org", *ethereum.org*, 2021. [Online]. Available: https://ethereum.org/es/what-is-ethereum/. [Accessed: 05- Jul- 2021].https://ethereum.org/es/what-is-ethereum/

[3] "La Mejor Guía sobre la Tecnología Blockchain: Una Revolución para Cambiar el Mundo", *101 Blockchains*, 2021. [Online]. Available: https://101blockchains.com/es/tecnologia-blockchain/. [Accessed: 05- Jul- 2021].

[4] "Solidity — documentación de Solidity - UNKNOWN", *Solidity-es.readthedocs.io*, 2021. [Online]. Available: https://solidity-es.readthedocs.io/es/latest/. [Accessed: 05- Jul- 2021].

[5] "Aplicaciones descentralizadas (dapps) | ethereum.org", *ethereum.org*, 2021. [Online]. Available: https://ethereum.org/es/dapps/. [Accessed: 05- Jul- 2021].

[6] "Node.js", *Node.js*, 2021. [Online]. Available: https://nodejs.org/es/. [Accessed: 05- Jul- 2021].

[7] "Sweet Tools for Smart Contracts | Truffle Suite", *Truffle Suite*, 2021. [Online]. Available: https://www.trufflesuite.com/. [Accessed: 05- Jul- 2021].

[8] "Truffle Suite: Herramientas de código abierto para Blockchain", *blog.desdelinux.net*, 2021. [Online]. Available: https://blog.desdelinux.net/truffle-framework-herramientas-codigo-abierto-blockchain/. [Accessed: 05- Jul- 2021].

[9] "Ganache | Truffle Suite", *Truffle Suite*, 2021. [Online]. Available: https://www.trufflesuite.com/ganache. [Accessed: 05- Jul- 2021].

[10] "MetaMask", *Metamask.io*, 2021. [Online]. Available: https://metamask.io/. [Accessed: 05- Jul- 2021].

[11] "Remix - Ethereum IDE", *Remix.ethereum.org*, 2021. [Online]. Available: https://remix.ethereum.org/. [Accessed: 05- Jul- 2021].

[12] V. Code, "Visual Studio Code - Code Editing. Redefined", *Code.visualstudio.com*, 2021. [Online]. Available: https://code.visualstudio.com/. [Accessed: 05- Jul- 2021].

[13] PhpMyAdmin contributors, "phpMyAdmin", *phpMyAdmin*, 2021. [Online]. Available: https://www.phpmyadmin.net/. [Accessed: 05- Jul- 2021].

[14] OpenZeppelin, "OpenZeppelin/openzeppelin-contracts", *GitHub*, 2021. [Online]. Available: https://github.com/OpenZeppelin/openzeppelin-contracts/tree/master/contracts/token/ERC20. [Accessed: 05- Jul- 2021].

[15] "Truffle | Running Migrations | Documentation | Truffle Suite", *Truffle Suite*, 2021. [Online]. Available: https://www.trufflesuite.com/docs/truffle/getting-started/running-migrations. [Accessed: 05- Jul- 2021].

[16] "Elecciones en España - Wikipedia", *Es.wikipedia.org*, 2021. [Online]. Available: https://es.wikipedia.org/wiki/Elecciones_en_Espa%C3%B1a. [Accessed: 05- Jul- 2021].

[17] "Sistema D'Hondt - Wikipedia ", *Es.wikipedia.org*, 2021. [Online]. Available: https://es.wikipedia.org/wiki/Sistema_D%27Hondt. [Accessed: 05- Jul- 2021].

[18] "Objetivos y metas de desarrollo sostenible", *Desarrollo Sostenible*, 2021. [Online]. Available: https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/. [Accessed: 10- Jul- 2021].

ANEXO I: ALINEACIÓN CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE(ODS)

Precio a la realización de este proyecto se plantearon y se tuvo en cuenta una alineación con los objetivos de desarrollo sostenible, los cuales se plantearon por los líderes mundiales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos. A continuación, en la *Ilustración 30* se muestra el conjunto de objetivos que se adoptaron:

Ilustración 30: Objetivos de desarrollo sostenible.[18]

Los objetivos que se tuvieron en cuenta durante este trabajo son tres, comenzando por orden de numeración:

- El número nueve, enfocado en construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación. Un objetivo importante a la

hora de desarrollar nuevas tecnologías, a través de las cuales se permite el uso eficiente de los recursos entre otras cosas. En este proyecto específicamente se intenta perseguir la primera de las metas indicadas por este objetivo (9.1), desarrollar infraestructuras fiables, sostenibles, resilientes y de calidad, incluidas infraestructuras regionales y transfronterizas, para apoyar el desarrollo económico y el bienestar humano, haciendo especial hincapié en el acceso asequible y equitativo para todos.

- En segundo lugar, el numero quince, enfocado en gestionar sosteniblemente los bosques, luchar contra la desertificación, detener e invertir la degradación de las tierras, detener la pérdida de biodiversidad. Objetivo planteado principalmente para la restauración de los ecosistemas, y muy relacionado con otro objetivo, "Acción por el clima". El desarrollo de este proyecto se planteó como un plan para velar por la conservación y el uso sostenible de los ecosistemas terrestres, en particular, a través de la sustitución de las papeletas tradicionales por votos electrónicos, lo que evitaría la tala de bosques para la producción de estas millones de papeletas, y con ello se aseguraría evitar la posibilidad de una deforestación.
- Y por último, la número dieciséis, enfocado en promover sociedades justas, pacificas e inclusivas. Buscando crear a todos los niveles instituciones eficaces y transparentes. Por lo que, un sistema aplicando la tecnología Blockchain, al aportar una transparencia y una fiabilidad que cumpliría estos requisitos con margen, sería una forma de crear un sistema totalmente justo y fiable, en el que todos podamos asegurarnos que se ha cumplido correctamente las acciones que hemos llevado a cabo, y no han sido manipuladas. En el caso de este proyecto, los votos de las personas.

Por lo que, a través de este sistema se busca alcanzar alguna de las metas propuestas por los objetivos de desarrollo sostenible para poder cumplir a través de ellos las propuestas realizadas por los líderes mundiales, y llevar a cabo todos sus propósitos.