UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Interfaz web 3.0 para un protocolo de voto electrónico desplegado en blockchain

Trabajo Fin de Máster

Máster Universitario en Ciberseguridad y Ciberinteligencia

AUTOR/A: González Campos, Héctor

Tutor/a: López Rodríguez, Damián

Director/a Experimental: LARRIBA FLOR, ANTONIO MANUEL

CURSO ACADÉMICO: 2022/2023

Resum

En el treball es planteja el desenvolupament d'una interfície web 3.0 per a un protocol de vot electrònic basat en smart contracts i proposat per un grup de la UPV. La interfície haurà d'interactuar amb una blockchain, registrant les interaccions dels usuaris i autoritats, però també accedint a tota la informació registrada per a realitzar l'escrutini i mostrar els resultats. En el projecte es planteja utilitzar Mumbai, testnet de Polygon, amb la qual el Backend interactuarà per a accedir als smart contracts. Aquests contractes, ja desplegats, contenen un prototip del protocol desenvolupat per l'equip d'investigació. Es planteja l'ús de diferents tecnologies com el framework React com frontend, Typescript, React-Boostrap i Material-ui. Destaca especialment la integració de Metamask, a través de llibreries especialitzades per al desenvolupaments web 3.0, com cartera de criptomonedas per a la gestió dels tokens que habiliten el vot, i la identitat digital dels votants. L'aplicació haurà de facilitar l'operativitat, oferint totes les funcions del protocol, des de la creació d'una votació, la creació i depòsit de paperetes, l'escrutini i la visualització d'informació referent a cada votació. Es planteja així mateix realitzar proves de validesa i correcció de la interfície i dels smart contracts.

Paraules clau: Blockchain, vot electrònic, web 3.0, smart contract

Resumen

En el trabajo se plantea el desarrollo de un interfaz web 3.0 para un protocolo de voto electrónico basado en smart contracts y propuesto por un grupo de la UPV. La interfaz deberá interactuar con una blockchain, registrando las interacciones de los usuarios y autoridades, pero también accediendo a toda la información registrada para realizar el escrutinio y mostrar los resultados. En el proyecto se plantea utilizar Mumbai, testnet de Polygon, con la que el Backend interactuará para acceder a los smart contracts. Estos contratos, ya desplegados, contienen un prototipo del protocolo desarrollado por el equipo de investigación. Se plantea el uso de distintas tecnologías como el framework React como frontend, Typescript, React-Boostrap y Material-ui. Destaca especialmente la integración de Metamask, a través de librerías especializadas para el desarrollos web 3.0, cómo billetera de criptomonedas para la gestión de los tokens que habilitan el voto, y la identidad digital de los votantes. La aplicación deberá facilitar la operatividad, ofreciendo todas las funciones del protocolo, desde la creación de una votación, la creación y depósito de papeletas, el escrutinio y la visualización de información referente a cada votación. Se plantea así mismo realizar pruebas de validez y corrección de la interfaz y de los smart contracts.

Palabras clave: Blockchain, voto electrónico, web 3.0, smart contract

Abstract

The research proposes the development of a web 3.0 interface for an electronic voting protocol based on smart contracts and proposed by a group from the UPV. The interface must interact with a blockchain, registering the interactions of users and authorities, but also accessing all the information registered to carry out the scrutiny and display the results. The project plans to use Mumbai, Polygon's testnet, with which the backend will interact to access the smart contracts. These contracts, already deployed, contain a prototype of the protocol developed by the research team. The use of different technologies such as the React framework as frontend, Typescript, React-Boostrap and Material-ui is proposed. Particularly noteworthy is the integration of Metamask, through specialised

libraries for web 3.0 developments, as a cryptocurrency wallet for the management of the tokens that enable voting, and the digital identity of voters. The application must facilitate operability, offering all the functions of the protocol, from the creation of a vote, the creation and deposit of ballots, the scrutiny and the visualisation of information regarding each vote. Validity and correctness testing of the interface and smart contracts is also envisaged.

Key words: Blockchain, electronic voting, web 3.0, smart contract

Índice general

Índice de figuras

CAPÍTULO 1
Introducción

En la era digital actual, la tecnología ha revolucionado diversos aspectos de nuestra vida cotidiana, incluyendo los procesos electorales. El voto electrónico se ha posicionado como una alternativa innovadora y prometedora para mejorar la eficiencia y transparencia en las elecciones. Además, la tecnología blockchain ha emergido como un pilar fundamental para garantizar la integridad y seguridad de los datos en diversas aplicaciones. En este contexto, el presente trabajo se enfoca en el diseño y desarrollo de una interfaz web 3.0 que integre un protocolo de voto electrónico desplegado en blockchain.

Adicionalmente, la criptografía juega un papel crucial en el voto electrónico al proporcionar mecanismos para proteger la información, asegurar la autenticidad e integridad de los votos, y garantizar el anonimato del votante. Estas capacidades criptográficas permiten que el proceso electoral no solo sea transparente, sino también resistente a manipulaciones y fraudes.

Sin embargo, es esencial reconocer que, aunque la combinación de blockchain y criptografía ofrece múltiples beneficios, también puede introducir una barrera tecnológica para muchos ciudadanos. El voto, más allá de ser un proceso tecnológico, es un acto social y democrático fundamental. Por lo tanto, es esencial que las soluciones tecnológicas sean accesibles y comprensibles para todos los votantes, independientemente de su nivel de familiaridad con la tecnología.

Es aquí donde la importancia de una interfaz bien diseñada cobra relevancia. Las interfaces deben ser intuitivas y transparentes, permitiendo a los usuarios comprender y confiar en el proceso de votación, sin necesidad de ser expertos en criptografía o blockchain. Esta investigación busca explorar el potencial de esta combinación tecnológica para contribuir a la modernización de los procesos democráticos y fortalecer la confianza ciudadana en las instituciones electorales, haciendo hincapié en la importancia de un diseño de interfaz centrado en el usuario.

1.1 Motivación

La motivación detrás de esta tesis surge de la necesidad imperante de mejorar los procesos electorales a nivel global. En muchos países, los sistemas de votación tradicionales enfrentan desafíos de seguridad, transparencia y accesibilidad, lo que puede socavar la legitimidad de los resultados electorales y la confianza en el sistema político. La tecnología blockchain ofrece características únicas, como la inmutabilidad y descentralización, que pueden contribuir a superar estos obstáculos y brindar una base sólida para el voto electrónico.

La creciente adopción de la web 3.0 y sus capacidades interactivas y personalizadas también presenta una oportunidad para mejorar la experiencia de votación, haciéndola más amigable y accesible para los ciudadanos de todas las edades. Al fusionar las ventajas de la tecnología blockchain con una interfaz web 3.0 intuitiva, esta investigación busca proporcionar una plataforma de voto electrónico innovadora y segura que se adapte a las necesidades de una sociedad cada vez más digitalizada.

1.2 Objetivos

1.2.1. Objetivo General

El objetivo general de este trabajo es diseñar e implementar una interfaz web 3.0 que facilite un protocolo de voto electrónico basado en blockchain, con el propósito de mejorar la eficiencia, transparencia y seguridad en los procesos electorales. A su vez reducir el gasto público que conlleva realizar unas elecciones de ámbito nacional. Fijándose en el presupuesto para las Elecciones Generales del 23 de julio de 2023 asciende a 220.872.805,92 euros (Interior, 2023).

1.2.2. Objetivos Especificos

- Investigar el estado actual del voto electrónico y analizar las diferentes tecnologías blockchain utilizadas en procesos electorales a nivel mundial.
- Diseñar y desarrollar una interfaz web 3.0 que ofrezca una experiencia de usuario intuitiva, senzilla y amigable para los votantes.
- Integrar un protocolo de votación basado en blockchain, asegurando la inmutabilidad y transparencia de los registros electorales.
- Evaluar la eficiencia y seguridad de la web 3.0 implementada, mediante pruebas y análisis exhaustivos.
- Proponer recomendaciones para la implementación y futuras mejoras del sistema, considerando aspectos técnicos y de usabilidad.

1.3 Metodología

- Investigación y Revisión de Literatura

  Este proceso implica un estudio de los trabajos existentes y las tecnologías utilizadas en el voto electrónico y la blockchain. Se realizará un análisis detallado de los sistemas existentes, sus ventajas y desventajas, y su pertinencia en el contexto actual. Se adoptará un enfoque crítico para sintetizar la información obtenida y formular la base para el desarrollo de la interfaz web y el protocolo de voto electrónico.

- Diseño de la Interfaz Web 3.0

  Basándonos en los hallazgos de la revisión de literatura y los principios del diseño de experiencia de usuario, procederemos a diseñar una interfaz de usuario prototipo que será simple, intuitiva y amigable para el usuario. Este proceso involucrará el uso de prototipado y pruebas iterativas para garantizar la facilidad de uso y la eficiencia del diseño.

- Integración del Protocolo de Votación basado Blockchain

  En esta fase, se integrará el protocolo de votación dentro de una red blockchain seleccionada. Aseguraremos la transparencia y la inmutabilidad de los registros de votación mediante la codificación y la implementación de las características de seguridad adecuadas. Este proceso requerirá un conocimiento profundo de las tecnologías blockchain y las técnicas de programación frontend, ya que se requiere de una comunicación eficaz y transparente para el usuario.

- Evaluación del Sistema

  Posteriormente, se evaluará el sistema de voto electrónico implementado. La evaluación se realizará mediante pruebas exhaustivas y análisis que incluirán pruebas unitarias, y de seguridad. Los resultados de estas pruebas proporcionarán información valiosa sobre la eficiencia y la efectividad del sistema.

- Recomendaciones para Implementación y Mejoras Futuras

  Finalmente, en función de los resultados obtenidos, se propondrán recomendaciones concretas para la puesta en marcha del sistema y futuras mejoras. Estas recomendaciones abordarán tanto aspectos técnicos como de usabilidad para garantizar una adopción sin problemas y un rendimiento óptimo en el futuro.

  La metodología adoptada garantiza un enfoque sistemático y coherente hacia el logro de los objetivos del estudio. Cada fase de la metodología es interdependiente y vital para el éxito de la investigación.

1.4 Estructura de la memoria

La memoria se inicia con una *Introducción* que establece la motivación para la investigación, los objetivos del estudio, la metodología utilizada y la estructura de la memoria en sí. La sección de *Objetivos* se divide en dos subsecciones que detallan el *Objetivo General* y los *Objetivos Específicos* del estudio. La sección de *Metodología* es más detallada, describiendo los seis pasos del proceso de investigación, desde la revisión de la literatura hasta las recomendaciones para futuras implementaciones y mejoras.

El capítulo 2, titulado **"Estado del Arte"**, presenta una revisión exhaustiva de la literatura existente sobre el sistema de voto, centrándose específicamente en la votación electrónica, los diferentes tipos de sistemas de votación electrónica y los principales retos que enfrenta esta tecnología. Este capítulo también examina varias tecnologías clave relacionadas con la votación electrónica y la Web 3.0, incluyendo blockchain, TAVS, Ethereum y Tecnologías Frontend, antes de concluir con una propuesta basada en la revisión de la literatura.

El capítulo 3, **"Tecnologías"**, se adentra en las distintas tecnologías y herramientas que son esenciales en el ámbito de la votación electrónica y la Web 3.0. Se abordan tecnologías como blockchain, TAVS, Ethereum y diversas tecnologías Frontend, proporcionando una base sólida para comprender las soluciones propuestas en capítulos posteriores.

El capítulo 4, **"Análisis del Problema"**, profundiza en el problema que esta memoria busca abordar, presentando un análisis detallado e identificando posibles soluciones.

El capítulo 5, **"Diseño de la Solución"**, presenta la arquitectura del sistema propuesto, el diseño de la interfaz de usuario y las tecnologías utilizadas en el desarrollo del sistema. Además, se describen las pruebas realizadas para validar la eficacia y la eficiencia del sistema.

El capítulo 6, **"Posible puesta en producción"**, explora las implicaciones de implementar el sistema propuesto en un entorno de producción real, considerando posibles retos y soluciones.

Finalmente, el capítulo 7 ofrece una conclusión a la memoria, resumiendo los hallazgos clave y sugiriendo áreas para futuros trabajos de investigación. En conjunto, esta memoria proporciona un análisis detallado y exhaustivo de la votación electrónica, así como la propia implementación de una interfaz web 3.0, un posible despliegue y trabajo futuro.

1.5 Convenciones

A continuación, se presenta las convenciones que seguirá este trabajo:

- El código fuente de la web se muestra en letra courier cursiva. Y sólo se empleará esta tipología para este tipo de contenido.
- Las palabras extranjeras estarán entre paréntesis.
- Se entrecomillarán las citas a los apartados de la obra.
- Si las figures provienen de una fuente se indicará en la misma.
- Conceptos criptográficos básicos a tener en cuenta:
  - **Hash:** Un *hash* es una función que convierte una entrada (o 'mensaje') en una cadena de caracteres de longitud fija, que típicamente parece aleatoria. La salida, conocida como valor hash, debería ser la misma longitud independientemente de la longitud de la entrada. Es computacionalmente difícil generar la misma salida hash a partir de dos entradas diferentes. Por lo tanto, incluso pequeños cambios en la entrada producirán un valor hash significativamente diferente (Menezes y col. 1996).
  - **Exponenciación modular:** La *exponenciación modular* es una técnica utilizada en computación para calcular el residuo cuando un número es elevado a una potencia y luego dividido por otro número. Matemáticamente, la exponenciación modular se expresa como a^b mód m, donde a es la base, b es el exponente, y m es el módulo. Esta operación es fundamental en la criptografía, especialmente en sistemas de clave pública como RSA (Rivest y col. 1978).

CAPÍTULO 2
Estado del Arte

2.1 Sistema de voto

El sistema de voto es el mecanismo a través del cual los ciudadanos expresan su voluntad en una variedad de decisiones públicas, desde la elección de sus representantes en el gobierno hasta la toma de decisiones sobre cuestiones de política pública. A lo largo de la historia, estos sistemas de votación han evolucionado para adaptarse a los avances tecnológicos y a las cambiantes necesidades de la sociedad.

La votación inicialmente se realizó de manera presencial y pública, a veces incluso a viva voz. Sin embargo, la necesidad de un voto secreto para proteger la integridad del proceso democrático llevó al desarrollo de la papeleta de papel en el siglo XIX, que permitía a los votantes marcar sus preferencias en privado antes de depositarlas en una urna para su conteo (Jarvis y Han, 2018)

La votación ha sido el mecanismo esencial para la toma de decisiones colectivas en sociedades democráticas. Esta acción, que tradicionalmente se ha realizado a través de medios físicos como papeletas y urnas, ha sufrido una evolución inevitable con la irrupción de la era digital. Los avances tecnológicos han llevado a la idea de que es posible agilizar y simplificar el proceso de votación, convirtiéndolo en algo más eficiente y, en teoría, más seguro.

Sin embargo, este camino hacia la digitalización de la democracia no está exento de obstáculos y preocupaciones. Mientras que la votación electrónica ofrece la posibilidad de alcanzar a un mayor número de votantes y de facilitar la rapidez en el conteo de votos, la integridad y confiabilidad de estos sistemas es un área de investigación y debate continuo. Las cuestiones sobre cómo proteger la infraestructura de votación contra potenciales amenazas cibernéticas y cómo garantizar que no haya manipulaciones malintencionadas son cruciales (Álvarez y Hall, 2008).

La votación por Internet, en particular, ha suscitado debates intensos y polarizados. Aunque proporciona una tecnología donde los votantes pueden ejercer su derecho desde cualquier lugar con conexión a Internet, es importante considerar las múltiples vulnerabilidades asociadas a la red. Las preocupaciones no se limitan únicamente a la seguridad técnica, sino también a la posibilidad de influencias externas y campañas de desinformación que pueden distorsionar el proceso democrático(Heiberg y col. 2015). Adicionalmente, si bien la tecnología ha avanzado, es crucial que se establezca un equilibrio en el que la innovación no supere la confianza y transparencia necesarias para que el proceso democrático funcione de manera óptima (Stewart, 2011).

En resumen, el sistema de votación ha evolucionado de manera significativa a lo largo del tiempo, y es probable que continúe haciéndolo en respuesta a los avances tecnológicos y a las cambiantes necesidades de la sociedad.

2.1.1. Retos principales

Los retos asociados con la votación electrónica son variados y complejos, y abarcan aspectos técnicos, organizativos y legales.

Seguridad: La integridad y confidencialidad de los votos en sistemas de votación electrónica son de suma importancia. Una de las principales preocupaciones es garantizar que las cuentas de los votantes y los resultados de la votación estén protegidos contra amenazas externas. Para abordar estos desafíos, la implementación de métodos criptográficos se ha convertido en una solución esencial. Utilizando algoritmos robustos como el RSA, es posible implementar técnicas criptográficas que se basan en problemas matemáticos sin solución eficiente conocida. Estas medidas criptográficas actúan como barreras contra el hacking, el malware y otras formas de ataques maliciosos, aumentado su seguridad significativamente (Suwarjono y col. 2021). Alguno de los posibles ataques pueden ser los siguientes:

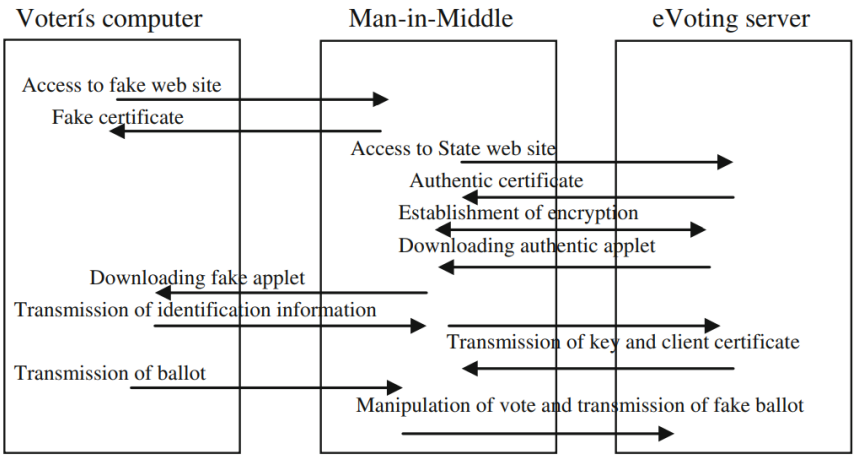

Figura 2.1: Vulnerabilidades de man-in-the-middle (Heiberg y col. 2015).

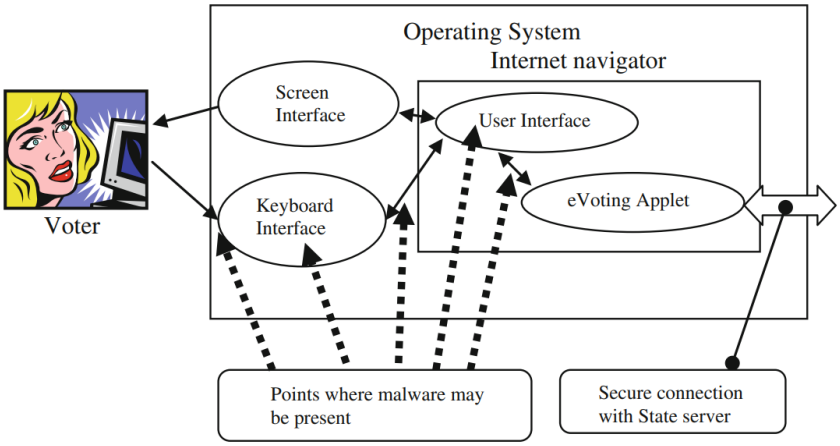

Figura 2.2: Vulnerabilidades de malware (Heiberg y col. 2015).

Privacidad: Los sistemas de votación electrónica deben proteger la privacidad de los votantes, asegurando que los votos sean secretos y que no se pueda rastrear a los votantes individuales. Esto puede ser particularmente desafiante en el caso de la votación por Internet, ya que los votos se transmiten a través de redes que podrían ser monitoreadas o interceptadas (Heiberg y col. 2015)

Accesibilidad: La accesibilidad en los sistemas de votación electrónica, especialmente para grupos específicos como las personas con discapacidad visual, aquellos con baja alfabetización y los ancianos. La accesibilidad no es solo una cuestión de diseño, sino una necesidad para garantizar la participación equitativa en los procesos democráticos. La co-creación con usuarios, como los previstos anteriormente, en las fases tempranas de conceptualización permite diseñar interfaces de usuario intuitivas, proporcionar instrucciones claras y ofrecer soporte adecuado para personas con discapacidades. Además, es fundamental que estos sistemas sean accesibles desde una variedad de dispositivos y ubicaciones, para facilitar a los votantes que pueden estar lejos de su lugar de votación habitual o que enfrentan otras barreras (Eijk y col. 2018).

Transparencia y verificabilidad: La blockchain en sistemas de votación aborda la seguridad y transparencia. Además de garantizar la privacidad y el anonimato mediante métodos criptográficos, asegura la verificabilidad e integridad de los resultados electorales. Una característica esencial de la verificabilidad es la "verificabilidad universal", que permite a cualquier observador, sin ningún privilegio especial, verificar que todos los votos han sido contados correctamente. La verificabilidad universal garantiza la transparencia en el proceso electoral y refuerza la confianza del público en el sistema. El código fuente del software de votación debe ser de acceso público, y la naturaleza inmutable del blockchain proporciona un registro transparente y seguro de los votos (Pereira y col. 2023).

Legalidad y regulación: La implementación de la votación electrónica puede enfrentar desafíos legales y regulatorios. Los sistemas de votación deben cumplir con las leyes y regulaciones electorales, que pueden variar de un lugar a otro. También puede haber desafíos en términos de obtener la aprobación y el apoyo de los legisladores, los partidos políticos, y el público (Heiberg y col. 2015)

En definitiva, la votación electrónica representa una evolución significativa en los procesos electorales, ofreciendo ventajas de eficiencia, accesibilidad y transparencia. Sin embargo, con estas ventajas vienen desafíos que deben ser abordados con delicadeza. Desde garantizar la seguridad y privacidad de los votantes hasta enfrentar desafíos legales y

regulatorios, es necesario que los sistemas de votación electrónica sean diseñados y implementados con una consideración cuidadosa de estos aspectos. La inclusión de tecnologías como blockchain puede ofrecer soluciones robustas a algunos de estos desafíos, pero la colaboración continua entre expertos técnicos, legisladores y la sociedad en general es fundamental para garantizar que la votación electrónica cumpla su promesa de mejorar la democracia sin comprometer la integridad del proceso electoral.

2.1.2. Votación electrónica

La votación electrónica es un término amplio que abarca una variedad de procesos de votación que utilizan tecnologías digitales. Ha sido implementada y utilizada en varios países para una serie de elecciones y votaciones, incluyendo elecciones presidenciales, parlamentarias y locales, y referéndums. A medida que la tecnología avanza, los sistemas de votación electrónica también están evolucionando para mejorar la eficiencia, la seguridad y la accesibilidad del proceso de votación.

Los sistemas de votación electrónica pueden ser implementados de varias maneras, que van desde máquinas de votación en las urnas hasta sistemas de votación en línea o por Internet. Los sistemas de votación en línea permiten a los votantes emitir sus votos desde cualquier lugar con acceso a Internet, mejorando así la accesibilidad y potencialmente aumentando la participación. Sin embargo, también plantean desafíos significativos en términos de seguridad y privacidad (Springall y col. 2014)

Las soluciones de seguridad, como el uso de criptografía avanzada, están siendo exploradas para abordar estos desafíos tan complicados. Los sistemas de votación electrónica también pueden implementar características de verificabilidad para garantizar que los votos se registren y cuenten correctamente, evitando sí la falsedad de los mismo. Esto puede implicar el uso "zero knowledge prove", que permiten a los votantes verificar que su voto se ha contado sin revelar cuál fue su voto, garantizando un anonimato del votante (Bernhard y col. 2017)

La adopción de la votación electrónica ha sido un tema de debate en muchos países. Mientras que algunos han adoptado completamente la votación electrónica, como Estonia (consultar Imagen 2.3), otros han sido más cautelosos debido a preocupaciones sobre la seguridad y la privacidad. Sin embargo, con las mejoras continuas en la tecnología y la creciente demanda de procesos de votación más eficientes y accesibles, es probable que la votación electrónica se vuelva cada vez más común en el futuro (Heiberg y col. 2015)

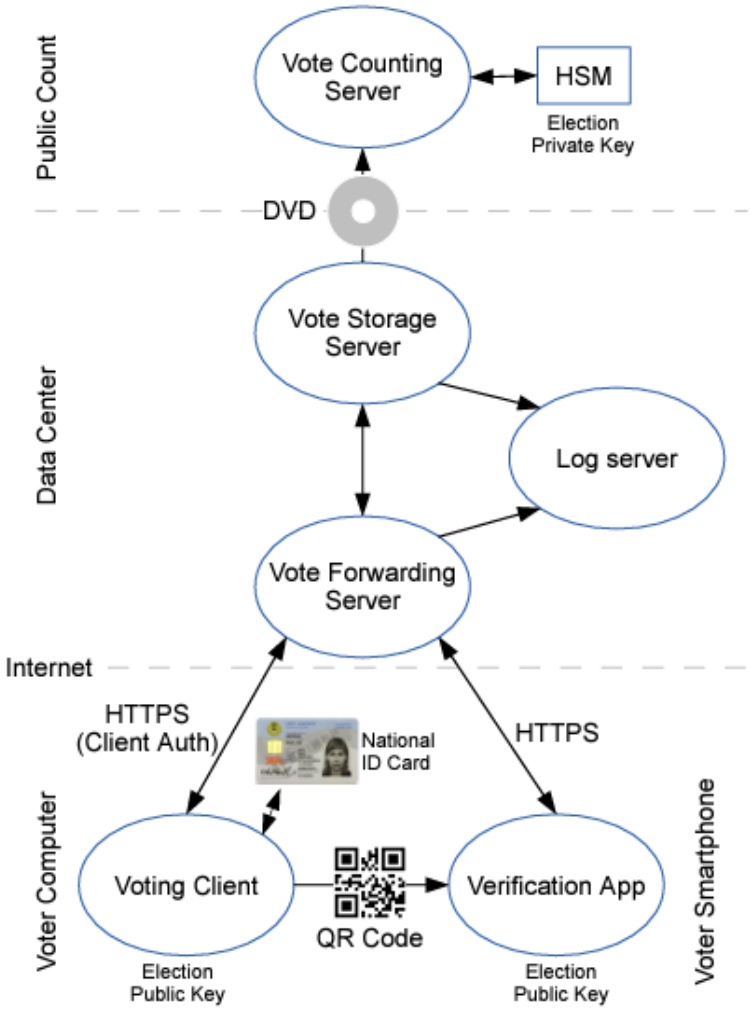

Figura 2.3: Sistema de I-voting de Estonia (Springall y col. 2014).

2.1.3. Tipos de sistemas de votación electrónica

Los sistemas de votación electrónica han evolucionado para satisfacer las necesidades cambiantes de la sociedad y abordar los desafíos asociados con la votación tradicional. Estos sistemas pueden clasificarse según el lugar de votación y el medio utilizado para emitir y contar los votos. Las principales categorías son:

- Votación electrónica en el lugar de votación (Polling place electronic voting).
- Votación por Internet (Internet voting).

Dada la creciente importancia y el potencial de la votación por Internet, este trabajo se centrará principalmente en este método, dada su capacidad de proporcionar mayor libertad y accesibilidad a los usuarios, así como los desafíos únicos que presenta. A continuación, se describen en detalle cada uno de estos métodos.

Votación electrónica en el lugar de votación

Este método se realiza en un lugar de votación designado, donde los votantes emiten sus votos en máquinas de votación electrónica. Estas máquinas pueden registrar directamente los votos en la memoria o utilizar tecnología de escaneo óptico para leer papeletas de papel marcadas por los votantes, proporcionando así una papeleta de respaldo auditable físicamente (Everett y col. 2008).

Votación por Internet

Permite a los votantes emitir sus votos desde cualquier ubicación con acceso a Internet, utilizando ordenadores personales o dispositivos móviles. La votación por Internet facilita la participación de votantes que se encuentren lejos de su lugar de votación habitual o que tengan dificultades para desplazarse (Springall y col. 2014).

Dentro de la categoría de votación por Internet, es posible clasificar los métodos según técnicas y tecnologías:

Basado en técnicas:

Votación basada en Mix-net: Garantiza el anonimato del votante utilizando una serie de servidores para mezclar los votos, asegurando que no puedan ser rastreados hasta el votante original.

Votación homomórfica: Utiliza propiedades criptográficas para garantizar la privacidad del voto. Gracias a la encriptación homomórfica, es posible realizar operaciones en datos cifrados sin descifrarlos primero.

Votación basada en firma ciega: Este enfoque emplea firmas criptográficas para asegurar la integridad del voto. Las firmas ciegas permiten obtener una firma en un mensaje sin revelar el contenido al firmante.

Basado en tecnologías:

Votación electrónica móvil: Es una variante específica de la votación por Internet que utiliza dispositivos móviles como teléfonos inteligentes y tabletas. La votación móvil puede ofrecer mayor accesibilidad y comodidad a los votantes, pero también plantea desafíos de seguridad y privacidad adicionales (Burton y col. 2015).

Votación basada en blockchain: Este método utiliza la tecnología blockchain para registrar y contar los votos, proporcionando una mayor transparencia y verificabilidad, ya que todos los votos se registran en un libro de contabilidad público y inmutable (Jafar y col. 2021).

2.1.4. Enfoques de Votación Electrónica por Internet: Ventajas y Desventajas

La necesidad de sistemas de votación que sean tanto seguros como eficientes se ha vuelto imperativa. A continuación, se presenta un análisis detallado de los principales enfoques de votación electrónica y sus características distintivas (Kho y col. 2022).

Votación Basada en Mix-net

El enfoque de votación basada en Mix-net, o redes de mezcla, implica que los votos cifrados enviados por los votantes se procesan a través de una serie de servidores independientes. Cada servidor desencripta y vuelve a cifrar los votos antes de pasarlos al siguiente servidor. Al final del proceso, los votos están completamente mezclados, asegurando el anonimato del votante (D. L. Chaum, feb. de 1981).

Ventajas:

- Proporciona una desvinculación entre los votantes y sus votos, garantizando que el voto de un individuo no pueda ser rastreado hasta él.
- Es altamente eficiente en la fase de recuento, optimizando el proceso de contabilización de votos.
- No requiere un alto costo de comunicación para la fase intensiva, lo que lo hace más eficiente en términos de transmisión de datos.

Desventajas:

- Su complejidad lo hace difícil de implementar en elecciones a gran escala.
- Es vulnerable a ataques DDOS, lo que puede interrumpir su operación.
- Requiere una prueba intensiva de conocimiento cero, lo que puede ser computacionalmente intensivo.

Votación Homomórfica

La votación homomórfica se basa en el principio de que es posible realizar operaciones en datos cifrados y obtener un resultado cifrado. Esto significa que se pueden sumar votos cifrados para obtener un total cifrado, que luego se puede descifrar para revelar el resultado final, sin nunca exponer el voto individual de un votante (Benaloh, 1987).

Ventajas:

- No requiere la desencriptación de los votos cifrados para contabilizar el resultado electoral.
- Ofrece una mayor seguridad y privacidad para los votantes, ya que sus votos permanecen cifrados durante todo el proceso.

Desventajas:

- No es adecuado para elecciones con múltiples candidatos debido al alto costo de cifrado.
- Requiere un canal anónimo, lo que puede ser difícil de implementar en entornos en línea.

Votación Basada en Firma Ciega

En la votación basada en firma ciega, el votante solicita que un tercero (por ejemplo, una autoridad electoral) firme su voto sin revelar su contenido. Una vez que el voto está firmado, el votante puede descifrar y enviarlo, asegurando que su voto es válido sin revelar su elección (D. Chaum, 1983).

Ventajas:

- Es simple y flexible, adaptándose a diversos escenarios.
- Es universalmente verificable, lo que garantiza la transparencia del proceso.

Desventajas:

- El factor ciego puede servir como recibo de votación, lo que puede ser explotado para coaccionar a los votantes.

Votación Basada en Blockchain

La votación basada en blockchain utiliza la tecnología de cadena de bloques para registrar y verificar cada voto. En este enfoque, cada voto se representa como una transacción en la blockchain. Una vez que se añade a la cadena, el voto se vuelve inmutable y no puede ser alterado ni eliminado. La naturaleza descentralizada de la blockchain garantiza que no haya una única entidad en control del proceso, lo que aumenta la transparencia y reduce el riesgo de manipulación. Además, la blockchain proporciona un registro auditable de todas las transacciones, lo que facilita la verificación posterior de los resultados (Noizat, 2015).

Ventajas:

- Proporciona un registro inmutable y transparente de cada voto.
- Reducción del riesgo de manipulación debido a su naturaleza descentralizada.
- Facilita la auditoría y verificación de los resultados.

Desventajas:

- Puede enfrentar problemas de escalabilidad al manejar un gran número de votantes simultáneamente.
- La adopción de blockchain para votación todavía es nueva y puede enfrentar resistencias regulatorias o políticas.

En conclusión, cada enfoque de votación electrónica tiene sus propias fortalezas y debilidades. La elección del enfoque adecuado dependerá de las necesidades específicas y los desafíos de cada escenario electoral. Es esencial que los diseñadores de sistemas de votación consideren cuidadosamente estos factores al seleccionar un enfoque para garantizar la integridad, seguridad y eficiencia del proceso electoral.

CAPÍTULO 3
Tecnologías

3.1 Blockchain

Blockchain, un término que ha acaparado gran atención en los últimos años, es un libro de contabilidad digitalizado y descentralizado que registra todas las transacciones entre pares sin necesidad de una autoridad centralizada. Esta tecnología consiste esencialmente en una serie de 'bloques' encadenados mediante complejos algoritmos computacionales. Cada bloque de esta cadena contiene información del bloque creado antes que él, formando una cadena interconectada de bloques. La cadena de bloques funciona en una red de ordenadores, también conocidos como nodos, cada uno de los cuales tiene acceso a toda la base de datos y a un historial de transacciones a partir del primer bloque, denominado "bloque génesis".

El nacimiento de la tecnología blockchain está estrechamente ligado a la aparición de Bitcoin, una popular criptomoneda. Satoshi Nakamoto, el creador seudónimo de Bitcoin, introdujo por primera vez el concepto de blockchain en un WhitePaper de 2008 (Nakamoto, 2008) que circuló entre los entusiastas de la criptografía. Al año siguiente, Nakamoto ya había publicado Bitcoin como software de código abierto y había acuñado los primeros Bitcoins, lo que marcó el éxito de la implantación de la tecnología blockchain (Woodside y col. 2017).

El funcionamiento de la tecnología blockchain implica la creación de nuevos bloques mediante un proceso conocido como "hashing". En este proceso, los datos que contienen una o más transacciones se recogen en la parte de datos de un bloque. Una copia de esta información se convierte en hash, un proceso que convierte una entrada de letras y números en una salida cifrada de una longitud fija. A continuación, este hash se empareja con otro hash, se vuelve a hashear, se vuelve a emparejar y se vuelve a hashear, lo que da como resultado un único hash denominado raíz. Como cada nuevo bloque contiene información del bloque creado anteriormente, los bloques se encadenan de forma que encajen computacionalmente en la cadena de bloques (Figura 3.1). Esta estructura garantiza la seguridad y la integridad de los datos almacenados en la cadena de bloques, ya que para alterar cualquier información sería necesario cambiar todos los bloques posteriores, lo que resulta poco práctico desde el punto de vista computacional.

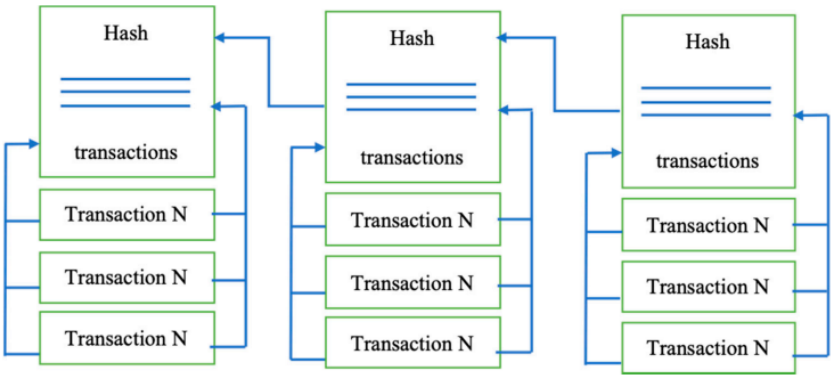

Figura 3.1: Estructura Blockchain (Jafar y col. 2021).

La arquitectura de la tecnología blockchain consta de varios componentes básicos, como se muestra en la figura 3.2.

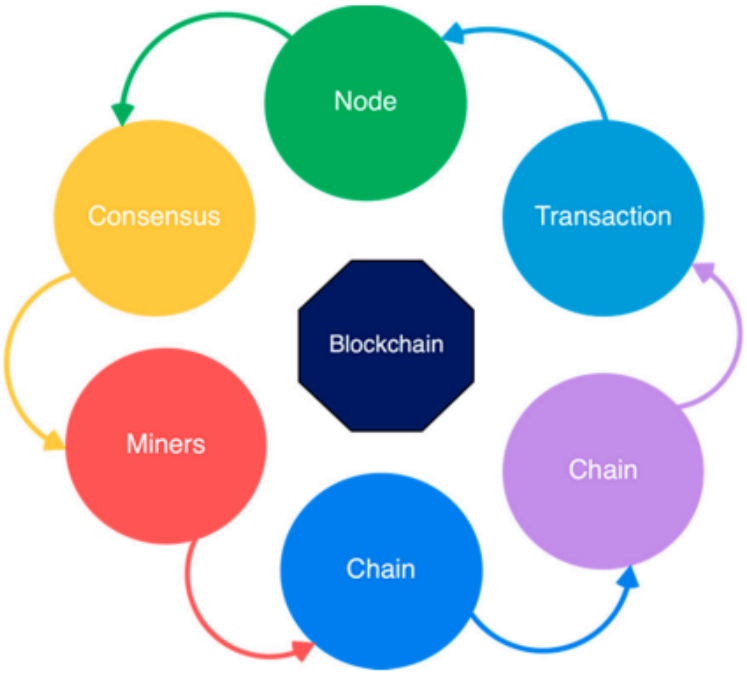

Figura 3.2: Componentes Blockchain (Jafar y col. 2021).

Los nodos son usuarios o computadoras que forman parte de la red blockchain. Cada uno de estos nodos mantiene una copia única del registro, asegurando la descentralización del sistema. Las transacciones, que son las operaciones registradas, se agrupan en bloques antes de ser añadidas a la cadena.

La cadena es una serie ordenada de bloques, que crea un registro ininterrumpido de todas las transacciones desde el inicio del *blockchain*. En este ecosistema, los mineros tienen un papel crucial. Estos mineros son nodos especializados que utilizan capacidad computacional para validar y verificar las transacciones. A través de un proceso denominado "proof of work" (PoW), los mineros compiten para resolver acertijos matemáticos complejos. El primero que resuelve dicho acertijo tiene el derecho de añadir el siguiente

bloque a la cadena y, como recompensa por su esfuerzo, recibe criptomonedas. Esta actividad no solo asegura la integridad y seguridad de las transacciones, sino que también mantiene la coherencia de la cadena.

Es esencial mencionar que, aunque PoW ha sido el mecanismo de consenso dominante en muchas *blockchains*, existen alternativas que buscan resolver los desafíos asociados con PoW. Una de esas alternativas es el "Proof of Stake" (PoS). En contraposición al PoW, donde los mineros compiten utilizando capacidad computacional para validar transacciones y crear nuevos bloques, en PoS, la creación de bloques se logra a través de la demostración de posesión de monedas existentes. Es decir, en lugar de usar energía para resolver acertijos matemáticos, en PoS, los participantes demuestran su "apuesta" o inversión en la criptomoneda para validar transacciones y crear bloques. Esta alternativa busca ser más eficiente en términos energéticos y sostenible a largo plazo.

El sistema PoS añade un nivel extra de seguridad y equidad. Dado que los validadores tienen un interés en actuar con honestidad (ya que tienen algo en juego), se reduce la posibilidad de ataques malintencionados. Además, al no requerir una vasta capacidad computacional, PoS permite una descentralización más amplia. Esto significa que más participantes pueden unirse al proceso de validación sin la necesidad de un hardware especializado y costoso (Kiayias y col. 2017).

3.2 TAVS: Two-Authorities Voting Scheme

El Sistema de Votación de Dos Autoridades (TAVS, por sus siglas en inglés "Two-Authorities Voting Scheme") (Larriba, Sempere y col. 2020) es una propuesta de sistema de votación electrónica basado en firmas ciegas. Este sistema está diseñado para ser simple y escalable, lo que lo hace adecuado para varios tipos de elecciones. Los autores limitan el número de autoridades a dos, reducen el número total de operaciones modulares y proponen un método para disminuir las interacciones necesarias para emitir un voto, lo que lo hace fácil para el votante a su vez. Como resultado, la complejidad del protocolo de votación escala linealmente con el número de votos.

El TAVS se basa en dos entidades no relacionadas: una Autoridad de Identificación (IA) que verifica la pertenencia del votante en el censo (PBB), y una Estación de Votación Remota (RPS) donde los electores emiten sus votos. Este sistema permite al elector verificar de forma anónima que su voto ha sido incluido en el recuento. El recuento final puede ser auditado de forma anónima para verificar la corrección del recuento (Figura 3.3).

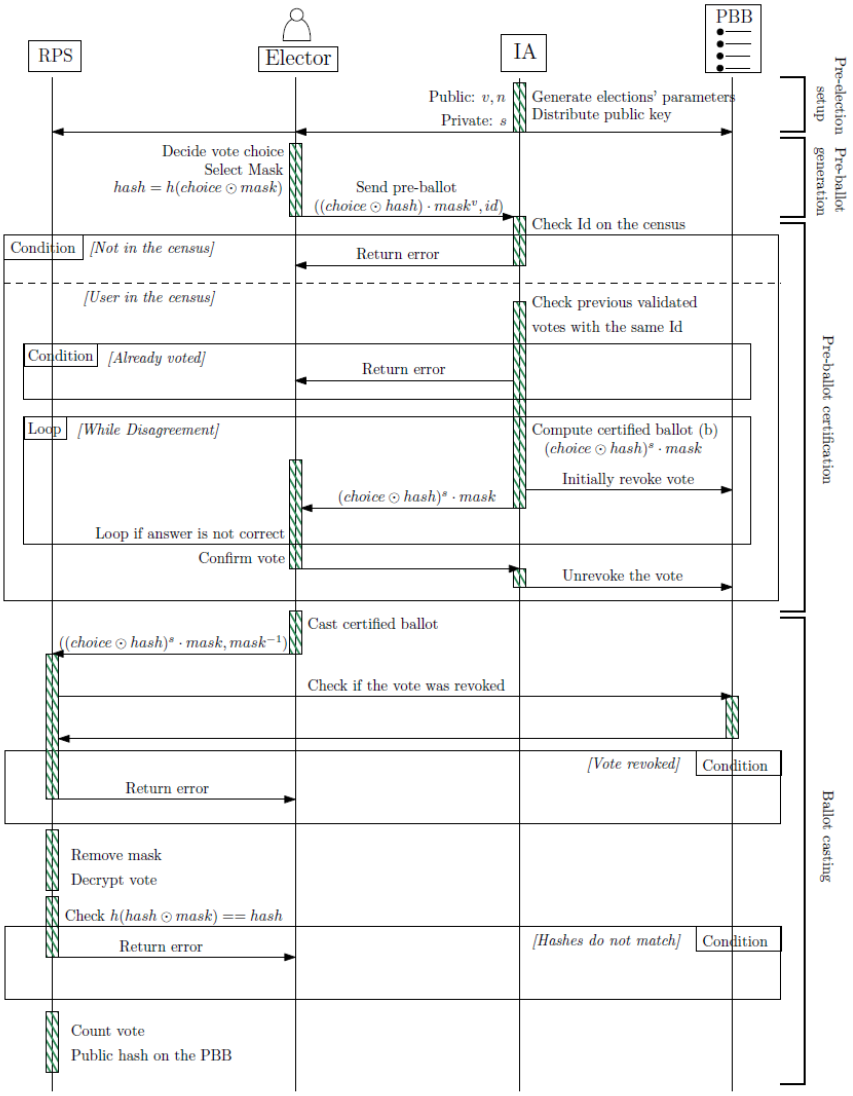

Figura 3.3: Todo el proceso que un elector debe completar para emitir un voto (Larriba, Sempere y col. 2020).

3.2.1. Creación del Voto:

1. **Pre-ballot Generation:** Antes del proceso de votación, es necesario acordar los métodos que se utilizarán para el hashing y un procedimiento de firma electrónica.

2. **Generación de la Pre-Boleto:** Una vez distribuido el componente público de la clave de firma de la IA, el elector puede generar un pre-boleto con su voto. Este pre-boleto debe estar oculta de tal manera que nadie excepto el elector pueda determinar la dirección del voto. Además, para prevenir el doble voto, la máscara debe estar vinculada al voto.

3.2.2. Firma del Voto:

1. **Generación de Clave RSA por IA:** Antes del inicio de las elecciones, la IA debe generar una clave pública de firma electrónica y transmitir el componente público de esta clave para permitir que cada miembro en el censo verifique la correcta validación de el boleto.

2. **Certificación del Boleto:** La IA responde al elector con un boleto certificada que se considera inválida hasta que el elector confirma el voto.

3.2.3. Verificación del Voto:

1. **Verificación por RPS:** Una vez que el RPS recibe el boleto y la inversa de la máscara, se aplica un algoritmo para recuperar el voto del elector y el hash para publicar en el tablón de anuncios.
2. **Integridad del boleto:** La computación de $h(\text{vote} \times \text{mask})$ permite al RPS verificar la integridad del boleto.

Las ecuaciones y algoritmos específicos para cada paso se pueden encontrar en el artículo original (Larriba, Sempere y col. 2020). Estos incluyen la generación de la máscara para la firma ciega, la verificación de la firma y el proceso de recuento.

Además, TAVS puede ser fácilmente extendido y escalado para adaptarse a situaciones más generales, ya que su complejidad no depende del número de candidatos, del número de autoridades involucradas o de la codificación del voto. Esto significa que el TAVS puede ser utilizado en una variedad de contextos de votación.

El sistema también proporciona mecanismos para detectar comportamientos maliciosos de las autoridades. Aunque se asume que las autoridades no están relacionadas de ninguna manera, el sistema está diseñado para funcionar correctamente incluso si una o ambas autoridades actúan de manera deshonesta.

En resumen, TAVS es un sistema de votación electrónica que busca mejorar la eficiencia y simplicidad de los procesos de votación, al tiempo que mantiene la seguridad y la privacidad de los votantes (Larriba, Sempere y col. 2020).

3.3 Ethereum

Ethereum, propuesto inicialmente por Vitalik Buterin y desarrollado en colaboración con Gavin Wood y otros, es un sistema que procesa transacciones de manera continua. Imagina Ethereum como un juego de mesa en constante evolución. Cada jugador realiza movimientos (transacciones) que cambian el estado del juego (el mundo de Ethereum). Este estado del juego puede contener todo tipo de información, como cuánto dinero virtual tiene cada jugador, acuerdos entre jugadores, y más (Buterin, 2013; Wood, 2014).

La Máquina Virtual de Ethereum (EVM) es el entorno en el que se ejecutan los contratos inteligentes, que son scripts programables que se ejecutan en la red Ethereum. Cada nodo de Ethereum ejecuta su propia implementación de EVM, lo que le permite ejecutar las mismas instrucciones. La EVM está completamente aislada de la red, el sistema de archivos y otros procesos de la computadora anfitriona, lo que la convierte en un entorno perfecto para ejecutar código no confiable. La EVM se sitúa encima de la cadena de bloques y funciona como un compilador que utiliza el software de los contratos inteligentes escrito en Solidity y lo compila para la EVM en lo que se denomina bytecode EVM, como se muestra en la figura 3.4.

Las transacciones en Ethereum no son simplemente transferencias monetarias, sino que también pueden contener código ejecutable denominado contrato inteligente (Smart Contract). Cuando se realiza una transacción a un contrato, el código del contrato se ejecuta en la EVM, y el contrato puede leer y escribir en su propio almacenamiento, llamar a otros contratos e incluso crear nuevos contratos. Todo esto se hace de manera descentralizada, con los resultados de la ejecución registrados en la cadena de bloques.

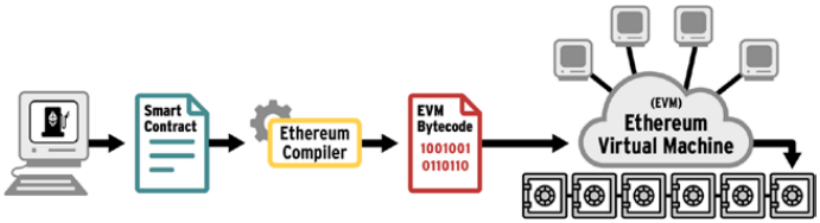

Figura 3.4: Ethereum Virtual Machine (Zhang y Anand, 2022).

Ethereum también introduce el concepto de gas, que es una medida del esfuerzo computacional. Cada operación que se ejecuta en la red Ethereum consume una cierta cantidad de gas. Los usuarios deben pagar por este gas para incentivar a los mineros a incluir sus transacciones en la cadena de bloques. Este mecanismo evita el colapso de la red y limita la cantidad de trabajo computacional que se realiza.

En conclusión, Ethereum extiende el concepto de la cadena de bloques desde un simple libro de transacciones hasta un motor computacional completamente desarrollado, permitiendo la creación y ejecución de aplicaciones descentralizadas complejas.

3.4 Implemetación en Solidity de TAVS

En el contexto de Ethereum, que es una red global para la computación distribuida, los smart contracts permiten implementar funcionalidades personalizadas y arbitrarias. Ethereum introdujo el concepto de gas para abordar las amenazas de los usuarios malintencionados que podrían colapsar la red, obligando a los usuarios a pagar (en la misma moneda que emplea la red) por las unidades de cómputo. Solidity es el lenguaje de programación de soporte completo de Turing que permite implementar estos smart contracts en Ethereum.

En este trabajo, los autores presentan una implementación de Solidity del protocolo de votación electrónica TAVS como un smart contract. Reemplazan una de las dos autoridades en TAVS con un smart contract inmutable en Ethereum. Al hacerlo, su implementación extiende las propiedades de seguridad de TAVS y logra un mayor grado de resistencia, verificabilidad y disponibilidad. El código de la implementación es de código abierto .

La implementación en Solidity se ha probado en una red local compatible con EVM y también se ha implementado en una red de prueba real. Los autores utilizaron la red de prueba de Mumbai debido al alto costo de implementar directamente en la red principal de Ethereum.

Por último, los autores señalan que ser público y ser accesible son dos cosas diferentes. Aunque cualquier error en el código será público y las personas lo explotarán para obtener una recompensa económica, los autores han realizado pruebas para abordar las amenazas de los usuarios malintencionados que podrían explotar cualquier error en el código del smart contract (Larriba y López, 2023).

3.5 Tecnologías Frontend

React es una biblioteca de JavaScript para la creación de interfaces de usuario. Fue desarrollada por Facebook y desde su creación en 2013 se ha convertido en una de las bibliotecas de JavaScript más utilizadas para la creación de aplicaciones (Facebook, 2023). React permite a los desarrolladores crear componentes web que pueden actualizarse y renderizarse de manera eficiente en respuesta a los cambios de datos, interacción del usuario, mejorando así el rendimiento de las aplicaciones web. Este enfoque basado en componentes promueve la reutilización de código y facilita la gestión de la complejidad a medida que las aplicaciones crecen en tamaño.

Para facilitar el desarrollo y mantener un diseño consistente en las aplicaciones basadas en React, se han desarrollado varias bibliotecas de componentes. Una de estas es React Bootstrap (R.-B. Team, 2015), que proporciona una colección de componentes de React predefinidos que siguen los estilos y las directrices de Bootstrap, el marco de diseño web más popular. Esto permite a los desarrolladores centrarse en la lógica de la aplicación, mientras que la biblioteca se encarga de la mayoría de los detalles de la interfaz de usuario.

Por otro lado, en el desarrollo de aplicaciones descentralizadas (Dapps) basadas en la blockchain de Ethereum, se utilizan bibliotecas como Ethers.js (Moore, 2015). Ethers.js es una biblioteca compacta y completa de JavaScript y TypeScript que proporciona funciones para interactuar con la red Ethereum. Esto incluye la manipulación de Bignums, transacciones, contratos inteligentes, herramientas de criptografía y mnemotécnicos.

Para interactuar con estas Dapps, se requiere una cartera de Ethereum. Una opción popular es MetaMask (ConsenSys, 2016), que permite a los usuarios acceder a sus cuentas de Ethereum a través de una extensión del navegador o una aplicación móvil. MetaMask proporciona una interfaz de usuario para la gestión de identidades y permite a los usuarios administrar las claves de sus cuentas de Ethereum, realizar transacciones, firmar mensajes y acceder a Dapps de manera totalmente gratuita.

3.6 Propuesta

En la era digital, donde la información fluye rápidamente y las tecnologías evolucionan a un ritmo sin precedentes, el desafío no es simplemente adaptarse, sino hacerlo de manera eficaz y amigable para el usuario. Esta propuesta se centra en desarrollar una interfaz web 3.0 para el protocolo TAVS, implementado en Solidity(Larriba y López, 2023).

Una interfaz bien diseñada no es un lujo, sino una necesidad. La confianza del elector en un sistema de voto electrónico se construye en gran parte a través de su interacción con la interfaz. Es por ello que, aunque aprovechemos las innovaciones de la web 3.0, la plataforma debe ser intuitiva y accesible para todos, independientemente de su nivel de conocimiento tecnológico.

La transparencia y eficiencia en la interacción con la blockchain son cruciales. Sin embargo, esta complejidad técnica debe presentarse al usuario de una manera que sea fácil de entender y navegar. La consistencia en el diseño y la presentación de la información son esenciales para generar confianza y reducir la confusión.

La visualización de los resultados electorales es un aspecto vital en cualquier proceso de votación. Aunque esta funcionalidad no está presente en la implementación de Solidity (Larriba y López, 2023), su integración en esta plataforma proporciona un valor añadido significativo, permitiendo al electorado una comprensión clara y gráfica de los resultados.

El corazón de esta propuesta radica en permitir votar y establecer elecciones. Estas funciones no sólo son esenciales desde el punto de vista operativo, sino que también representan la esencia del voto electrónico: brindar al electorado un sistema seguro, transparente y fácil de usar.

En conclusión, diseñar y desarrollar una interfaz para un sistema de voto electrónico no es una tarea trivial. Requiere una cuidadosa consideración de las necesidades del usuario, así como una profunda comprensión de la tecnología subyacente. Esta propuesta se esfuerza por combinar ambos aspectos, ofreciendo una solución innovadora y centrada en el usuario.

CAPÍTULO 4
Diseño de la solución

La concepción de una aplicación de voto electrónico basada en blockchain no solo implica la implementación de tecnologías avanzadas, sino también la creación de una solución intuitiva y segura para los usuarios. El diseño, por lo tanto, es un aspecto crítico que determina la eficacia, seguridad y facilidad de uso de la plataforma. En este apartado, se explorará la arquitectura del sistema, se detallarán las decisiones de diseño tomadas, se describirán las tecnologías empleadas y se abordará el proceso de pruebas y verificación. Todo ello con el objetivo de ofrecer una visión clara de cómo se ha estructurado y construido esta solución para satisfacer las necesidades de un voto electrónico confiable y accesible.

Desde el punto de vista técnico, se ha optado por un frontend construido con React, aprovechando la flexibilidad y robustez que esta biblioteca ofrece. En situaciones donde se requiere una tipificación más estricta o funcionalidades avanzadas, se ha incorporado TypeScript, garantizando así un código más limpio y mantenible. Además, para asegurar una integración fluida con el mundo blockchain, se utiliza la biblioteca 'ethers', que facilita la comunicación con Metamask, el método de autenticación escogido. Metamask no solo es ampliamente reconocido y confiable en la comunidad blockhain y web 3.0, sino que también proporciona una capa adicional de seguridad y transparencia para los usuarios.

Con este diseño, se busca ofrecer una solución de voto electrónico que combine lo mejor de la tecnología blockchain con una experiencia de usuario fácil y e intuitiva.

4.1 Análisis del problema

4.1.1. Identificación y análisis del problema

A pesar del progreso positivo y las capacidades técnicas de TAVS, la implementación actual revela un área importante que debe abordarse: la falta de una interfaz de usuario conveniente y de fácil acceso. Aunque los algoritmos y las técnicas utilizadas en este proceso parecen simples y prometedores, su desempeño sufre de esta falla.

Actualmente, la interacción con TAVS se limita a tecnologías muy avanzadas y está reservada para aquellos que saben cómo escribir y probar software. Esta limitación también excluye al grupo más grande que se beneficiaría del sistema: los ciudadanos comunes que buscan nuevas formas seguras de ejercer su derecho al voto. Además, las medidas actuales dificultan las cosas para quienes participan en el proceso electoral, desde asesores hasta administradores electorales y contadores. Por lo que a parte del ahorro monetario, ofrece facilidades para la administración de la elección.

La votación, por naturaleza, debe ser inclusiva. Pero sin una manera fácil de interactuar con TAVS, muchos quedan excluidos. No basta con tener un buen motor de votación si la gente no puede, o no sabe cómo, usarlo. En resumen, para que TAVS sea realmente revolucionario y útil en el mundo real, necesita una interfaz que todos puedan usar sin problemas.

4.1.2. Solución propuesta

Para ello, las bases de la propuesta son las siguientes:

- Autenticación basada en blockchain utilizando Metamask y la red Ethereum.
- Tres principales funciones a desarrollar: creación de elecciones, votar en las elecciones disponibles y consultar los resultados de las mismas.
- Realizar pruebas unitarias para verificar su funcionalidad y seguridad.

Es importante mencionar que la implementación de TAVS en Solidity carece de la implementación total de TAVS. TAVS se basa en dos entidades no relacionadas: una Autoridad de Identificación (IA) que verifica la pertenencia del votante en el censo (PBB), y una Estación de Votación Remota (RPS) donde los electores emiten sus votos. Siendo La implementación en Solidity de TAVS, la RPS, aun faltaría esa autoridad IA que verificase la pertenecía de los votantes al censo. En el último apartado de esta memoria (*Conclusión y trabajo futuro* 6) se detalla ese añadido y su implantación en este proyecto.

Para implementar esta propuesta de manera efectiva, es esencial proteger la aplicación, garantizando que solo los usuarios autenticados puedan acceder a sus funcionalidades, eso quiere decir que han iniciado sesión en Metamask. Esta necesidad se deriva del hecho de que Ethereum opera a través de transacciones para comunicarse con los contratos inteligentes, lo que implica la necesidad de contar con una cuenta de origen para generarlas. Además, en situaciones donde se necesite gas como medio de transacción, es imprescindible transferir la cantidad adecuada para llevar a cabo el proceso de minería.

De tal manera, para crear las funcionalidades descritas, hay que realizar ciertas modificaciones en el backend, las cuales se mencionarán más adelante. En particular, se necesita revisar la implementación de TAVS en Solidity. No se trata de alterar su algoritmo, sino de mejorar los 'getters' para facilitar la lectura de los datos almacenados en el Smart Contract. De esta forma, se reduce la carga computacional en el frontend encargada de interpretar los datos, permitiendo obtenerlos directamente de la blockchain. En cuanto al frontend, se requiere crear 3 vistas diferentes para cada funcionalidad, y en concreto para visualizar los resultados se necesitan mostrar de manera gráfica.

En relación a las pruebas unitarias, el principal objetivo es garantizar la seguridad de la web 3.0, verificando que el usuario está debidamente autenticado y que no genera conflictos en la blockchain. Por otra parte, es crucial limitar la libertad del usuario al crear elecciones para prevenir posibles errores. Para lograr esto, es necesario implementar filtros y realizar tantas verificaciones como sean necesarias. Esto nos asegurará de que solo la información correcta y validada es enviada a la blockchain.

4.2 Arquitectura del Sistema

La arquitectura del sistema describe la estructura general y el flujo de comunicación entre los diferentes componentes de la aplicación de voto electrónico. Esta sección detalla

las funcionalidades principales, el flujo de interacción del usuario, la comunicación con los contratos inteligentes, el diseño de la interfaz, entre otros aspectos clave que conforman la solución.

4.2.1. Funcionalidades del sistema

La aplicación de voto electrónico presenta tres funcionalidades principales:

- **Votar:** Los usuarios pueden participar en las elecciones disponibles. Una vez autenticados, se les presenta una lista de elecciones en las cuales pueden participar. Al seleccionar una, se les redirige a una vista con los candidatos disponibles y pueden emitir su voto.
- **Crear elección:** Los usuarios tienen la capacidad de crear nuevas elecciones, proporcionando detalles como el nombre, el rango de tiempo durante el cual estará disponible la elección y los candidatos que participarán.
- **Ver resultados:** Los usuarios pueden consultar los resultados. Se les presenta una lista de elecciones y, al seleccionar una, pueden visualizar los resultados detallados.

4.2.2. Flujo de comunicación

Para la explicación de este proceso se hace uso de esta figura 4.1. El usuario comienza accediendo a la app, la cual le redirigirá a "/login"para ser autenticado a través de MetaMask, siempre y cuando no esté autenticado ya. Posteriormente, se accede a un menú principal desde donde se puede elegir entre las tres funcionalidades principales. Dependiendo de la decisión, el usuario es redirigido a diferentes vistas para completar la acción deseada (votar, crear una elección o ver resultados). En el caso de acceder a votar o ver resultados, el usuario elegirá entonces que elección es la que quiere. En caso de una operación exitosa o un error, se proporciona un alerta visual adecuada al usuario, ya sea a través de redirecciones o alertas.

4.2.3. Interacción con los contratos inteligentes

Se utilizan dos contratos inteligentes principales:

- **ElectionFactory.sol:** Este contrato gestiona la creación de nuevas elecciones y proporciona funciones para obtener listas de elecciones, ya sean todas o solo las disponibles.
- **Election.sol:** Una vez creada una elección, este contrato se encarga de las operaciones relacionadas con dicha elección, como votar, obtener candidatos y calcular el ganador.

4.2.4. Mecanismo de transacciones

La aplicación utiliza un enfoque simplificado para gestionar las transacciones con la blockchain. Cuando se realiza una solicitud asíncrona a la blockchain, se utiliza un bloque `catch`. Si ocurre un error durante la transacción o bien en la propia lectura de la blockchain, se muestra una alerta al usuario. En caso contrario, la operación se considera exitosa y el usuario es redirigido al menú principal.

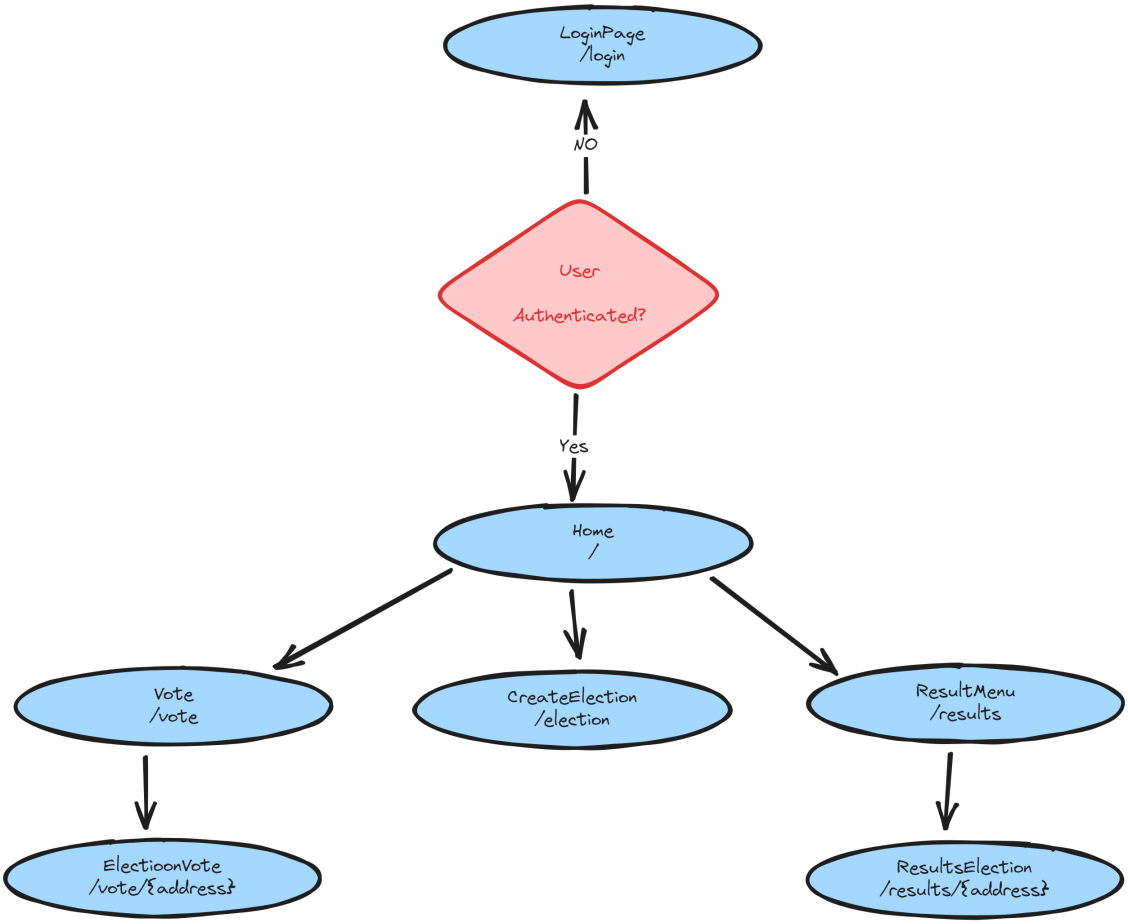

Figura 4.1: Diagrama de flujo de comunicación entre componentes (Nombre componente, URL)

Siempre que el sistema esté en espera de una respuesta proveniente de la blockchain, se presentará una "pantalla de carga" al usuario (Figura 4.2). Esta característica es esencial para mantener informado al usuario sobre el estado actual de la operación, asegurando transparencia y evitando confusiones o incertidumbres.

Figura 4.2: Pantalla de carga.

4.2.5. Autenticación y seguridad

MetaMask sirve como principal mecanismo de autenticación. Aprovechando la naturaleza inmutable de la blockchain, se garantiza que los votos, una vez registrados, no puedan ser alterados.

4.2.6. Rendimiento y escalabilidad

Dado que se trata de un proyecto en local, no se han realizado pruebas de escalabilidad en un entorno de producción. Sin embargo, es importante señalar que el coste de escalabilidad con respecto al número de candidatos es lineal, lo que sugiere un comportamiento predecible a medida que el sistema crece. Por otro lado, en el apartado de un *Posible puesta en producción* 5 se tratará en detalle la escalabilidad.

4.3 Diseño

El diseño de la aplicación de voto electrónico se ha abordado desde tres perspectivas fundamentales para garantizar no solo la funcionalidad, sino también la accesibilidad y la usabilidad para todos los usuarios. Estas tres perspectivas son: la Interfaz de Usuario (UI), que se refiere al aspecto visual y la disposición de los elementos en la aplicación; la Experiencia de Usuario (UX), que trata sobre cómo se siente y cómo interactúa el usuario con la aplicación; y la Responsividad, que asegura que la aplicación sea funcional y estéticamente agradable en diversos dispositivos, desde teléfonos móviles hasta ordenadores de escritorio. A continuación, se detalla el enfoque y las decisiones tomadas en cada uno de estos aspectos.

4.3.1. Interfaz de Usuario (UI)

La interfaz de usuario fue diseñada con un enfoque de simplicidad y facilidad de uso. Se buscó que cada elemento tuviera un propósito claro y que la navegación fuera intuitiva. A continuación, se describen las distintas secciones de la interfaz:

- **Menú:** Es la primera pantalla con la que se encuentra el usuario al acceder a la aplicación (figura 4.3). Desde aquí, se puede dirigir a las distintas funcionalidades que ofrece el sistema. El menú ha sido diseñado de forma clara y concisa, con botones prominentes y etiquetas descriptivas que guían al usuario hacia la acción deseada. También, puede cerrar sesión desde el botón superior derecho. Las opciones están dispuestas de manera lógica, permitiendo una navegación fluida.

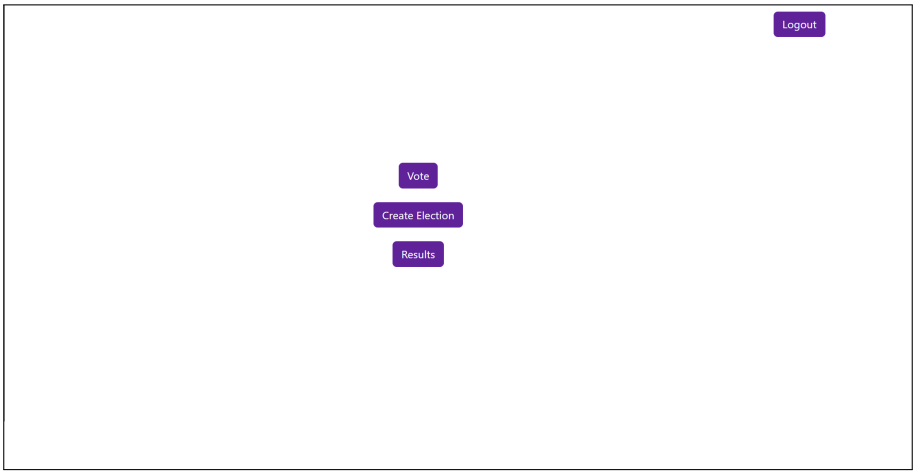

Figura 4.3: Pantalla del Menu Principal

- **Votar:** Esta sección ha sido diseñada para que el proceso de votación sea lo más sencillo y directo posible. Se presenta una lista de elecciones disponibles (figura 4.4), y al seleccionar una, el usuario puede visualizar los candidatos y emitir su voto (figura 4.5). Se ha puesto especial cuidado en garantizar que la selección del candidato y la confirmación del voto sean acciones evidentes y sin ambigüedades (figura4.6). Para lograrlo, se ha decidido minimizar las posibilidades de error del usuario, permitiéndole elegir solo una opción entre los candidatos.

Figura 4.4: Pantalla Elecciones disponibles.

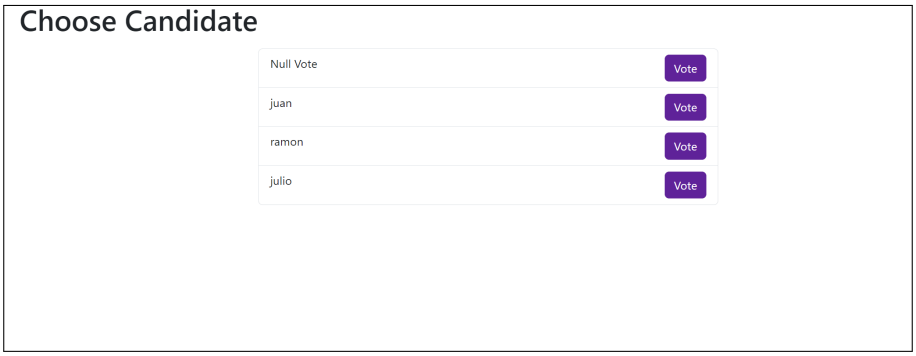

Figura 4.5: Pantalla Votar un candidato o en blanco.

- **Crear Elección:** En esta parte, se proporciona un formulario para la creación de nuevas elecciones (figura 4.7). Se han utilizado campos claramente etiquetados para introducir detalles como el nombre de la elección, el rango de tiempo y los candidatos. Se ha implementado una validación en tiempo real para garantizar que

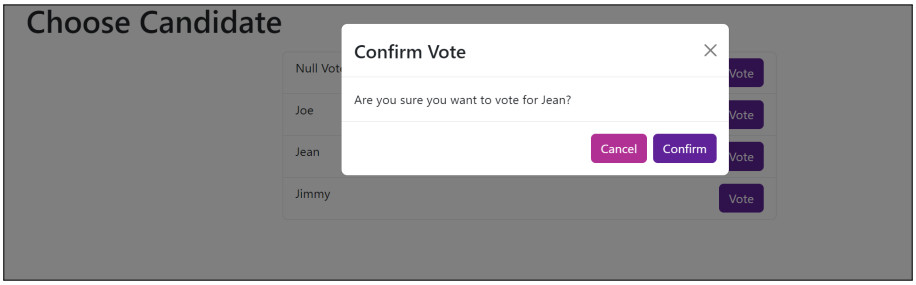

Figura 4.6: Pantalla de confirmación

la información introducida sea correcta y completa antes de proceder. Tanto desde no permitir al usuario introducir caracteres deferentes a los alfabéticos, como un manejador de errores para datos incongruentes como por ejemplo que la fecha de comienzo se anterior al momento actual. Tal y como se muestra en la figura 4.8.

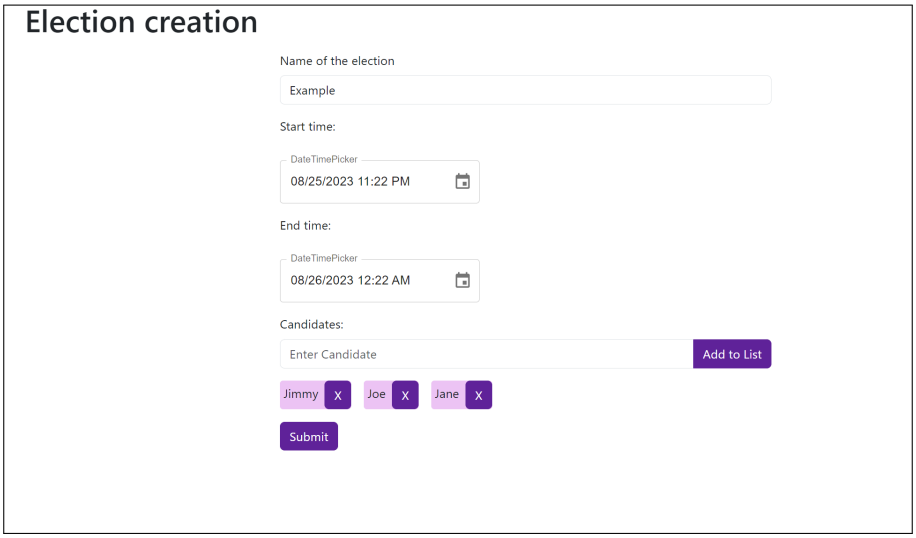

Figura 4.7: Pantalla Creación de elección.

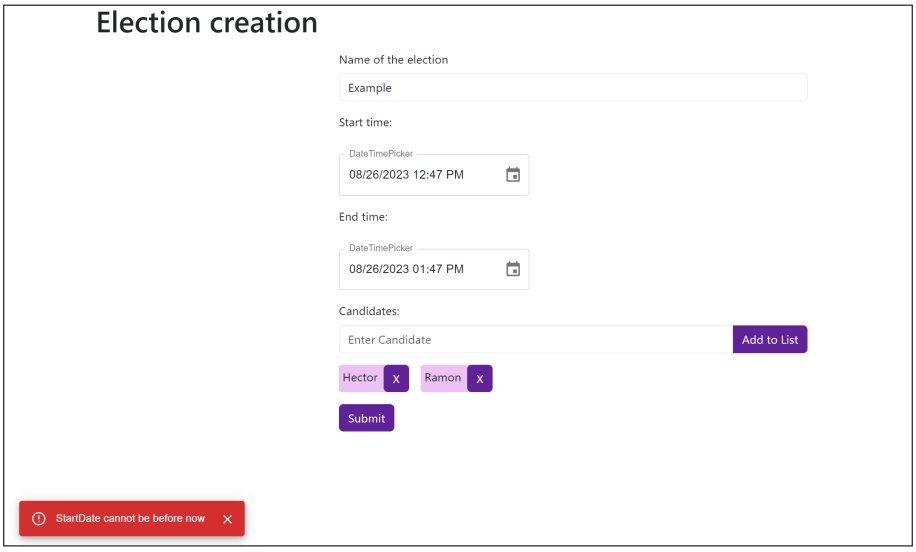

Figura 4.8: Pantalla Creación de elección con error.

- **Ver Resultado:** Esta sección se ha diseñado para ofrecer una visualización clara y comprensible de los resultados de las elecciones. Al seleccionar una elección, se presentan los resultados en forma gráfica dónut y numérica (figura 4.9). Se ha optado por un diseño limpio que evite distracciones y permita al usuario comprender rápidamente el resultado de la votación.

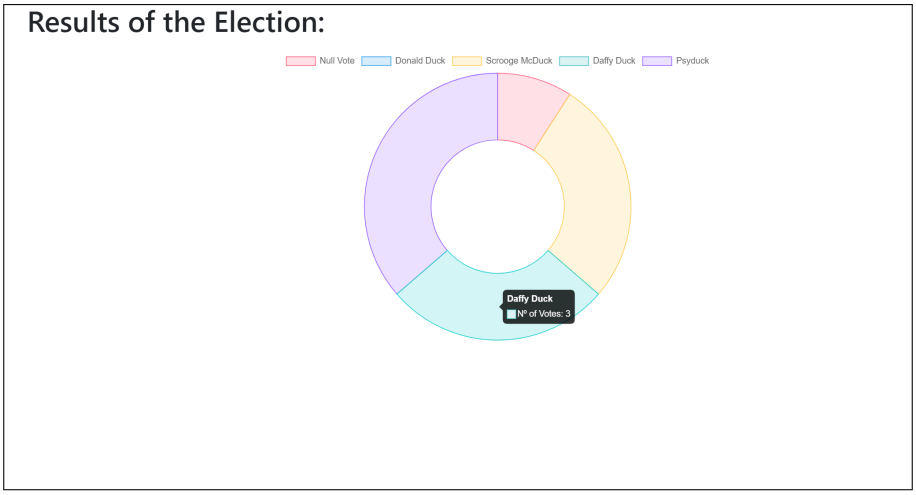

Figura 4.9: Pantalla Resultados.

En todas estas secciones, se ha mantenido una coherencia en términos de diseño, colores y tipografía. Las transiciones entre las distintas partes son suaves, y se ha incluido retroalimentación visual para indicar el estado y resultado de las acciones realizadas por el usuario. Las capturas de pantalla adjuntas proporcionan una visión detallada de cada una de estas secciones y la interacción propuesta.

4.3.2. Experiencia de Usuario (UX)

La experiencia de usuario (UX) es esencial para garantizar que una aplicación no sólo funcione, sino que también sea agradable, intuitiva y fácil de usar. Al diseñar la experiencia de usuario para la aplicación de voto electrónico, se tomaron en cuenta varios aspectos clave para optimizar la interacción del usuario con el sistema.

- **Flujos de Navegación:** Se diseñaron flujos de navegación lógicos y coherentes. Desde el momento en que un usuario ingresa a la aplicación hasta que completa una acción (como votar o ver resultados), se le guía a través de una serie de pasos claros y secuenciales, minimizando las posibilidades de confusión o errores.

- **Facilidad de Uso:** La simplicidad fue una prioridad. Se evitó la sobrecarga de información y se optó por presentar sólo los elementos esenciales en cada pantalla, asegurando que los usuarios puedan realizar sus tareas de manera rápida y sin distracciones innecesarias.

- **Accesibilidad:** Se implementaron prácticas de diseño accesible para garantizar que la aplicación sea usable por todos, independientemente de sus habilidades o limitaciones. Esto incluye tamaños de fuente legibles, contrastes adecuados y la posibilidad de navegar usando solo el teclado.

- **Retroalimentación:** Es crucial que los usuarios reciban retroalimentación después de realizar acciones. Ya sea a través de mensajes de éxito, alertas de error o in-

dicadores visuales, se proporciona información clara sobre los resultados de sus acciones.

- **Consistencia:** Mantener una experiencia uniforme en toda la aplicación es fundamental. Desde la paleta de colores hasta el diseño de los botones y formularios, todo sigue un patrón coherente, lo que facilita la adaptación del usuario al entorno y refuerza la confianza en el sistema.

La meta principal en el diseño de la UX fue que los usuarios se sintieran confiados y seguros al interactuar con la aplicación, comprendiendo cada paso del proceso y teniendo la certeza de que su voto se registró correctamente.

4.3.3. Responsividad

En la era actual de la tecnología, es esencial que las aplicaciones sean adaptables a una amplia variedad de dispositivos para garantizar su accesibilidad y usabilidad. La responsividad se refiere a la capacidad de una aplicación para ajustarse y funcionar de manera óptima en distintos tamaños y orientaciones de pantalla, desde smartphones y tablets hasta ordenadores de escritorio.

- **Adaptabilidad:** La aplicación ha sido diseñada con un enfoque de polivalencia buscando ser funcional en cualquier tipo de pantalla, teniendo en cuenta que hay un gran numero de usuarios que pueden usar smartphones o dispositivos con pantallas pequeñas.
- **React-Bootstrap:** Para facilitar la tarea de crear una interfaz responsiva, se utilizó React-Bootstrap, una biblioteca que proporciona componentes de diseño adaptativos para React. Con React-Bootstrap, se pueden crear interfaces que se reajustan y reorganizan automáticamente según el tamaño de la pantalla del dispositivo. Detalles más específicos sobre React-Bootstrap se explicarán en el siguiente apartado 4.4.1.
- **CSS Personalizado:** Aunque React-Bootstrap proporciona una base sólida para la responsividad, se utilizaron hojas de estilo en cascada (CSS) personalizadas, e concreto SCSS (Sassy CSS), para refinar aún más la experiencia de usuario y garantizar que cada elemento de la interfaz se vea y funcione de manera óptima en todos los dispositivos.
- **Pruebas en Diferentes Dispositivos:** Para asegurar la responsividad, la aplicación fue probada en una variedad de dispositivos y navegadores, identificando y corrigiendo cualquier inconsistencia o problema que pudiera surgir.

La responsividad es más que un simple ajuste de diseño; es un principio fundamental que garantiza que todos los usuarios, independientemente del dispositivo que utilicen, tengan acceso a la aplicación y puedan usarla de manera efectiva.

4.4 Tecnología Frontend

4.4.1. Reactjs

React Facebook, 2023 fue elegido como la biblioteca principal para construir la interfaz de usuario debido a su eficiencia, escalabilidad y el concepto de componentes reutilizables. React utiliza un algoritmo que optimiza la actualización del DOM, garantizando

que solo se realicen los cambios necesarios. Esta eficiencia es crucial para aplicaciones interactivas que requieren una alta reactividad y una experiencia de usuario fluida.

La naturaleza modular de React y su sistema basado en componentes favorecen una organización de código más eficiente, facilitando así la escalabilidad y el mantenimiento del proyecto. La adopción de estos módulos en el proyecto se justifica por la confiabilidad que ofrecen las bibliotecas adecuadas, ya que proporcionan código ampliamente probado y verificado en comparación con el desarrollo desde cero. A continuación, se detallan los módulos más relevantes:

React Router

Dentro del proyecto, React Router Jackson y Florence, 2021 desempeña un papel crucial en la navegación entre diferentes vistas o componentes. Por ejemplo, una vez que un usuario se autentica a través de MetaMask, React Router redirige al usuario a la pantalla principal o a la vista específica que el usuario estaba tratando de acceder antes de la autenticación.

React-Bootstrap y Material-UI

Para la estilización y diseño de la interfaz de usuario, se empleó React-Bootstrap R.-B. Team, 2015 y Material-UI M.-U. Team, 2021. React-Bootstrap es una reinvención de la biblioteca Bootstrap para React, ofreciendo componentes estilizados y optimizados para esta biblioteca. En este proyecto, React-Bootstrap se ha utilizado en toda la aplicación, otorgando un diseño consistente y adaptativo.

Por otro lado, Material-UI, inspirado en el diseño Material de Google, se ha utilizado específicamente para la creación de formularios en la aplicación, proporcionando componentes visuales atractivos y fáciles de usar para la recopilación de datos de los usuarios.

Ethers.js

Ethers.js Moore, 2015 es la biblioteca que media entre la aplicación React y la blockchain de Ethereum. Cada vez que la aplicación necesita leer datos de la cadena de bloques o enviar una transacción, se hace a través de Ethers.js. Esta biblioteca facilita la conexión con la red Ethereum, la gestión de transacciones y la interacción con contratos. inteligentes.

En este proyecto Ethers.js se ha usado principalmente para comunicarse con los dos contratos inteligentes que implementa TAVS, **Election.sol** y **ElectionFactory.sol**. Haciendo uso de Metamask, Ethers es capaz de llamar a los métodos públicos del Smart Contract, un ejemplo de ello seria la función `getCandidates()` del contrato **Election.sol**:

```
const provider = new ethers.providers.Web3Provider(window.ethereum);
const erc20 = new ethers.Contract(address,erc20ABI, provider);
return await erc20.functions.getCandidates();
```

4.4.2. Typescript

TypeScript Microsoft, 2021 es un lenguaje de programación de código abierto desarrollado y mantenido por Microsoft. Es un conjunto estricto de JavaScript, que agrega tipado estático opcional y otras características a la sintaxis del lenguaje. Gracias a este tipado estático, TypeScript permite a los desarrolladores detectar errores de tipo en

tiempo de compilación, mejorando la robustez y calidad del código. En el contexto de la aplicación, TypeScript se ha utilizado en algunas partes específicas, y solo en el archivo `utils.ts`, principalmente para el formateo de datos antes de enviarlos a los contratos inteligentes. Reutilizando en gran medida funciones del la implementación en Solidity de TAVS Larriba y López, 2023. Como por ejemplo esta función:

```
function packAsNbytes(hexStr: string, n = 32) {
    if (hexStr.substring(0, 2) === "0x") {
      hexStr = hexStr.substring(2);
    }
    const words = Math.ceil(hexStr.length / (n * 2));
    const targetLength = words * (n * 2);
    const zeroes = "0".repeat(targetLength - hexStr.length);
    return "0x" + zeroes + hexStr;
}
```

Al proporcionar una estructura de tipo clara y coherente, TypeScript garantiza que los datos enviados a la blockchain sean consistentes y estén formateados correctamente.

4.4.3. SCSS

SCSS (Sassy CSS) Weizenbaum y Eppstein, 2021 es una extensión del CSS que permite el uso de variables, anidación y otras características avanzadas que no están disponibles en CSS plano. En el proyecto, se ha utilizado SCSS para definir estilos personalizados, especialmente para configurar colores, como el distintivo color morado que predomina en la aplicación. Gracias a las capacidades de SCSS, mantener y actualizar los estilos de la aplicación se vuelve más manejable y modular.

4.4.4. Autenticación con MetaMask

En el universo de las aplicaciones descentralizadas (dApps) construidas sobre la blockchain de Ethereum, MetaMask ConsenSys, 2016 se presenta como un puente crucial entre las aplicaciones web tradicionales y la blockchain de Ethereum. A través de MetaMask, los usuarios pueden interactuar con contratos inteligentes y aplicaciones descentralizadas directamente desde sus navegadores sin la necesidad de comprender completamente la complejidad subyacente, siendo así totalmente transparente para el usuario.

Autenticación Descentralizada

A diferencia de la autenticación tradicional basada en correo electrónico y contraseña, las dApps a menudo utilizan direcciones Ethereum como identidades de usuario. En este escenario, en lugar de pedir a los usuarios que se registren o inicien sesión, simplemente verificamos y usamos su dirección Ethereum. Es aquí donde MetaMask juega un papel esencial, permitiendo a los usuarios autenticarse utilizando su dirección Ethereum.

Flujo de la Autenticación

1. **Inicialización y Creación de Contexto**:

   En React, un contexto permite compartir valores entre componentes sin tener que pasar explícitamente un prop a través de cada nivel del árbol de componentes. Para este caso, se ha creado un contexto (`MetaMaskContext`) para gestionar y proveer el estado de autenticación del usuario a lo largo de la aplicación.

```
const MetaMaskContext = React.createContext(false);
```

2. **Iniciando Sesión con MetaMask**:

   En el componente `LoginPage`, se establece una conexión con MetaMask y se solicita al usuario habilitar y conectar su cuenta Ethereum con la aplicación.

```
// Establish a connection with MetaMask
const provider = new ethers.providers.Web3Provider(window.ethereum);
// Request user to enable their Ethereum account
await window.ethereum.enable();
// The context updates automatically post this
```

3. **Ruta Privada**:

   El componente `PrivateRoute` actúa como un guardián. Comprueba el estado de autenticación del usuario consultando el valor del contexto. Si el usuario está autenticado, renderiza el componente deseado; de lo contrario, redirige al usuario a la página de inicio de sesión.

```
const isLoggedIn = useMetaMask();
return isLoggedIn ? props.element : <Navigate to="/login"/>
```

4. **Cierre de Sesión**:

   Para cerrar sesión, simplemente se reinicia la dirección Ethereum en el objeto `window.ethereum`.

```
window.ethereum.selectedAddress = null;
```

En definitiva haciendo uso de MetaMask para la autenticación, la aplicación proporciona una forma descentralizada para que los usuarios inicien sesión utilizando su dirección Ethereum. Esta metodología asegura que los usuarios tengan control total sobre sus identidades sin depender de servidores o bases de datos centralizados. La utilización de contextos en React simplifica aún más el proceso al proveer a los componentes con la información necesaria para ajustar dinámicamente su comportamiento en función del estado de autenticación del usuario.

4.4.5. Comunicación entre las tecnologías

En la arquitectura de la aplicación, el flujo de comunicación y la interconexión entre tecnologías juegan un papel crucial. El diagrama 4.10 ilustra cómo estas tecnologías se entrelazan y colaboran para ofrecer una solución robusta y eficiente.

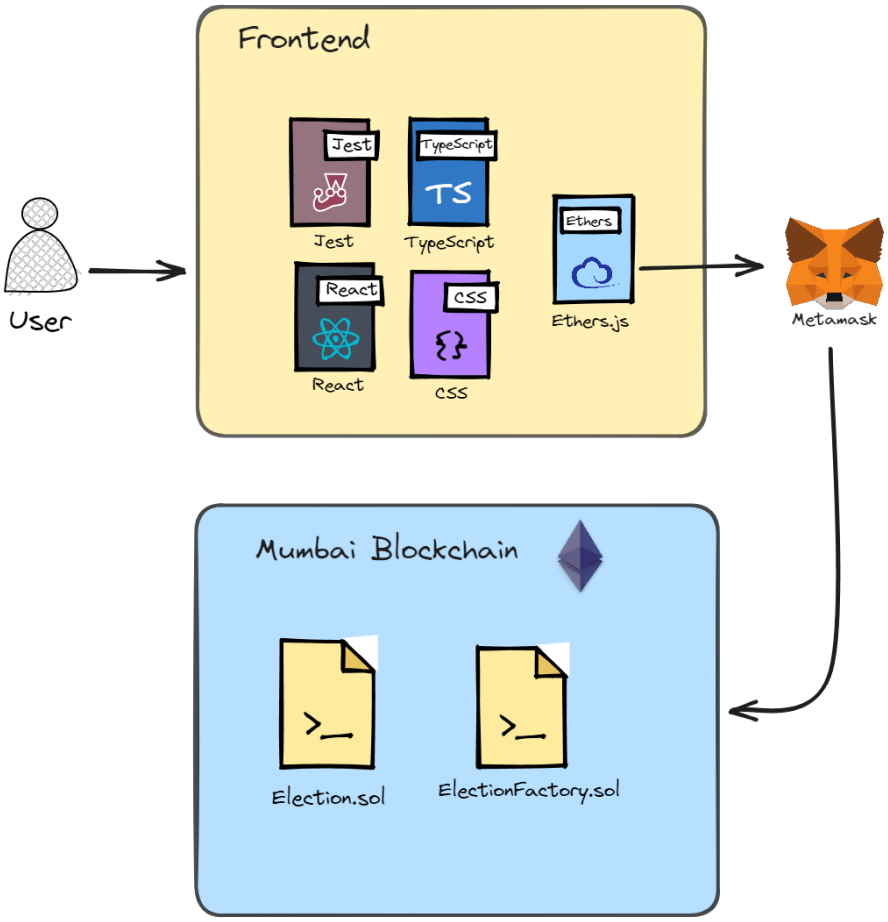

Figura 4.10: Diagrama de comunicación entre las tecnologías utilizadas

El corazón del sistema es el **frontend**, que ha sido desarrollado utilizando una combinación de tecnologías como React, SCSS y TypeScript, ya mencionadas anteriormente. Estas tecnologías trabajan en conjunto para garantizar una experiencia de usuario fluida e interactiva, además de proporcionar un código limpio y mantenible.

Para las pruebas y aseguramiento de la calidad, se utiliza **Jest**, el cual se detallará en el siguiente apartado 4.5, lo que garantiza que cada componente de la aplicación funcione como se espera y que cualquier regresión sea detectada rápidamente durante el proceso de desarrollo.

El puente entre el frontend y la blockchain es **ethers.js**, una biblioteca que facilita la comunicación directa con la blockchain de Ethereum. A través de ethers.js, la aplicación puede interactuar con los contratos inteligentes, realizar transacciones y consultar datos.

MetaMask actúa como la puerta de entrada para los usuarios a la blockchain. No solo proporciona autenticación, sino que también permite a los usuarios aprobar transacciones, lo que es esencial para operaciones como votar en una elección.

En la blockchain, encontramos los contratos inteligentes **Election.sol** y **ElectionFactory.sol**, desplegados en la blockcahin de Polygon, y en concreto la red Mumbai, siendo esta un red de pruebas. Estos contratos son la base del proceso de votación, encargándose de aspectos como la creación de elecciones, el registro de votos y la computación de resultados.

El atractivo de esta arquitectura radica en cómo estas tecnologías, aunque diversas, trabajan conjuntamente para permitir un proceso de votación transparente, seguro y descentralizado. La comunicación entre el frontend y la blockchain, facilitada por herramien-

tas como ethers.js y MetaMask, garantiza que los usuarios puedan interactuar con el sistema de manera intuitiva, mientras que en el backend, la blockchain asegura la integridad y la inmutabilidad de cada voto.

4.5 Pruebas y Verificación

En cualquier desarrollo de software, pero especialmente en contextos críticos como el voto electrónico, la verificación y las pruebas son esenciales para garantizar que la aplicación funcione según lo previsto y sea segura.

4.5.1. Importancia de las pruebas

En el ámbito del voto electrónico, la confianza en el sistema es fundamental. Si los votantes o las partes interesadas no confían en que el sistema registrará y contará correctamente los votos, o si temen que pueda ser vulnerable a ataques o manipulaciones, el sistema no será adoptado. Por ello, probar y verificar la solución es crucial para garantizar su integridad, seguridad y fiabilidad.

4.5.2. Enfoque de pruebas

Se adoptó un enfoque de pruebas multifacético para garantizar una cobertura exhaustiva del sistema:

- **Pruebas Unitarias:** Estas pruebas se centran en pequeñas unidades de código para asegurarse de que funcionen según lo previsto. Componentes críticos, como el formulario “CreateElection.js”, la página de login “LoginPage” y el componente de comprobación de autenticación “PrivateRoute.js”, fueron sometidos a pruebas unitarias para garantizar su correcto funcionamiento. Algunas de esas pruebas,por ejemplo para “CreateElection.js” :

```
1  it("should not call the 'handleSubmit' function if the 'endTime' field
       is before the 'startTime' field", () => {
2    const handleSubmitSpy = jest.fn();
3    render(<CreateElection handleSubmit={handleSubmitSpy} />);
4    const startTimeInput = screen.getByTestId("create-election-startTime"
       );
5    const submitButton = screen.getByTestId("create-election-submit");
6    startTimeInput.value = now.toISOString();
7    const endTimeInput = screen.getByTestId("create-election-endDate");
8    endTimeInput.value = now.toISOString() - "1m";
9    fireEvent.click(submitButton);
10   expect(handleSubmitSpy).not.toBeCalled();
11 });
12
13 it("should not call the 'handleSubmit' function if the 'candidates'
       list is empty", () => {
14   const handleSubmitSpy = jest.fn();
15   render(<CreateElection handleSubmit={handleSubmitSpy} />);
16   const submitButton = screen.getByTestId("create-election-submit");
17   const candidatesInput = screen.getByTestId("create-election-
       candidates");
18   candidatesInput.value = "";
19   fireEvent.click(submitButton);
20   expect(handleSubmitSpy).not.toBeCalled();
21 });
```

- **Pruebas Manuales:** Además de las pruebas automatizadas, se realizaron pruebas manuales para simular la interacción real del usuario con la aplicación y identificar posibles fallos o mejoras desde una perspectiva de usuario real.

4.5.3. Herramientas y tecnologías

Para llevar a cabo las pruebas y verificaciones, se optó por herramientas especializadas que permiten una evaluación profunda y precisa del código. A continuación, se detallan estas herramientas y su aplicación en el proyecto:

- **Jest:** Jest Inc., 2021 es una biblioteca de pruebas de JavaScript desarrollada por Facebook, conocida por su rapidez y flexibilidad. Es especialmente útil para probar aplicaciones React, como la desarrollada en este proyecto. Algunas características y usos de Jest en este proyecto incluyen:

  - *Instantáneas (Snapshots):* Esta funcionalidad permite capturar el estado de un componente y garantizar que no cambie inesperadamente con futuras modificaciones.

  - *Mocking:* Jest proporciona potentes capacidades de simulación que fueron esenciales para aislar componentes y realizar pruebas unitarias en componentes como los mencionados.

  - *Cobertura de Código:* Con Jest, se generaron informes de cobertura de código para asegurar que la mayoría del código fuente estuviera sometido a pruebas.

- **SonarQube:** Se trata de una plataforma de inspección continua de calidad del código que examina el código fuente para detectar problemas potenciales SonarSource, 2021. En el contexto de este proyecto, SonarQube se utilizó para:

  - *Detección de Bugs y Vulnerabilidades:* SonarQube analiza el código en busca de patrones conocidos que pueden indicar bugs o vulnerabilidades de seguridad.

  - *Mantenibilidad:* A través de su análisis, SonarQube identifica "code smell" o patrones de código que pueden hacer que el software sea más difícil de mantener a largo plazo.

  Tras ejecutar el anális de Sonarqube los resultados obtenidos fueron bastante positivos, no se encontró ningún bug, solo "code smells" en concreto 27 como se muestra en la figura 4.11.

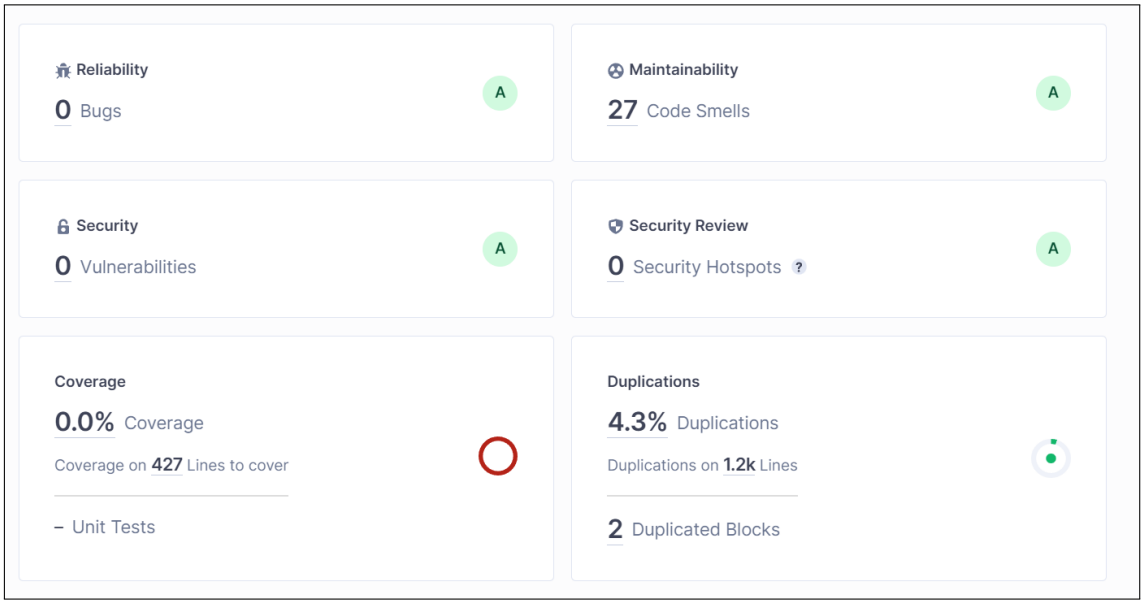

Figura 4.11: Resultados tras análisis SonarQube.

Muchos de estos "code smells" se tratan de repeticiones de imports, variables sin asignación, repetición de variables, uso de palabra primitiva `var` en vez de `const`... Es decir nada critico, se han tenido en cuenta todos ellos, y en la mayoría de casos se han solventando exceptuando algunos casos que eran falsos positivos y arreglarlos afectaba a la funcionalidad total de la aplicación. Alguno ejemplos de estos "code smells" son los siguientes:

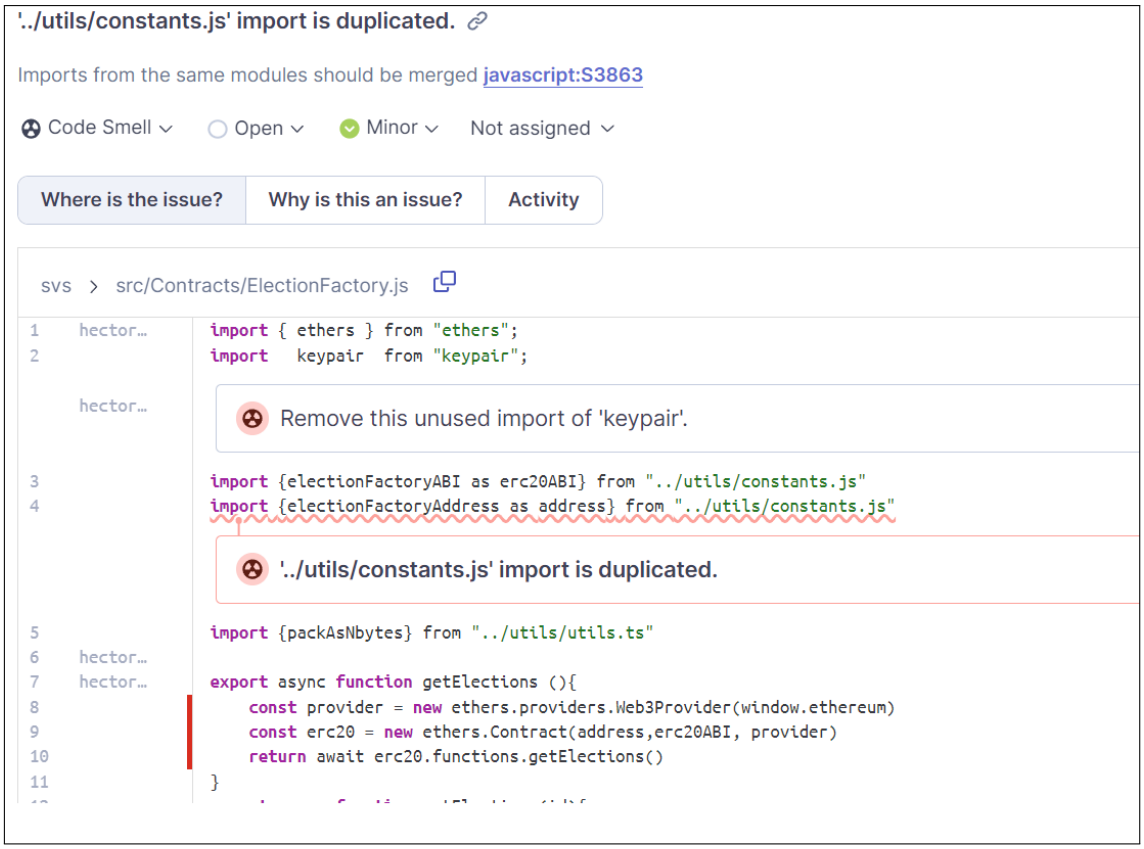

Figura 4.12: "Code smells": Imports repetidos.

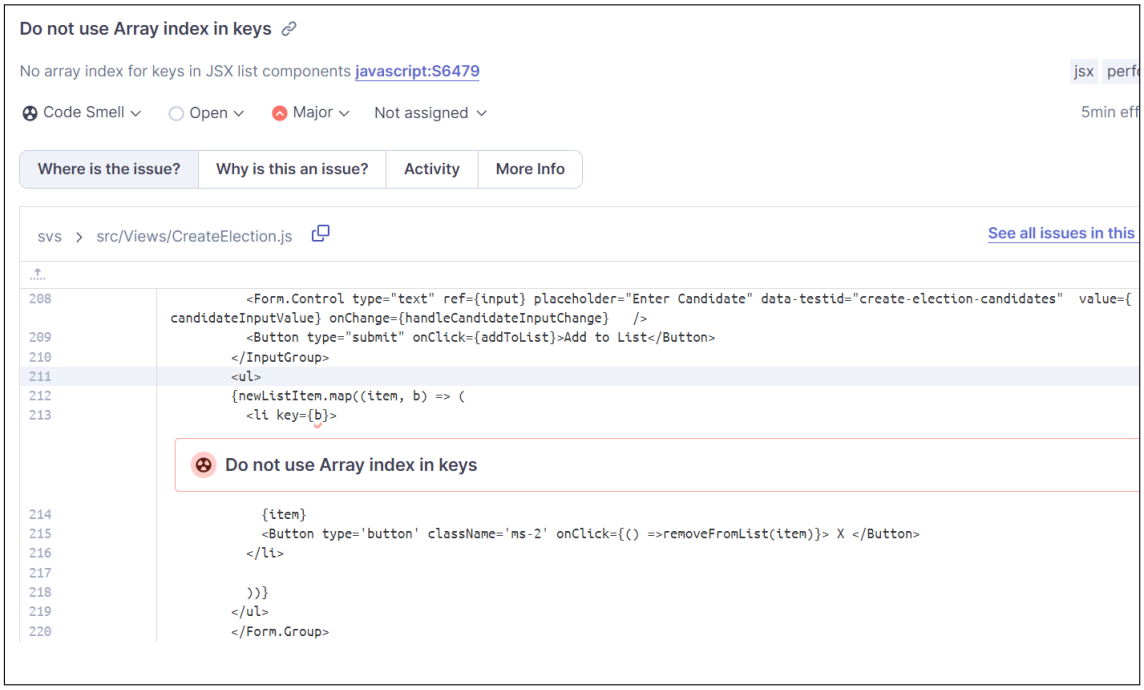

Figura 4.13: "Code smells": Asignación de índice de array a atributo key.

Además de estas herramientas, se implementaron diversos filtros en el formulario y se realizaron numerosas pruebas manuales para asegurar la robustez y seguridad de la aplicación.

4.5.4. Resultados

La ejecución de las pruebas reveló que la aplicación es robusta y funciona según lo previsto en la mayoría de los escenarios. Los pocos problemas o desafíos encontrados fueron abordados y corregidos a tiempo, garantizando así la calidad y fiabilidad del sistema de la web 3.0. Las pruebas manuales, en particular, proporcionaron valiosos conocimientos sobre la experiencia del usuario y permitieron realizar ajustes para mejorar la usabilidad y eficiencia de la aplicación.

CAPÍTULO 5
Posible puesta en producción

Cabe mencionar esta aproximación de propuesta final esta enfocado a un despliegue de la aplicación totalmente integrado con la segunda autoridad del sistema TAVS, IA; a pesar que este proyecto la tiene en cuenta, tal y como se mencionó en la *solución propuesta* 4.1.2.

A continuación se describen recomendaciones y precauciones a la hora de llevar a cabo la puesta en producción:

5.1 Despliegue en la Red Principal de Polygon

5.1.1. Auditoría de Smart Contracts

Antes de cualquier despliegue en la red principal, es fundamental realizar una auditoría exhaustiva de los smart contracts para garantizar la seguridad y la integridad del sistema. La auditoría debe ser realizada por expertos en seguridad de blockchain para identificar y corregir posibles vulnerabilidades.

5.1.2. Optimización del Gas

El gas es un factor crucial en las transacciones de blockchain. Es esencial optimizar el consumo de gas de los smart contracts para garantizar transacciones económicas y eficientes. Herramientas como Solc (Solidity Compiler) pueden ser útiles para este propósito.

5.1.3. Despliegue de los Contratos

Una vez auditados y optimizados, los contratos están listos para ser desplegados en la red principal de Polygon. Se recomienda usar herramientas como Truffle o Hardhat(usado en (Larriba y López, 2023)) para facilitar el proceso.

5.2 Configuración del Frontend

5.2.1. Selección de Hosting

Para una aplicación nacional de voto electrónico, es vital garantizar alta disponibilidad y resistencia a fallos. Se recomienda utilizar servicios de alojamiento escalables y

confiables como AWS, Google Cloud o Azure. Estos servicios ofrecen soluciones específicas para aplicaciones web que requieren alta disponibilidad.

5.2.2. Integración con la Blockchain

El frontend debe estar configurado para interactuar con la red Polygon. Esto implica configurar correctamente un proveedor de web3 y asegurarse de que se conecte a los nodos adecuados de la red Polygon.

5.2.3. Dominio y Certificado SSL

Para reforzar la confianza de los usuarios y garantizar la seguridad, se debe adquirir un dominio personalizado y configurar un certificado SSL. Esto asegura que las comunicaciones entre el usuario y el servidor estén cifradas.

5.3 Pruebas y Control de Calidad

5.3.1. Entorno de Pruebas

Antes de lanzar la aplicación al público, es esencial establecer un entorno de pruebas que imite el entorno de producción. Esto permite realizar pruebas de carga, pruebas de seguridad y otras pruebas esenciales sin afectar a los usuarios reales.

5.3.2. Pruebas Beta

Realizar pruebas con un grupo seleccionado de usuarios puede ofrecer información valiosa sobre posibles problemas o mejoras necesarias antes del lanzamiento oficial.

5.3.3. Pruebas de Seguridad

Hacer pruebas de seguridad y auditorías de todo el sistema ahorrará problemas futuros, además aportando más capas de seguridad y robustez al proyecto. A pesar de que la seguridad es un campo caro siempre es una buena inversión y más tratándose de un sistema de voto electónico.

5.4 Lanzamiento y Post-Lanzamiento

5.4.1. Estrategia de Lanzamiento

Es vital tener una estrategia de lanzamiento bien planificada. Esto puede incluir anuncios en medios nacionales, capacitación para los usuarios y soporte técnico durante el periodo inicial. Al igual, que seria conveniente dar a los ciudadanos tutoriales de como efectuar seu derecho a voto.

5.4.2. Monitoreo y Soporte

Una vez que la aplicación esté en funcionamiento, es crucial llevar a cabo un seguimiento constante para identificar y solucionar cualquier problema que pueda surgir. Este

monitoreo no solo garantiza que la aplicación funcione sin problemas, sino que también proporciona información valiosa sobre el comportamiento del usuario, lo que puede ser útil para futuras actualizaciones o mejoras. Además, ofrecer soporte técnico a los usuarios es esencial para abordar sus inquietudes y garantizar una experiencia de usuario satisfactoria.

5.4.3. Actualizaciones y Mantenimiento

La tecnología y las necesidades de los usuarios evolucionan con el tiempo. Es crucial mantener la plataforma actualizada y realizar mejoras según las necesidades de los usuarios y los avances tecnológicos.

CAPÍTULO 6
Conclusión y trabajo futuro

A lo largo de este proyecto, se ha llevado a cabo una exhaustiva investigación y desarrollo en el ámbito del voto electrónico, específicamente utilizando el Sistema de Votación Anónimo Transparente (TAVS) como base conceptual. La implementación de este sistema en la red de Polygon y la creación de smart contracts en Solidity han sido esenciales para lograr un mecanismo de votación robusto y descentralizado.

La elección de React como tecnología principal para el desarrollo del frontend ha demostrado ser acertada. Gracias a sus características de eficiencia, escalabilidad y adaptabilidad, ha sido posible diseñar una interfaz intuitiva y responsiva. La integración de React-Bootstrap y Material-UI ha enriquecido la experiencia del usuario, permitiendo una navegación fluida y una interacción agradable con la plataforma.

Uno de los logros más destacados ha sido la capacidad de garantizar la transparencia y la integridad de los votos emitidos. A través de la tecnología blockchain, se ha logrado un registro inmutable y verificable de cada voto, eliminando las posibilidades de fraude y manipulación. Además, se ha dado especial énfasis en proteger la privacidad de los votantes, garantizando su anonimato en todo el proceso.

Sin embargo, a pesar de estos logros, es esencial reconocer que aún hay aspectos del sistema que requieren mayor desarrollo y refinamiento. La implementación actual, aunque funcional, representa una primera etapa en la construcción de un sistema de votación plenamente integrado y seguro.

6.1 Acceso al Proyecto

El desarrollo y la implementación de este proyecto no serían posibles sin el uso de herramientas y plataformas modernas que facilitan la colaboración y el acceso al código fuente. Todo el trabajo realizado, desde la implementación en Solidity del TAVS hasta el frontend desarrollado en React, está disponible de manera abierta en GitHub (González, 2023). Esto no solo asegura la transparencia del proyecto, sino que también permite a otros desarrolladores y entusiastas acceder, revisar y contribuir al código. La decisión de hacer de este un proyecto de código abierto refuerza nuestro compromiso con la transparencia, la colaboración y la innovación continua. Invitamos a todos los interesados a explorar el repositorio, aportar sus conocimientos y participar activamente en el perfeccionamiento y expansión de este sistema de votación.

6.2 Proyectos Futuros

El siguiente paso lógico y esencial en la evolución de este proyecto es la incorporación de una Autoridad de Identificación (IA). Esta entidad se encargará de verificar la identidad de los votantes y su pertenencia al censo electoral, añadiendo una capa adicional de seguridad y confianza al sistema. Al completar la visión original del TAVS con la IA, se estará más cerca de un sistema de votación totalmente transparente y a prueba de manipulaciones.

Una aproximación añadida a considerar es la realizar la interfaz más interactiva que permita a los votantes acceder a información relevante sobre los candidatos, propuestas o iniciativas en las que están votando. Integrando recursos educativos y formativos, se podría facilitar a los ciudadanos la toma de decisiones informadas en el proceso electoral.

Adicionalmente, la expansión geográfica del sistema es otra área de oportunidad. Adaptando el sistema a las regulaciones y necesidades específicas de diferentes regiones o países, ponerla en distintos idiomas; podría abrir puertas a una adopción más amplia de esta tecnología en elecciones a nivel mundial.

6.3 Palabras Finales

En resumen, este proyecto representa un paso significativo hacia la modernización y digitalización del proceso electoral. Aunque todavía hay desafíos por enfrentar, estamos convencidos de que, con esfuerzo y dedicación, es posible crear un sistema de voto electrónico que sea seguro, transparente y accesible para todos. La visión de un futuro donde la tecnología y la democracia van de la mano es, sin duda, una perspectiva emocionante y llena de posibilidades.

Bibliografía

Álvarez, R. Michael y Thad E. Hall (2008). *Electronic Elections: The Perils and Promises of Digital Democracy*. Princeton University Press. ISBN: 9780691146225. URL: http://www.jstor.org/stable/j.ctt7ss68 (visitado 07-08-2023).

Benaloh, Josh Daniel Cohen (1987). *Verifiable secret-ballot elections*, págs. 67-71. DOI: info:doi/.

Bernhard, Matthew y col. (2017). *Public Evidence from Secret Ballots*. arXiv: 1707.08619 [cs.CR]. URL: https://arxiv.org/pdf/1707.08619.pdf.

Burton, Craig, Chris Culnane y Steve Schneider (2015). *Secure and Verifiable Electronic Voting in Practice: the use of vVote in the Victorian State Election*. arXiv: 1504.07098 [cs.CR].

Buterin, Vitalik (2013). *Ethereum White Paper: A Next-Generation Smart Contract and Decentralized Application Platform*. URL: https://ethereum.org/en/whitepaper/.

Chaum, David (1983). «Blind Signatures for Untraceable Payments». En: *Advances in Cryptology*. Ed. por David Chaum, Ronald L. Rivest y Alan T. Sherman. Boston, MA: Springer US, págs. 199-203.

Chaum, David L. (feb. de 1981). «Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms». En: *Commun. ACM* 24.2, págs. 84-90. ISSN: 0001-0782. DOI: 10.1145/358549.358563. URL: https://doi.org/10.1145/358549.358563.

ConsenSys (2016). *MetaMask: A crypto wallet and gateway to blockchain apps*. https://metamask.io/.

Eijk, D. y col. (2018). «Electronic Voting for All: Co-creating an Accessible Interface». En: DOI: 10.1007/978-3-319-96071-5_84. URL: https://dx.doi.org/10.1007/978-3-319-96071-5_84.

Everett, Sarah P. y col. (2008). «Electronic Voting Machines versus Traditional Methods: Improved Preference, Similar Performance». En: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '08. Florence, Italy: Association for Computing Machinery, págs. 883-892. ISBN: 9781605580111. DOI: 10.1145/1357054.1357195. URL: https://doi.org/10.1145/1357054.1357195.

Facebook (2023). *React – A JavaScript library for building user interfaces*. https://react.dev/.

González, Hector (2023). *Interfaz web 3.0 para TAVS*. Repositorio GitHub. URL: https://github.com/hectorgonza/svs.git.

Heiberg, Sven, Arnis Parsovs y Jan Willemson (2015). *Log Analysis of Estonian Internet Voting 2013–2015*. Cryptology ePrint Archive, Paper 2015/1211. https://eprint.iacr.org/2015/1211. DOI: 10.1007/978-3-319-22270-7_2. URL: https://eprint.iacr.org/2015/1211.

Inc., Facebook (2021). *Jest: Delightful JavaScript Testing*. URL: https://jestjs.io/.

Interior, Ministerio del (2023). *Dosier de prensa de las elecciones generales de España del 23 de Julio del 2023*. URL: https://www.interior.gob.es/opencms/export/sites/default/.galleries/galeria-de-prensa/documentos-y-multimedia/noticias/documentos/2023/07_Julio/DOSIER-23J.pdf (visitado 23-07-2023).

Jackson, Michael y Ryan Florence (2021). *React Router: Declarative routing for React*. Ver. 6. URL: https://reactrouter.com/.

Jafar, Uzma, Mohd Juzaidin Ab Aziz y Z. Shukur (2021). «Blockchain for Electronic Voting System—Review and Open Research Challenges». En: *Sensors* 21.17, pág. 5874. DOI: 10.3390/s21175874. URL: https://dx.doi.org/10.3390/s21175874.

Jarvis, Sharon E. y Soo-Hye Han (2018). *How Journalists Sideline Electoral Participation (Without Even Knowing It)*. University Park, USA: Penn State University Press. ISBN: 9780271082905. DOI: doi:10.1515/9780271082905. URL: https://doi.org/10.1515/9780271082905.

Kho, Yun-Xing, Swee-Huay Heng y Ji-Jian Chin (2022). «A Review of Cryptographic Electronic Voting». En: *Symmetry* 14.5, pág. 858. DOI: 10.3390/sym14050858. URL: https://www.mdpi.com/2073-8994/14/5/858/pdf?version=1650535235.

Kiayias, Aggelos y col. (2017). «Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol». En: *Advances in Cryptology – CRYPTO 2017*. Ed. por Jonathan Katz y Hovav Shacham. Cham: Springer International Publishing, págs. 357-388.

Larriba, Antonio M. y Damián López (2023). «A Solidity implementation of TAVS». En: *Frontiers in Blockchain* 6. ISSN: 2624-7852. DOI: 10.3389/fbloc.2023.1105119. URL: https://www.frontiersin.org/articles/10.3389/fbloc.2023.1105119.

Larriba, Antonio M., José M. Sempere y Damián López (2020). «A two authorities electronic vote scheme». En: *Computers and Security* 97, pág. 101940. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2020.101940. URL: https://www.sciencedirect.com/science/article/pii/S0167404820302169.

Menezes, Alfred, Paul Van Oorschot y Scott Vanstone (1996). *Handbook of applied cryptography*. CRC press.

Microsoft (2021). *TypeScript*. https://www.typescriptlang.org/. Version 4.3.

Moore, Richard (2015). *Ethers.js: Complete Ethereum wallet implementation and utilities in JavaScript and TypeScript*. https://docs.ethers.io/.

Nakamoto, Satoshi (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. URL: https://bitcoin.org/bitcoin.pdf.

Noizat, Pierre (2015). «Chapter 22 - Blockchain Electronic Vote». En: ed. por David Lee Kuo Chuen, págs. 453-461. DOI: https://doi.org/10.1016/B978-0-12-802117-0.00022-9. URL: https://www.sciencedirect.com/science/article/pii/B9780128021170000229.

Pereira, Bruno Miguel Batista y col. (2023). «Blockchain-Based Electronic Voting: A Secure and Transparent Solution». En: *Cryptography* 7.2, pág. 27. DOI: 10.3390/cryptography7020027. URL: https://dx.doi.org/10.3390/cryptography7020027.

Rivest, Ronald L, Adi Shamir y Leonard Adleman (1978). «A method for obtaining digital signatures and public-key cryptosystems». En: *Communications of the ACM* 21.2, págs. 120-126.

SonarSource (2021). *SonarQube: Continuous Code Quality*. URL: https://www.sonarqube.org/.

Springall, Drew y col. (2014). «Security Analysis of the Estonian Internet Voting System». En: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. CCS '14. Scottsdale, Arizona, USA: Association for Computing Machinery, págs. 703-715. ISBN: 9781450329576. DOI: 10.1145/2660267.2660315. URL: https://doi.org/10.1145/2660267.2660315.

Stewart, Charles (2011). «Voting Technologies». En: *Annual Review of Political Science* 14.1, págs. 353-378. DOI: 10.1146/annurev.polisci.12.053007.145205. eprint: https://doi.org/10.1146/annurev.polisci.12.053007.145205. URL: https://doi.org/10.1146/annurev.polisci.12.053007.145205.

Suwarjono, Suwarjono, L. Sumaryanti y Lusia Lamalewa (2021). «Cryptography Implementation for electronic voting security». En: *E3S Web of Conferences* 319, pág. 03005.

DOI: 10.1051/e3sconf/202131903005. URL: https://www.e3s-conferences.org/articles/e3sconf/abs/2021/104/e3sconf_icstunkhair2021_03005/e3sconf_icstunkhair2021_03005.html.

Team, Material-UI (2021). *Material-UI: A popular React UI framework*. URL: https://mui.com/.

Team, React-Bootstrap (2015). *React-Bootstrap: The most popular front-end framework, rebuilt for React*. https://react-bootstrap.github.io/.

Weizenbaum, Natalie y Chris Eppstein (2021). *Sass: Syntactically Awesome Style Sheets*. URL: https://sass-lang.com/.

Wood, Gavin (2014). *ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER*. URL: https://ethereum.github.io/yellowpaper/paper.pdf.

Woodside, J, Fred K Augustine y W Giberson (2017). «Blockchain Technology Adoption Status and Strategies». En: *Journal of Information Technology Management* 30.1, págs. 2-18. DOI: 10.58729/1941-6679.1300. URL: https://dx.doi.org/10.58729/1941-6679.1300.

Zhang, Weijia y Tej Anand (2022). «Ethereum Architecture and Overview». En: *Blockchain and Ethereum Smart Contract Solution Development: Dapp Programming with Solidity*. Berkeley, CA: Apress, págs. 209-244. ISBN: 978-1-4842-8164-2. DOI: 10.1007/978-1-4842-8164-2_6. URL: https://doi.org/10.1007/978-1-4842-8164-2_6.

ANEXO

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. **Fin de la pobreza.**		X		
ODS 2. **Hambre cero.**		X		
ODS 3. **Salud y bienestar.**		X		
ODS 4. **Educación de calidad.**				X
ODS 5. **Igualdad de género.**		X		
ODS 6. **Agua limpia y saneamiento.**				X
ODS 7. **Energía asequible y no contaminante.**				X
ODS 8. **Trabajo decente y crecimiento económico.**			X	
ODS 9. **Industria, innovación e infraestructuras.**	X			
ODS 10. **Reducción de las desigualdades**.		X		
ODS 11. **Ciudades y comunidades sostenibles**.		X		
ODS 12. **Producción y consumo responsables.**				X
ODS 13. **Acción por el clima.**		X		
ODS 14. **Vida submarina.**				X
ODS 15. **Vida de ecosistemas terrestres.**				X
ODS 16. **Paz, justicia e instituciones sólidas.**	X			
ODS 17. **Alianzas para lograr objetivos.**		X		

Reflexión sobre la relación del Trabajo con los ODS y con los ODS más relacionados.
El proyecto que estoy llevando a cabo tiene una fuerte conexión con los Objetivos de Desarrollo Sostenible (ODS) de las Naciones Unidas, y considero esencial destacar cómo mi iniciativa contribuye a estos objetivos globales. La relación de mi proyecto con los ODS no solo refleja la relevancia y pertinencia del mismo, sino que también subraya mi compromiso con la creación de un mundo mejor y más justo.

1. Fin de la pobreza (ODS 1): La implementación de sistemas de votación electrónica puede mejorar la participación ciudadana, especialmente en áreas rurales o marginadas. Un mayor compromiso con la democracia puede traducirse en políticas más incluyentes, que aborden de manera efectiva las causas subyacentes de la pobreza.

2. Hambre cero (ODS 2) y Salud y bienestar (ODS 3): Aunque mi proyecto no se centra directamente en estos objetivos, el empoderamiento de las comunidades a través de sistemas de votación más transparentes y accesibles puede llevar a decisiones políticas que prioricen la salud y la alimentación.

3. Educación de calidad (ODS 4): La votación electrónica puede ser una herramienta educativa, enseñando a las personas sobre tecnología, ciberseguridad y civismo. Además, al promover la participación, se pueden impulsar políticas que favorezcan la educación.

4. Igualdad de género (ODS 5): Mi proyecto promueve la igualdad de oportunidades, permitiendo que mujeres, independientemente de su ubicación o condiciones, tengan un acceso igualitario al voto, ya que no distingue por género.

5. Industria, innovación e infraestructuras (ODS 9): El proyecto es un reflejo de la innovación en la intersección de la tecnología y la democracia, promoviendo el desarrollo de nuevas infraestructuras tecnológicas.

6. Reducción de las desigualdades (ODS 10) y Ciudades y comunidades sostenibles (ODS 11): Al facilitar la participación, especialmente en comunidades marginadas, se promueve una democracia más equitativa. Esto puede llevar a decisiones políticas que reduzcan las desigualdades y construyan comunidades más inclusivas y sostenibles.

7. Acción por el clima (ODS 13): La votación electrónica, al reducir la necesidad de materiales físicos y transporte, puede tener un menor impacto ambiental que los métodos de votación tradicionales.

8. Paz, justicia e instituciones sólidas (ODS 16): Este es uno de los núcleos del proyecto. Un sistema de votación transparente, accesible y seguro puede fortalecer la confianza en las instituciones democráticas y promover la paz y la justicia.

9. Alianzas para lograr objetivos (ODS 17): Estoy abierto a colaboraciones y alianzas para mejorar y expandir el proyecto, reconociendo que mediante la cooperación podemos lograr un impacto más significativo.

Los ODS marcados con un nivel de intensidad "Bajo" o "No Procede" no se abordaron en detalle aquí para centrarse en aquellos con una relación más directa con el proyecto. Sin embargo, es esencial reconocer que todos los ODS están interconectados y que avanzar en uno puede tener efectos positivos en otros. Mi proyecto busca ser una pieza en el puzzle de construir un futuro más sostenible y equitativo.