CIS2010CP02

SISTEMA DE VOTACIÓN BASADO EN BLOCKCHAIN: POCKET BALLOT CHAIN

Brandonn Andrés Cruz López

Briam Daniel Agudelo Mogollón

Diego Gerardo Barajas Suarez

Santiago Alejandro Chaparro Palacio

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA DE SISTEMAS

BOGOTÁ, D.C.

2020

CIS2010CP02

SISTEMA DE VOTACIÓN BASADO EN BLOCKCHAIN: POCKET BALLOT CHAIN

Autores:

Brandonn Andrés Cruz López

Briam Daniel Agudelo Mogollón

Diego Gerardo Barajas Suarez

Santiago Alejandro Chaparro Palacio

MEMORIA DE TRABAJO DE GRADO REALIZADA PARA CUMPLIR UNO DE LOS REQUISITOS PARA EL TÍTULO DE INGENIERÍA DE SISTEMAS

Director

Ing. Rafael Vicente Páez Méndez, PhD.

Jurados

Ing. Edgar Enrique Ruiz García, MS.

Ing. Jaime Andrés Pavlich Mariscal, PhD.

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERIA

PROGRAMA DE INGENIERÍA DE SISTEMAS

BOGOTÁ, D.C.

NOVIEMBRE, 2020

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERIA

PROGRAMA DE INGENIERÍA DE SISTEMAS

Rector de la Pontificia Universidad Javeriana

Jorge Humberto Peláez Piedrahita, S.J.

Decano de la Facultad de Ingeniería

Ing. Eng. Lope Hugo Barrero Solano, PhD.

Director del Programa de Ingeniería de Sistemas

Ing. Alexandra Pomares Quimbaya, PhD.

Director del departamento de Ingeniería de Sistemas

Ing. Efraín Ortiz Pabón, PhD.

Artículo 23 de la Resolución No. 1 de Junio de 1946

"La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia"

AGRADECIMIENTOS

Brandonn Andrés Cruz López

Agradezco especialmente a mi madre y abuela por su increíble apoyo no solo durante estos últimos 5 años sino durante toda mi vida. También agradezco al resto de mi familia por ser pacientes y estar allí cuando los necesito. Agradezco al programa Ser Pilo Paga, así como a la Pontificia Universidad Javeriana por darme esta experiencia tan maravillosa. Finalmente, agradezco al profesor Rafael Vicente Páez y a mis compañeros y amigos de trabajo de grado, pues sin ellos esto no hubiera sido posible.

Briam Daniel Agudelo Mogollón

Agradezco a mi madre por el apoyo en los momentos más difíciles de este arduo proceso, también a mi pareja por alentarme cuando más lo necesitaba, y gracias a mis amigos de la universidad por su aporte en mi desarrollo como profesional y sobre todo como persona. Agradezco a Rafael Páez, quien ha dedicado gran parte de su tiempo a ayudarnos a mí y a mis compañeros a lograr este objetivo. También agradezco al programa de Ser Pilo Paga, ya que sin este nada de esto hubiera podido ser posible.

Diego Gerardo Barajas Suarez

En primer lugar, quiero agradecer a mis padres quienes me han ofrecido su apoyo incondicional durante el transcurso de mi vida y quienes me han transmitido los conocimientos y valores que me marcaron como persona. También agradezco a mis hermanos, en quienes siempre he podido confiar y quienes siempre han estado junto a mi cuando los he necesitado. De igual manera agradezco a mis abuelos personas con las que he crecido las cuales nunca dudaron en brindarme su apoyo y ayuda. De igual forma agradezco a mis compañeros con los cuales he compartido experiencias inolvidables y mis maestros que durante el transcurso de mi etapa escolar y universitaria me motivaron a llegar a este punto, especialmente agradezco al

profesor Rafael Páez quien nos brindó ayuda y guía para el desarrollo de este trabajo. Finalmente, pero no menos importante agradezco al programa de Ser Pilo Paga, ya que gracias a este tuve la oportunidad de estudiar en esta grandiosa universidad.

Santiago Alejandro Chaparro Palacio

Agradezco a mi familia, por todo el apoyo que me dieron durante las etapas más difíciles de la carrera. Agradezco a mis amigos de Valledupar, y Barranquilla: Asaf Aguilar, Hernando Silva y Jesús Padilla. También a mis amigos de Bogotá entre los que se encuentran mis compañeros de proyecto (Briam, Brandonn y Diego), Daniela Beltrán y Nicolas Ospina.

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

ABSTRACT

The vote is one of the biggest tools in the society that take advantage of the technology and started to get some visibility in digital environments. For this reason, the team from this project developed an e-voting system accessible for business environments. This system guarantees the user security based on the blockchain properties. On the other hand, this document also shows the mechanism that maintains the anonymity of the voters. This was archived because of the authentication and the vote storage was separated in two different components of the system.

El voto es una de las grandes herramientas de la sociedad actual que, gracias a la tecnología empieza a tener visibilidad en entornos digitales. Por eso, se desarrolló un sistema de voto electrónico que fuese accesible a entornos empresariales, en el cual se garantice la seguridad de los usuarios, brindándoles un sistema descentralizado. Además, se desarrolla y muestra la integración del sistema con la tecnología blockchain para lograr este objetivo. Por otra parte, se presenta un mecanismo para mantener el anonimato total de los votantes, que se logra al separar la información de los votos y de los votantes en dos sitios diferentes, dado que la información de los votos se almacena en la blockchain, mientras que la información de los votantes se aloja en un sistema de base de datos.

I INTRODUCCIÓN

En este documento se describe el proceso que se realizó para desarrollar la aplicación web Pocket Ballot Chain. Este es un sistema de voto electrónico que se basa en la tecnología blockchain para permitir a las personas realizar votaciones sin la necesidad de un intermediario, teniendo como principio que los mismos usuarios sean quienes guardan los votos.

Para el desarrollo de este proyecto, primero se realizó el análisis del problema, en el cual se expone el contexto en el que tiene cabida este proyecto, se formulan los problemas específicos, se propone la solución y se justifica el uso de la tecnología blockchain como solución a los problemas formulados. Luego, se plantean los objetivos mediante el cual se definió el alcance del proyecto. Una vez identificado el contexto del problema y expuesta la solución que se propone, se expone todo el marco teórico que hay detrás de blockchain y de los sistemas de votación electrónicos que han usado esta tecnología. Aquí también se discute sobre las características que diferencian a otros sistemas con el sistema propuesto Pocket Ballot Chain.

Teniendo claro el contexto del problema y la solución que se propone, se continúa con el resumen de la especificación de los requisitos, en donde se muestra cuáles son las principales funcionalidades y atributos de calidad que el sistema provee. A partir de esta especificación de requisitos, se presenta el diseño del sistema en donde se exponen las principales herramientas utilizadas durante el proyecto, la arquitectura general del sistema y la explicación de su funcionamiento.

A partir del diseño, se realizó la implementación y despliegue de la solución. Luego, se presentan los resultados obtenidos tanto de la aplicación de scrum, como de la página web desarrollada. Se continúan presentando los resultados de las pruebas que se realizaron a la aplicación para asegurar que se cumplieron los requisitos y finalmente se hace un análisis de los resultados del proyecto para formular las conclusiones de este.

II DESCRIPCIÓN GENERAL

1 Oportunidad, Problema

1.1 Contexto del problema

Los sistemas de votación han ganado mucha atención en los últimos años. Sin embargo, la mayoría de las investigaciones de seguridad e integridad se han enfocado en los sistemas a gran escala, como los sistemas de votación gubernamentales. No obstante, el voto también es una herramienta importante para la toma de decisiones en grupo [1]. El voto electrónico es conocido por ser una herramienta que puede hacer el proceso electoral más eficiente, confiable y seguro si es que está bien implementado. Si no lo está, este puede perjudicar la confiabilidad de todo el proceso de votación[2]. En la actualidad, existen proyectos de sistemas de votación electrónicos que aprovechan las ventajas que ofrece blockchain para atacar los problemas de seguridad. Estos sistemas priorizan la confidencialidad y el anonimato de los votos.

En un contexto cerrado, como el empresarial, blockchain ha demostrado que puede mejorar la relación entre una empresa y sus accionistas. Las reuniones generales anuales (AGM) presentan muchos fallos cuando se trata de la votación remota, que un sistema de voto electrónico basado en blockchain puede solucionar. Estos fallos están relacionados con problemas de transparencia, verificabilidad y autenticación. Ya existen varias iniciativas de votos electrónico (*e-voting)* enfocados a las votaciones durante las AGM. No obstante, dichas iniciativas han dado como resultado prototipos y pruebas de concepto con escasa documentación y poca retroalimentación pública[1]. Entre los sistemas de votación por blockchain más populares se encuentran:

1. *Vote Coin*: es un sistema de votación descentralizado que utiliza *Zcash*. Este último es un sistema de blockchain público basado en criptomonedas y enfocado en proteger la privacidad cifrando al emisor, al receptor y el mensaje de cada transacción[3].

2. *Coalichain*: es otro proyecto basado en blockchain que se enfoca en la creación de no solo un sistema de votación, sino que también permite un ecosistema que elimina la brecha entre los votantes y los representantes. Este proyecto también tiene servicios adicionales que facilitan la transparencia y la trazabilidad de los datos[3].
3. *Polys*: es un proyecto similar a los anteriores que hace uso de contratos inteligentes y que permite realizar la votación por medio de una plataforma de internet con algoritmos de criptografía transparente[4].

Durante los últimos dos siglos, se ha tratado de encontrar un esquema de elección social perfecto. Dicho esquema debe satisfacer cuatro condiciones: universalidad, monotonía, preferencia no impuesta y equidad. Sin embargo, se ha determinado que es imposible que un solo esquema de elección social pueda cumplir todas las condiciones[4].

Por esta razón, es deseable que el sistema de votación no se limite a seguir solamente la metodología tradicional de votación. Las votaciones por medio de clasificación, evaluación y aprobación de los candidatos son muy comunes en los sistemas de votación de baja escala, pero muy poco comunes en los sistemas de votación electrónicos basados en blockchain. La complejidad de incluir un sistema de clasificación o evaluación durante unas elecciones de gran escala es muy elevada, pero si el proceso electoral es relativamente cerrado, es una opción viable[1].

1.1 Formulación del problema

Teniendo en cuenta lo anterior, es necesario disponer de mecanismos que apoyen los procesos que requiera la toma de decisiones a baja escala. Sin embargo, los sistemas de voto electrónico actuales tienen varias falencias, de las cuales, para el desarrollo de este proyecto se van a tener en cuenta las siguientes[3] [4]:

- No hay confianza con los administradores del sistema o con el sistema mismo.
- No ofrecen trazabilidad o auditabilidad en sus sistemas de votación.

De esta forma, resulta necesario un sistema de voto electrónico que resuelva las falencias mencionadas.

1.2 Propuesta de solución

Se propone desarrollar un sistema de votación electrónico basado en la tecnología blockchain que puede ser utilizado por: empresas, instituciones educativas, asambleas, conjuntos residenciales y en general comunidades que quieran usar una aplicación confiable de voto electrónico. Este sistema busca solucionar los problemas planteados. Como cualquier sistema de votación electrónico, la solución propuesta se divide en tres etapas: preparativos, votaciones y conteo de votos.

En la etapa de preparativos, el organizador ingresa las entradas para la votación: candidatos, votantes válidos, la fecha de inicio, la fecha final y el tipo de votación. Los tipos de votación son: **ranking**, donde se debe ordenar un conjunto de opciones y la opción ganadora es la que en promedio obtiene mejores puestos que las demás opciones. **Clasificación**, donde se debe situar cada opción entre un conjunto de grupos y al final se cuenta cuantas veces fue colocada dicha opción en cada grupo. Por último, **voto popular**, donde se debe distribuir una cantidad de votos entre un conjunto de opciones y al final la opción ganadora es la que obtenga la mayor cantidad de votos.

En la etapa de votación, el votante se autentica mediante usuario y contraseña en el sistema. Luego, el votante emite su voto, el cual se cifra y se envía a un conjunto de nodos distribuidos denominados validadores. Entre los validadores, un líder propone un bloque que contiene los votos. Luego, los demás validadores verifican la consistencia del bloque frente a la blockchain correspondiente a la votación y finalmente se agrega el bloque a la blockchain indicada si se ha logrado un consenso.

Una vez alcanzada la fecha de finalización de la votación, los nodos de la red dejan de recibir votos para esa votación y se generan los resultados correspondientes en el momento en que alguno de los votantes o el organizador de la votación los solicite.

1.3 Justificación de la solución

La solución propuesta surgió por la necesidad de resolver las falencias mencionadas en la sección 1.1. Por lo cual, se optó por aprovechar las características de integridad, descentralización y trazabilidad propias de la tecnología blockchain [5]. De esta forma, se resuelven los problemas de confianza y auditabilidad de los sistemas de votación electrónico convencionales, asegurando:

1 La integridad e inmutabilidad en las votaciones. Dado que una votación esté activa, no será posible modificar los votos validados y agregados a la blockchain. Por consiguiente, las características de integridad e inmutabilidad en las votaciones generan confianza en los votantes, ayudando a solucionar el problema de desconfianza en los sistemas de votación electrónico.

2 La descentralización en la validación de los votos. Dado que el proceso de validación no es ejecutado en una sola máquina, no es posible la existencia de una entidad administradora que posea los suficientes privilegios para intervenir deliberadamente en el proceso de validación en una votación. De esta forma, se ayuda a solucionar el problema de desconfianza sobre los administradores que puedan intervenir en los procesos de validación de votos electrónicos, esto debido a que las características de blockchain no lo permiten.

3 La trazabilidad en las votaciones. Dado que cada vez que se emite un voto, la blockchain permite almacenar un registro de la transacción que contiene el voto, junto con un seudónimo del votante que no lo relaciona con su identidad (para preservar el anonimato), es posible verificar la integridad de las votaciones, comparando los seudónimos de los votos frente otros registros externos (e g. bases de datos).

2 Descripción del Proyecto

2.1 Objetivo General

Desarrollar un sistema de votación electrónico basado en blockchain que permita a las organizaciones crear distintos tipos de votaciones (ranking, clasificación y voto popular).

2.2 Objetivos específicos

1. Especificar el levantamiento de requerimientos del sistema

2. Diseñar la arquitectura del sistema de votación propuesto
3. Desarrollar los componentes de la arquitectura del sistema de votación basado en blockchain
4. Realizar pruebas y ajustes del sistema

2.3 Entregables, estándares y Justificación

Tabla 1. Entregables y estándares del proyecto

Entregable	Estándares asociados	Justificación
SPMP	ISO/IEC/IEEE 16326 - 2009	Este estándar tiene la ventaja de poder aplicarse en prácticamente cualquier tipo de proyecto de desarrollo de software y no necesariamente en los más nuevos. No obstante, hay que tener en cuenta que no es ideal usarlo en proyectos pequeños [7].
SRS	IEEE Std 830-1998 ISO/IEC/IEEE 29148 – 2011	Este estándar describe los enfoques recomendados para la especificación de los requisitos de software. Se basa en un modelo en el que el resultado del proceso es un documento de especificación completo y sin ambigüedades.

		Esta norma internacional proporciona un tratamiento unificado de los procesos y productos involucrados en la ingeniería de requisitos a lo largo del ciclo de vida de los sistemas y el software.
SDD	IEEE Std 1016-2009	Esta norma específica los requisitos sobre el contenido de información y la organización de los SDD. El estándar especifica los requisitos para la selección de los lenguajes de diseño que se utilizarán para SDD, y los requisitos para documentar los puntos de vista de diseño que se utilizarán para organizar un SDD.
Código fuente del prototipo	Google Style Guides	Las guías de estilos de Google a pesar de no ser un estándar ingenieril establecen lineamientos para los proyectos de código abierto. Seguir las guías de estilos garantizan el entendimiento del código fuente y una uniformidad en el mismo.

Documentación para pruebas de software	ISO / IEC / IEEE 29119 – 3	El estándar IEEE 29119 – 3: es una parte del estándar IEEE 29119 y especifica los documentos necesarios para un proceso de prueba de software. Además, de ofrecer ejemplos para la fácil elaboración de los documentos. [8]
Prototipo de la solución	ISO 25010	El ISO 25010 establece varias características con las que debe contar un software de calidad, a pesar de que el prototipo no se certificara con el ISO 25000 y no asegurara poseer todas las características para hacerlo, si se desarrollara en base a él. [9]
Manual de usuario	ISO/IEC 26514:2008	Este estándar fue creado para asistir a los diseñadores y desarrolladores en el diseño y creación de la documentación de usuario como parte del ciclo de vida del software. Incluye documentación impresa, de pantalla y del producto [10].

Post-mortem	Introduction to the team software process	Plantea un proceso de post-mortem para evaluar el desempeño del trabajo realizado, que se ajusta a la metodología ágil que sigue el grupo.

III CONTEXTO DEL PROYECTO

1 Marco teórico

El voto es una herramienta importante para la toma de decisiones en grupos, empresas u organizaciones sobre todo cuando estas involucran información sensible o decisiones monetarias[1]. En la actualidad es muy común ver como este tipo de organizaciones tienen principalmente dos métodos para realizar sus procesos de votación, mediante software que les facilite la planeación y desarrollo de las votaciones o por medios más convencionales como el papel, que a pesar de ayudar a la auditabilidad de las votaciones se presta para problemas de seguridad y de planificación. Además de esto, en la actualidad por situaciones como la pandemia y el confinamiento causado por el Covid-19 alrededor del mundo, las metodologías de votación en varias empresas u organizaciones han cambiado. Las votaciones en papel no son igual de efectivas y en muchos casos no se pueden realizar, por estas razones se considera al *e-voting* como una alternativa para estas situaciones [6].

El voto electrónico se refiere al uso de medios electrónicos que se usan durante un proceso de elección. Para que se considere que un sistema de votación posee *e-voting*, por lo menos la introducción de votos debe estar digitalizada [2].

El sistema *e-voting* desarrollado para dar solución a la problemática planteada en la sección 1.1 del presente documento se basa en una arquitectura blockchain. Esta tecnología consiste en una red de nodos *peer-to-per* (P2P) en donde un nodo puede actuar como cliente y servidor; al mismo tiempo que comparten una estructura de datos sin la necesidad de intervención de una autoridad centralizada. Esta estructura de datos es conocida como *ledger* (libro mayor en español), la cual se compone de bloques que almacenan información referente a intercambios de bienes o transacciones en la red. Cada bloque tiene su propio *hash* que es un código calculado a partir de la información de este, a su vez, el bloque también almacena el *hash* del bloque anterior. Esta característica le otorga la cualidad a la blockchain de ser difícilmente modificable, porque habría que cambiar todos los bloques que se encuentran almacenados

en la mayoría de los nodos de la red [7], [5]. En la Figura 1 se muestra un ejemplo de cómo se vinculan las transacciones que se encuentran en los bloques de la blockchain.

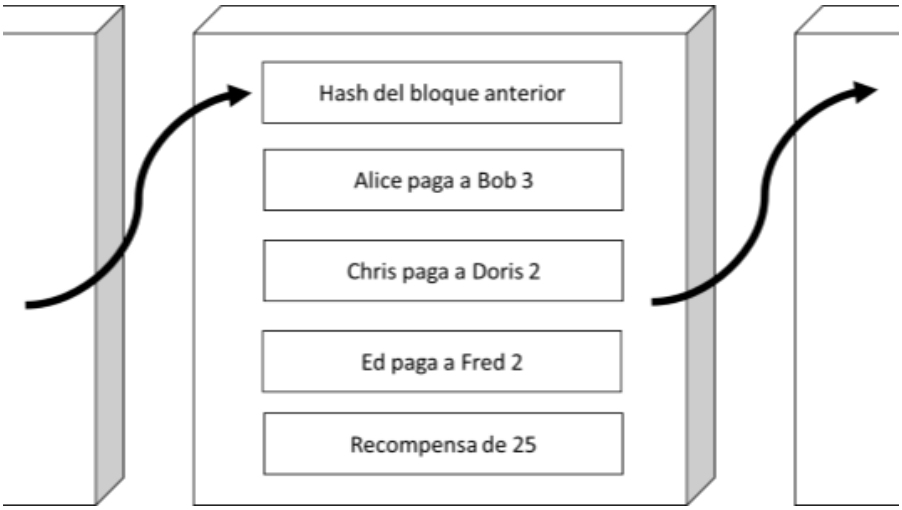

Figura 1. Concatenación de bloques en blockchain [7]

A fin de agregar un bloque a la blockchain, este se debe insertar en la estructura de datos que se aloja en cada nodo del sistema P2P. Para coordinar como se agrega cada bloque se utilizan los algoritmos de consenso. Actualmente hay dos tipos de algoritmos de consenso: *lottery-based,* que consisten en seleccionar aleatoriamente un nodo de la red para que proponga nuevos bloques; teniendo como ejemplos *proof-of-work* (PoW) y *proof-of-stake* (PoS). El segundo es *voting-based,* donde por medio de algún mecanismo se selecciona un nuevo líder, que es el que propone los bloques, mientras que los validadores los verifican. En este segundo tipo entran los protocolos *Bizantine Fault Tolerance* [8].

El primer tipo es usado generalmente en redes públicas, donde hay un mayor número de nodos, aunque su costo en términos de tiempo y recursos es alto. Con respecto al segundo tipo, este es más adecuado para redes privadas, donde los participantes de la red son conocidos. Su mayor costo está basado en el intercambio de mensajes, por ello no es adecuado en términos de escalabilidad [8].

Otro aspecto importante que diferencia a los dos tipos de algoritmos de consenso es que *lottery-based* puede permitir más de un ganador, en cuyo caso el tiempo resolverá el problema, cuando una de las cadenas de los ganadores sea la más larga. A esta solución se le conoce como consenso eventual. Por otro lado, *voting-based* si garantiza la finalidad de consenso, ya que los votantes eligen un ganador específico (en caso de empate siempre se resuelve por algún mecanismo).

A partir de los dos tipos de algoritmos de consenso mencionados anteriormente se deriva un tercer tipo, de manera que se genera un algoritmo de consenso híbrido, que busca permitir la escalabilidad de los *lottery-based* y la velocidad de los *voting-based* [8]. Este algoritmo de consenso se conoce como P*ractical Bizantine Fault Tolerance* (PBFT). Su funcionamiento está basado en vistas, en las cuales existe un líder y un conjunto de réplicas (usuarios que realizaran el proceso de validación de los bloques). Al momento de crear un bloque, las réplicas iniciarán un intercambio de mensajes para alcanzar un consenso de las transacciones a agregar. Por ello, si hay un mal comportamiento del líder, las réplicas se darán cuenta y elegirán uno nuevo. Además, una de sus ventajas es que, si *f* de *N* nodos son controlados por un atacante, el algoritmo de consenso PBFT garantiza una fuerte consistencia con $f < N/3$ [8].

Uno de los algoritmos de consenso basados en PBFT más conocido y de mayor consideración para el presente proyecto es *proof-of-authority* (PoA o prueba de autoridad por su traducción al español) es un algoritmo de consenso que hace parte de la familia de los algoritmos de consenso híbridos. PoA se basa en un conjunto de nodos confiables denominados autoridades, las cuales se identifican por un *id.* El algoritmo sigue un esquema de rotación, que asigna la responsabilidad de crear bloques a una de las autoridades, la cual se conoce como líder. El líder propone un nuevo bloque y los demás lo validan. Si la validación es exitosa el bloque es agregado a la cadena, si no, los miembros pueden reportar al líder y este podría perder su puesto como validador, por ende, las recompensas de este cargo. Además, en PoA las autoridades son conocidas por todos los nodos de la red, por tanto, también arriesgan su reputación [8].

Dado que la solución desarrollada trabaja de manera distribuida, se debe intercambiar gran cantidad de información entre usuarios asegurando la confidencialidad de los datos, para esto se hace uso de criptografía asimétrica. En este tipo de criptografía se usan dos claves: la clave pública, que es conocida por el usuario que encriptará la información, y la clave privada, que es conocida sólo por el emisor de la llave publica, esta llave se usa para desencriptar la información [9].

Aparte de la utilización de cifrado asimétrico se utilizarán las firmas digitales para validar el origen e integridad del mensaje enviado entre los nodos del sistema. La implementación de firmas digitales proviene de la criptografía asimétrica. Al igual que esta, las firmas digitales funcionan con un par de llaves una pública y una privada. Sin embargo, a diferencia de la criptografía asimétrica en este caso se utiliza la clave privada para realizar el cifrado por parte del emisor del mensaje, a este mensaje se le adjunta la clave publica o se deja disponible en un servidor para que los receptores puedan utilizarla y que servirá para validar la firma del voto. Este mecanismo permite que cualquier sistema pueda validar la firma del voto, pero solo el que tenga la clave privada puede firmar el mensaje [10].

En cuanto a la etapa de desarrollo, se decidió utilizar la metodología Scrum la cual, según la guía de Scrum escrita por *Schawaber* y *Sutherland,* brinda al proceso de desarrollo del producto las características de transparencia, inspección y adaptabilidad [11]. Scrum se rige por una serie de eventos que se realizan de forma regular durante los *sprints;* cada sprint corresponde al periodo en el cual se desarrolla una parte del producto potencial. Los eventos realizados durante los *sprints* son: *Scrum Planning*, en este se deciden las actividades a desarrollar en el *sprint*; el *Daily Scrum*, es una reunión diaria donde se ve el avance del *sprint*; el *Sprint Review*, reunión que se realiza al final del sprint para desarrollar medidas que mejoren el proceso de trabajo, y por último el *Sprint Retrospective*, el cual es igual al scrum *review*, pero en este solo se reúnen los integrantes del *Scrum team.* Mas adelante se especificará como se realizó el proceso de scrum y cuales modificaciones fueron necesarias para el correcto funcionamiento en el equipo de trabajo.

2 Análisis del contexto

El voto electrónico es una herramienta de la cual se venía hablando desde el año de 1869, como un sistema propuesto por el reconocido científico Thomas Alva Edison, quien propuso un sistema electromecánico que permitiría realizar votaciones para el congreso de Estados Unidos, este proyecto fue rechazado por diversas complicaciones de seguridad. A pesar de esto, el científico patento su idea de voto electrónico (véase la patente *90646)* [12]. Posteriormente, el voto electrónico se ha venido viendo como una alternativa más viable gracias a las nuevas tecnologías [13]. Estas mismas tecnologías han permitido la creación de variedad de herramientas que facilitan su uso a nivel empresarial, por ello es importante mostrar los beneficios y diferencias principales con el proyecto realizado.

Conociendo esta información cabe recalcar que la solución desarrollada se basa en la utilización de un sistema blockchain en donde los usuarios son los responsables de validar los votos, garantizando que ninguna entidad sea capaz de modificarlos. Además, nuestro sistema blockchain se respalda con una base de datos que permite asegurar el voto secreto, pues a pesar de que en la base de datos se lleva registro de quienes fueron las personas que participaron en determinada votación, no almacena los votos que realizaron. El único sitio donde se almacenan los votos es en la blockchain. Sin embargo, esta solo conoce una parte de la información, la cual corresponde a los votos realizados y un pseudónimo o clave única que identifica que el voto es válido.

Con lo expuesto anteriormente, al no relacionar un voto con un votante se garantiza la confidencialidad de los participantes. Esto también brinda la ventaja a las votaciones de ser auditadas, pues se debe lograr contrastar la cantidad de votos almacenados en la blockchain con la cantidad de seudónimos almacenados en la base de datos.

El funcionamiento específico del sistema se muestra más adelante en este mismo documento. En resumen, el sistema propuesto garantiza votaciones descentralizadas en donde la comunidad de usuarios es la encargada de contar y registrar los votos. Además, asegurá la confidencialidad de los votantes, es auditable, de fácil acceso, ofrece diversos tipos de votación y facilidades para la visualización de resultados.

Entre los principales competidores con características similares al sistema propuesto encontramos a *Polys, eBallot, Simply voting, BallotChain* y *Proof of Vote*.

Polys

Polys es un sistema de votación basado en blockchain que provee tres tipos de servicio de votación: votación para los cuerpos gubernamentales estudiantiles, partidos políticos y para procesos de financiación participativa. Toda la documentación disponible se encuentra en la página oficial Polys.me. Esta plataforma solo tiene disponible dos tipos de voto: el voto tradicional, donde el candidato con más votos gana y el voto acumulativo, en donde a un votante se le da una cantidad específica de votos que puede asignar a los candidatos que quiera. La blockchain de *Polys* utiliza el algoritmo de consenso *proof of work*, esto conlleva a que el despliegue de *Polys* tenga alto costo energético [14].

Finalmente, cabe resaltar que *Polys* no realiza su autentificación por medio de su blockchain. Según la página oficial de la plataforma, en diciembre del 2018 el sistema tuvo una actualización mediante la cual toda la información relacionada con los votantes y los organizadores de las votaciones se guardaba en un servidor. Esto se debe a que la centralización de los datos personales se volvió un requerimiento estipulado en la GDPR (*General Data Protection Regulation*) [14].

eBallot

eBallot es un sistema de votación electrónico que se divide en dos productos principales: *eBallot for Bussiness* y *eBallot Ess*ential. El primero está enfocado en el proceso electoral a nivel organizacional, mientras que el segundo se centra en brindar una forma de crear votaciones de forma rápida y fácil de usar. *eBallot* utiliza *Amazon Web Services* para almacenar datos y llevar a cabo el proceso electoral. También utiliza redundancia para asegurar la disponibilidad del servicio y de los datos de sus usuarios [15].

eBallot no contiene ningún tipo de documentación oficial que esté pública, pero su página web *eballot.com* explica las características de varios componentes del sistema. Los componentes que más destacan de *eBallot* son: Los Mecanismos de autentificación para evitar los votos repetidos, la personalización de votos, y los servicios que permiten facilitar su auditoría. A continuación, se explican las características principales de cada uno de los componentes [15]:

a. El proceso de autentificación en *eBallot* se puede realizar a partir de parámetros estándar, como el correo electrónico, el número de identificación o el número de teléfono. Asimismo, también tiene una funcionalidad que permite generar *tokens* aleatoriamente para que los usuarios se puedan autenticar [15].
b. La personalización de los votos ofrece varias opciones, entre las que se encuentra: Las opciones de anonimidad, el cual se implementa dependiendo de las necesidades organizacionales; el peso del voto, el cual puede ser configurado de tal forma que a un grupo de personas se les considere más el voto que a otras; el filtrado de votantes, que permite que solo un grupo de personas puedan votar en un proceso específico, y la restricción por dirección IP, el cual determina desde donde se puede votar [15].
c. Para el proceso de auditoria, *eBallot* genera reportes que, dependiendo del nivel de anonimidad de la votación, permite obtener información relevante del proceso electoral, como, por ejemplo: quien voto, por quien voto y cuando voto. De forma exclusiva, en *eBallot for Bussiness* se permite retener los datos de la votación y exportar el *raw data.* Este proceso contiene información relevante como el grupo o departamento al que pertenece cada votante [15]. Por otra parte, estas auditorias presentan el problema que se debe vincular los votos con los votantes, lo cual es una de las características que soluciona Pocket Ballot Chain.

Simply Voting

Simply Voting es un sistema de votación en línea basado en la web, que ayuda a administrar las elecciones de manera fácil y segura [16].

A continuación, se explica el proceso implementado por *Simply Voting* [16]:

1. Preparación de la votación:

   En esta etapa, el organizador de la votación especifica los parámetros necesarios para poder iniciar la votación. El sistema logra dar soporte a esta funcionalidad mediante el módulo de *Election Manager,* con la característica *easy-to-use*. De esta forma, el usuario (organizador de la votación) puede especificar: fechas, horas, preguntas y método de autenticación.

   Asimismo, en cualquier momento de esta etapa, el usuario siempre tendrá la opción de contactar con el soporte (*Rapid-support*). Cabe resaltar que este soporte es limitado y que el soporte más avanzado solo hace parte de la versión premium.

2. Votación:

   Luego de que la votación es creada, los votantes deben autenticarse mediante el método definido en la etapa anterior. Luego de la autenticación, el votante emite su voto, el cual permanece cifrado y es anónimo. Por último, se le entrega un recibo al votante, el cual contiene un código que servirá para verificar su voto luego de que la votación finalice.

3. Resultados:

   Luego de que se cumple la fecha y hora de finalización de la votación, se inicia la tabulación de los votos de forma automática. *Election Manager* permite visualizar los resultados de la votación, mediante diferentes formatos y antes de que estos sean publicados.

   Posteriormente, los resultados son publicados por el organizador en un sitio web personalizado que provee *Simply Voting*. En este sitio web se puede descargar un archivo con una tabla, que contiene los votos y los códigos de los recibos. De esta forma todos los votantes pueden verificar sus votos.

BallotChain

Reply es una organización especializada en ayudar al crecimiento de empresas ofreciéndoles herramienta tecnológicas y consultorías para mejorar sus ventas. Dentro de los servicios ofrecidos por *Reply* encontramos a *BallotChain* un sistema de votación basado en blockchain que funciona bajo BaaS (*Blockchain as a Service*), donde su principal atractivo es garantizar la seguridad y trazabilidad de sus votaciones. El sistema asegura el anonimato de los votos y fácil uso. Las etapas de la votación en *BallotChain* son:

1. Preparación de la votación:

En esta etapa el administrador de la empresa encargado de realizar el proceso de votación decide como se va a constituir la boleta de votación en la cual se pueden incluir imágenes y textos. Luego de que se definen las personas por las cuales será posible votar, se le dará un token al administrador con el cual los electores tendrán acceso a la votación.

2. Proceso de Votación

En esta etapa, los usuarios asignados para votar realizarán su elección por alguna de las plataformas ofrecidas por *BallotChain*, los electores pueden realizar las elecciones durante un periodo de tiempo asignado por el administrador de la votación.

3.Proceso posterior a las votaciones:

En esta etapa, los resultados serán mostrados a los usuarios, si así lo decidió el administrador al momento de crear la votación. De no ser así los resultados solo estarán disponibles para el administrador. Estos resultados son generados a partir de la blockchain creada para esa votación y se generan instantáneamente una vez finalizada la votación.

Proof of vote

Proof of Vote usa un protocolo que permite votaciones E2E. Además, hace uso de la tecnología blockchain para asegurar la verificabilidad, seguridad y transparencia de las elecciones. El sistema se enfoca en proveer evidencia irrefutable de que el voto de un usuario fue contado [17].

Las principales características de este sistema son:

1. Credenciales de los votantes creadas con criptografía *ElGamal*
2. Uso de pseudónimos para los votantes
3. Se alcanza anonimato por medio de criptografía homomórfica y redes de mezcla
4. Verificación de votos por nodos
5. Uso de blockchain

A continuación, se hace un breve resumen de los pasos de este protocolo:

1. Creación de las elecciones: Una autoridad define las reglas de las elecciones. Estas son fijadas, haciendo uso de un *hash* y una firma digital.
2. Generación de llave: Por medio de criptografía *ElGamal* se crea la llave pública de la votación.
3. Creación del pseudónimo: Se crea el pseudónimo de los votantes, el cual es único para esa votación.
4. Autenticación y autorización: Los votantes se autentican por medio de un sistema que involucra un conjunto de autenticadores. Una vez autenticado, el votante recibe el ID del formato de una boleta.
5. Inicialización y emisión de la boleta: Cuando el votante tiene todo lo necesario para votar, se comunica con la red blockchain para ser aceptado por un conjunto de servicios de autenticación, luego, solicita a uno de los nodos una boleta.
6. Envío y almacenamiento del voto: La boleta es marcada, encriptada y luego enviada a la red, la cual expide una confirmación al votante si todo sale bien.
7. Finalización y conteo: Cuando la fecha final de las votaciones es alcanzada, si se quiere, una red de mezcladores toma todos los votos emitidos para volverlos anónimos completamente. Por último, estos son desencriptados y contados [17].

Comparación entre alternativas

Para realizar la comparación entre las diferentes alternativas que se pueden encontrar para realizar votación a nivel empresarial se tomaron en cuenta los siguientes criterios:

1. Cuenta los votos de manera descentralizada garantizando que no es posible monopolizar el conteo de votos.
2. Brinda transparencia a los usuarios permitiéndoles participar en el proceso de validación.
3. Fácil de usar, con diversos tipos de votaciones y con respuestas inmediatas una vez terminada la votación.
4. Almacena votos y votantes y además desvincula cualquier relación entre ellos.
5. Auditable, cuenta con un mecanismo capaz de identificar si los resultados de la votación concuerdan con la información, tanto en el sistema como en la blockchain, permitiendo validar la veracidad de una votación.

Una vez analizados los diversos sistemas se realizó la Tabla 2, en donde se comparan las ventajas de la solución propuesta con las alternativas actuales del mercado.

Tabla 2. Comparación de la propuesta con otras alternativas

	1	2	3	4	5
P*olys*	Si, utiliza un sistema blockchain para garantizar esto	No, su blockchain es manejada por una sola organizacion	Si, cuenta con varios tipos de votaciones	No se cuenta con información al respecto	No, La plataforma no especifica una manera de lograrlo
eBallot	Si, utiliza un sistema blockchain para garantizar esto	No, su blockchain es manejada por una sola organización	SI, Su sistema es muy personalizable y fácil de usar	No, Es uno de los principales problemas de la plataforma	Si, la plataforma incorpora un mecanismo de auditabilidad

Simply voting	No, *el sistema es centralizado*	No, *el sistema es centralizado*	Si, Su nivel de personalización en muy bueno	No se cuenta con información al respecto	No, no tiene un sistema para hacer auditorias
BallotChain	Si, *el sistema usa un blockchain para garantizar esto*	No se cuenta con información al respecto	Si, Su nivel de personalización en muy bueno	No se cuenta con información al respecto	No, no tiene un sistema para hacer auditorias
Proof of Vote	No, *El sistema es centralizado*	No, *El sistema es centralizado*	No, El sistema prioriza la seguridad ante la usabilidad	Si, El sistema no vincula votos y votantes al almacenarlos, a pesar de esto si los vincula mientras se realiza la votación.	Si, la plataforma presta un sistema de auditorias
Sistema Propuesto (Pocket Ballot Chain)	SI, *el sistema usa un blockchain para garantizar esto*	SI, el sistema permite contar los votos a la comunidad de votantes haciendo el proceso transparente	SI, Presenta varios tipos de votación y resultados instantáneos una vez finalizan las votaciones	Si, el sistema no vincula los votos con los votantes en ningún punto de la votación ni posteriormente a la mimas.	Si, gracias a la utilización de una base de datos y una blockchain se cuenta con un sistema para realizar auditorías a las votaciones

IV ANÁLISIS DEL PROBLEMA

En este capítulo se resume la especificación de requisitos del proyecto. La información que se muestra a continuación hace referencia a aquellos requisitos que fueron más significativos para el sistema. Para ver la explicación detallada de los mismos, consulte la sección 7 del anexo SRS_voto_electronico, el cual corresponde al documento de especificación de requisitos de software del proyecto.

Durante el ciclo de vida del desarrollo del software, en la fase de análisis se lleva a cabo la especificación de los requisitos que fueron identificados en la elicitación de requisitos [18]. Para este proyecto, esta etapa dio como resultado los siguientes artefactos: modelo de dominio, diagrama de casos de uso para los requisitos funcionales y especificación de los requisitos no funcionales.

1 Requisitos

"Los requisitos para un sistema son descripciones de lo que el sistema debe hacer: el servició que ofrece y las restricciones en su operación" [19]. A continuación, se presentan los requisitos del sistema Pocket Ballot Chain. Primero se listan los requisitos funcionales (RF), que corresponden a los servicios que se ofrecen; luego, se presentan los requisitos no funcionales (RNF), que corresponden a las restricciones en la operación de estas funcionalidades.

1.1 Requisitos funcionales

En la Tabla 3 se listan los principales requisitos funcionales del sistema Pocket Ballot Chain. Para ver la lista de todos los requisitos funcionales consulte la sección 7 del anexo SRS_voto_electronico.

Tabla 3. Requisitos funcionales

CÓDIGO	**REQUISITO**
REQ-001	El sistema permite que los usuarios del sistema se registren

REQ-002	El sistema permite iniciar sesión a los usuarios con cuentas registradas
REQ-004	El sistema permite a los usuarios autenticados crear votaciones
REQ-005	El sistema permite que los creadores de una votación escojan entre las siguientes metodologías de votación: ranking, clasificación y voto popular
REQ-007	El sistema permite a los usuarios ver las votaciones a las cuales han sido agregados y las que han creado
REQ-010	El sistema permite que los usuarios creen grupos de votación
REQ-025	El sistema permite a los usuarios registrados postularse como validadores en la blockchain
REQ-026	El sistema permite que el creador y los participantes de una votación generen un reporte que muestren los resultados finales de una votación
REQ-029	El sistema debe seleccionar a los validadores de la blockchain por medio de un torneo que compare sus reputaciones
REQ-030	El sistema selecciona validadores líderes
REQ-031	El sistema permite que cada validador líder proponga un bloque para la blockchain
REQ-032	El sistema permite que los validadores activos validen el bloque propuesto por el líder
REQ-033	El sistema permite que lo validadores realicen el conteo de los votos
REQ-034	El sistema permite a los validadores guardar bloques válidos en su propia blockchain

1.2 Requisitos no funcionales

En la Tabla 4 se presentan los requisitos no funcionales más importantes del sistema Pocket Ballot Chain.

Tabla 4. Requisitos no funcionales

CÓDIGO	**REQUISITO**
REQ-037	El sistema se debe ejecutar en cualquier dispositivo con alguno de los siguientes navegadores (Chrome, Safari, Mozilla y Opera) y además debe contar con acceso a internet
REQ-039	La red de validadores asegura la consistencia de la blockchain mientras al menos el 60% de los validadores funcionen correctamente
REQ-040	El sistema soportará un máximo de 10 validadores activos simultáneos
REQ-042	El sistema debe almacenar la información de usuarios, grupos y votaciones en la base de datos
REQ-043	El sistema debe almacenar los votos en una blockchain por cada votación
REQ-044	El sistema debe soportar como mínimo la conexión con la base de datos de 200 usuarios simultáneos

2 Restricciones

Las restricciones del sistema se clasificaron en cinco categorías: restricciones de adaptabilidad, disponibilidad, confiabilidad, escalabilidad y seguridad; como se menciona en [20].

2.1 Restricciones de adaptabilidad

- El sistema *está diseñado para ejecutarse correctamente* en los *siguientes* navegadores*:* Chrome, Safari, Mozilla y Edge. *Además, e*l sistema posee las cualidades de una página web responsive.

2.2 Restricciones de disponibilidad

- La disponibilidad del sistema depende del servidor en el que se desplegó.
- La disponibilidad de la blockchain depende de que haya por lo menos un validador activo en el sistema.

2.3 Restricciones de confiabilidad

- La confiabilidad de la red de validadores (donde se validan, almacenan y cuentan los votos) se asegura, mientras al menos el 60% de los validadores funcione correctamente y no estén corruptos. Si este porcentaje no se cumple, la red de validadores estará comprometida y el sistema no será capaz de detectar esta falla. En este caso, los votos almacenados de la blockchain podrían estar corruptos.

2.4 Restricciones de escalabilidad

- El sistema debe soportar 5 validadores por defecto.
- El sistema soporta un máximo de 10 validadores activos simultáneos. Esto, debido a que se debe establecer un equilibrio entre seguridad y rendimiento.

2.5 Restricciones de seguridad

- El sistema asegura la integridad de las votaciones, mientras al menos el 60% de los validadores no se encuentren comprometidos.
- El sistema asegura la autenticidad del voto, mientras que el dispositivo de emisión del voto no se encuentre comprometido.
- El sistema provee autenticación por medio de las credenciales: nombre de usuario y contraseña.

3 Funcionalidad

En esta sección se hace una descripción de alto nivel de las principales funcionalidades del sistema, agrupadas según su responsabilidad.

- **Administración de cuentas:** el sistema permite a los usuarios registrarse por medio de nombre de usuario, correo y contraseña. También, permite iniciar sesión ingresando el nombre y la contraseña para permitir a los usuarios usar las demás funcionalidades del sistema.

  Los casos de uso relacionados con la administración de cuentas son: Registrar e Iniciar sesión.

- **Votaciones:** un usuario que ha iniciado sesión puede crear una votación de tipo popular, clasificación o ranking. Cuando se crea se debe especificar el título de la votación, una descripción, fecha de inicio, fecha final, opciones por las cuales se puede votar y los usuarios que pueden participar en esta. Las votaciones creadas están disponibles en la lista de votaciones de los usuarios que han sido añadidos como participantes. Con esta lista los usuarios pueden votar o consultar los resultados dependiendo de la fecha límite de la votación. Las votaciones también podrán ser visualizadas por sus creadores, sin embargo, si ellos no se agregaron como participantes, únicamente podrán consultar los resultados de estas.

  Los casos de uso asociados con las votaciones son: Crear votación, Invitar votantes, Ver mis votaciones y Generar reporte.

- **Votar:** cuando el usuario ha seleccionado una votación que aún no ha finalizado, este puede diligenciar su voto según el tipo de la votación (popular, ranking o clasificación). Cuando este voto es emitido, este es enviado a los validadores para ser almacenado en la blockchain.

  Los casos de uso asociados con votar son: Votar.

- **Grupos:** un usuario que se ha autenticado puede crear grupos invitando a usuarios registrados en el sistema o usuarios pertenecientes a otros grupos. Los usuarios que han sido invitados al grupo pueden aceptar o rechazar la invitación. También pueden visualizar la información de los grupos a los que pertenecen e indicar si desean salir

de alguno. Además, los grupos permiten crear votaciones en las cuales los miembros del grupo son añadidos directamente.

Los casos de uso asociados con grupos son: Crear grupo, Invitar a grupo, Aceptar invitación a grupo, Rechazar invitación a grupo, Salir de grupo y Ver lista de grupos.

- **Validadores:** un usuario autenticado puede postularse como validador ante el sistema. Este, por medio de un torneo en el que se compara la reputación de cada postulante, escoge los validadores activos del sistema. Asimismo, los perdedores permanecen como validadores inactivos hasta el siguiente torneo o hasta que se retiren de la postulación.

  Cuando se termina el torneo, comienza una etapa en la que los validadores activos insertan los votos que han recogido. Estos votos se insertan en la blockchain correspondiente a la votación, por medio del algoritmo de consenso *Proof of Authority* (PoA).

  Durante el periodo de inserción de votos en la blockchain, los validadores activos reportan automáticamente a aquellos validadores líderes que han propuesto bloques con transacciones que no pertenecen a la cola de transacciones. En este caso, si el sistema detecta que el 60% de los validadores ha reportado a ese líder, este disminuirá su reputación. Al finalizar esta etapa y antes de comenzar el siguiente torneo, el sistema aumentará la reputación de los validadores que no fueron reportados.

  Cuando se solicita el resultado de una votación, los validadores buscan en la blockchain la transacción con los resultados. Si esta transacción no existe, se crea y se añade a la blockchain. En estos resultados se muestra al usuario las personas que votaron y las que no.

  El sistema es capaz de indicar quien votó y quien no, sin necesidad de violar el anonimato de los votantes. Para ello, cuando se crea una votación el sistema almacena en la base de datos a los participantes junto a los votos que tienen disponibles. También

se crea una lista de seudónimos por votación, con la misma cantidad de participantes, quedando almacenados en la base de datos. De esta forma cuando un usuario vota, en la base de datos se le resta el voto de esa votación, se le asigna uno de los seudónimos al voto y el seudónimo se marca como ya usado, sin que se sepa el contenido del voto. Finalmente, el voto y su seudónimo son almacenados en la blockchain.

En caso de que la base de datos permita más votos de los que el creador de la votación indicó, los validadores son capaces de detectar este fallo verificando el número de votos permitidos para esa votación, que se encuentra almacenado en el bloque génesis de la blockchain de esa votación.

Los casos de uso asociados a los validadores son: Proponer bloque, Validar bloque, Realizar conteo de votos, Seleccionar validadores y Agregar bloque a la blockchain.

A continuación, en la Figura 2 se muestra el diagrama de casos de uso del sistema Pocket Ballot Chain.

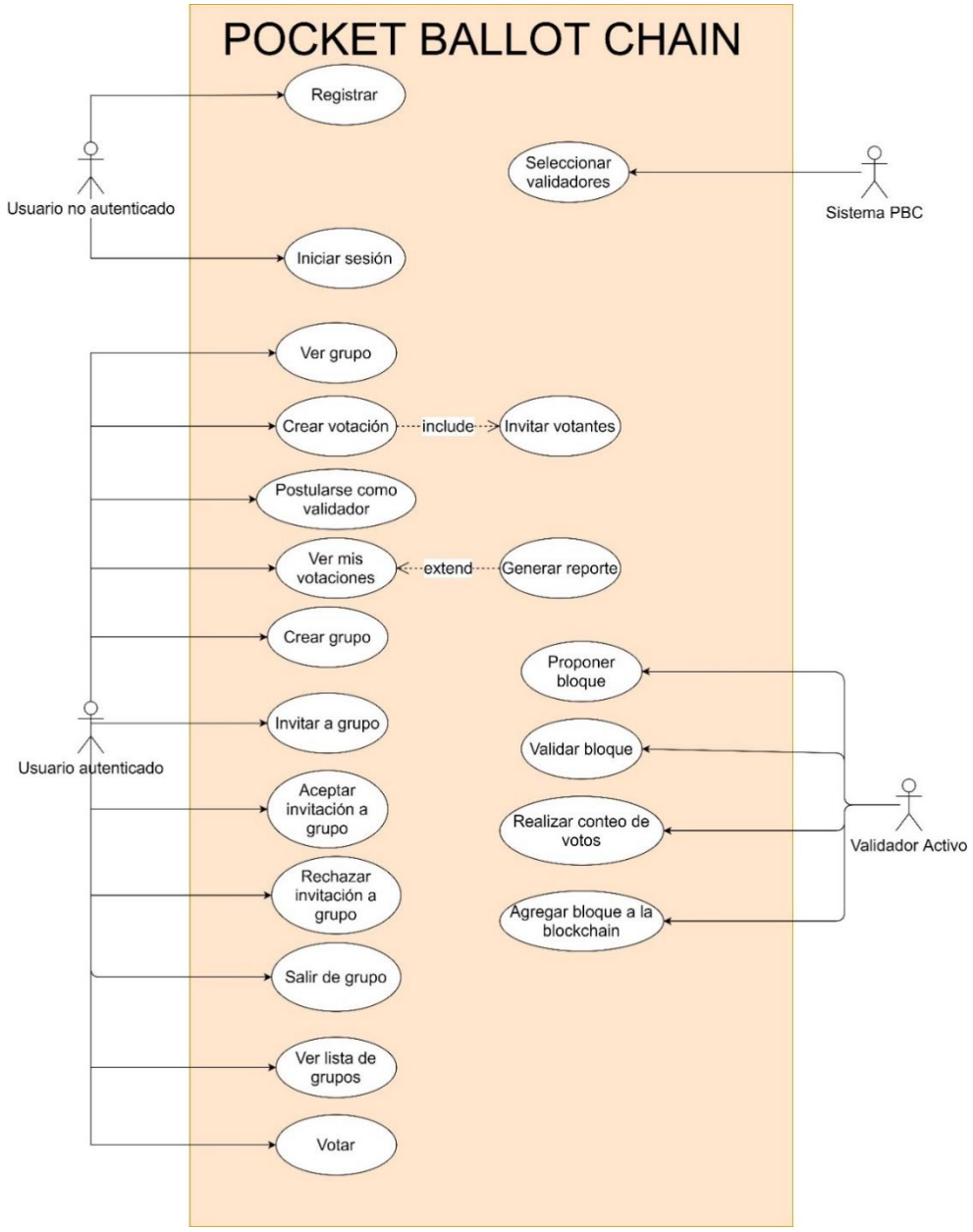

Figura 2. Diagrama de Casos de Uso del sistema Pocket Ballot Chain

V DISEÑO DE LA SOLUCIÓN

En este capítulo se exponen las partes más importantes de la arquitectura del sistema, incluyendo el diagrama de componentes, el diagrama de despliegue, el diagrama de bases de datos y las herramientas fundamentales para el desarrollo y despliegue del proyecto. Todo esto se definió durante el desarrollo del proyecto, específicamente en el SDD y en el SRS. Para más información consultar el anexo SRS_voto_electronico y el anexo SDD_voto_electronico.

1 Principales herramientas

En la Tabla 5 se presentan las principales herramientas utilizadas durante el desarrollo del proyecto.

Tabla 5. Principales herramientas del proyecto

Nombre	Utilidad durante el desarrollo	Justificación
Angular	Este *framework* se utilizó para el desarrollo del cliente web del proyecto. Permitió trabajar: la vista de la página web por medio de *HTML* y *CSS*; la lógica de la blockchain, y la comunicación con el servidor por medio de *Javascript* y *Typescript* [21].	Este *framework* se escogió gracias a que el equipo de desarrollo está familiarizado con el uso de este. Además, en Angular se puede desarrollar páginas web de tipo *Single Page Application*, lo cual facilita que el cliente pueda ejecutar de forma local el componente de la blockchain.
	Git es una herramienta de control de versiones que fue indis-	Todo el equipo de desarrollo está familiarizado con *Github.* Además, el

Git y Github	pensable para el trabajo colaborativo durante la implementación del prototipo. *Github* ofrece un servicio en la nube que guarda los repositorios del *front-end* y *back-end* de la aplicación web [22] [23].	control de versiones compartido es indispensable para el desarrollo de proyectos en los que varias personas estarán trabajando sobre el código simultáneamente.
MySQL	*MySQL* es un manejador de bases de datos relacional desarrollado en C/C + + que ofrece confiabilidad y un tiempo de respuesta rápido para grandes cantidades de datos. Durante el desarrollo del proyecto se utilizó *MySQL* para guardar la información de los usuarios, las votaciones, y los grupos [24].	El equipo ya ha tenido experiencia manejando bases de datos relacionales y otros manejadores de bases de datos con sintaxis similar.
Node.js	*Node.js* es un *framework* de desarrollo que permite implementar aplicaciones web por medio de *Javascript* o a través del instalador de paquetes que tiene por defecto: *npm* que permite utilizar otros *frameworks* como Angular. El servidor del proyecto se hizo exclusivamente utilizando *Node.js*,	*Node.js* provee la versatilidad para instalar una gran variedad de paquetes que agilizó el desarrollo del proyecto.

	mientras que el cliente web se realizó utilizándolo junto con *Angular* [25].	

Teniendo en cuenta que *Node.js* permite instalar paquetes que poseen librerías fundamentales para el proyecto, en la Tabla 6 se expone cuáles son las tres librerías más relevantes para el funcionamiento del sistema:

Tabla 6. Principales librerías del proyecto

Nombre	**Utilidad durante el desarrollo**
Socket.io	Esta librería permitió realizar la comunicación bidireccional entre los validadores de la blockchain y el servidor para la ejecución del torneo que determina cuáles son los validadores activos del sistema. Además, esta librería también se utilizó para comunicar votos entre el cliente web y el servidor [26].
Peerjs	Esta librería hace uso de *WebRTC*, lo cual permite la comunicación *peer-to-peer* entre los validadores y los votantes [27].
js-sha512 y *node-rsa*	Durante el desarrollo del proyecto se utilizó las librerías de *sha-512* y *node-rsa,* para realizar los hashes de la blockchain, cifrar el voto con criptografía asimétrica, realizar las firmas digitales y encriptar la contraseña de los usuarios durante el inicio de sesión y registro [28].

Para encontrar la explicación detallada de las herramientas, revisar la sección 7.1 del SPMP.

2 Distribución general del sistema

Antes de comenzar a explicar los diagramas más importantes de la arquitectura, cabe señalar las principales partes del sistema. En la Figura 3 se muestra una representación a alto nivel del sistema Pocket Ballot Chain.

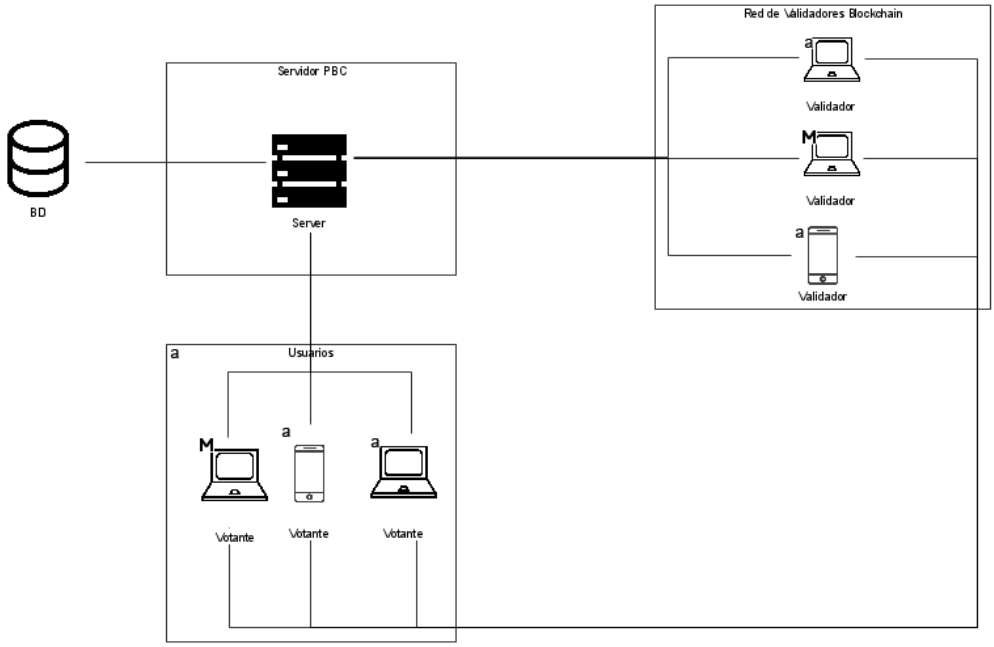

Figura 3. Diagrama de distribución general del sistema

El sistema posee un conjunto de usuarios que se comunican con el servidor para iniciar sesión, crear una votación, crear un grupo o solicitar información a la base de datos. El sistema también posee un conjunto de validadores que reciben los votos de los usuarios. Los validadores se comunican entre sí para proponer bloques nuevos a la blockchain y mantenerla actualizada. También se comunican con el servidor PBC para inscribirse al torneo que determina cuáles nodos fungirán como validadores activos. El servidor PBC procesa las solicitudes provenientes de los usuarios y de los validadores, y se comunica con la base de datos para realizar consultas, persistir información nueva o modificar la información ya existente.

3 Explicación detallada de la blockchain

La blockchain requiere de los validadores para existir en el sistema. Por defecto existen cinco validadores provistos por el equipo de desarrollo. Estos cinco validadores son confiables y siempre serán seleccionados para proponer y validar bloques (a menos de que se desincronicen). Sin embargo, para proveer mayor seguridad a la blockchain y evitar que el control sobre la misma este centralizado, se da espacio a otros cinco usuarios del sistema que son seleccionados como validadores mediante el torneo que realiza el servidor.

El torneo consiste en comparar de dos en dos cuales son los concursantes que tienen mayor reputación, de tal forma que los ganadores sean los validadores con mayor reputación, pero dando posibilidad a los validadores de reputación media. Esto último es importante para construir la reputación de los nuevos validadores sin comprometer la seguridad del sistema.

Cuando un torneo se termina, los ganadores pasan a ser validadores activos y estos se turnan para proponer bloques a partir de las transacciones que vayan recibiendo. En Pocket Ballot Chain hay tres tipos de transacciones: las que comienzan una votación, las que contienen un voto y las que cierran la votación una vez se haya alcanzado la fecha límite. Para disminuir el tiempo de procesamiento de los resultados, se manejan una sub-blockchain por cada votación.

Los validadores proponen, validan e insertan bloques a partir de un algoritmo de consenso basado en el *proof of authority.* En este algoritmo, se le da un espacio de tiempo (*step*) a cada validador activo para proponer sus bloques y que estos sean validados por los demás validadores activos del sistema. Cuando un validador activo va a proponer bloques (máximo un bloque por cada sub-blockchain), este va a seleccionar todas las transacciones que se encuentran en cola y las empaqueta en un bloque por cada votación. Luego las envía a los demás validadores. Cada validador activo valida el bloque propuesto, y a partir de su cola de transacciones decide si aceptar o rechazar el bloque. Si lo acepta, envía su confirmación a los demás validadores y cuando el 60% de los validadores activos hayan aceptado los bloques, estos los insertarán en sus propias sub-blockchains. Una vez realizado esto, se termina un *step* y se pasa al siguiente *step*. En la Figura **4** se muestra de forma gráfica como es el proceso de proponer y

aceptar bloques cuando hay cuatro validadores activos y el validador número 0 es el que va a proponer el bloque.

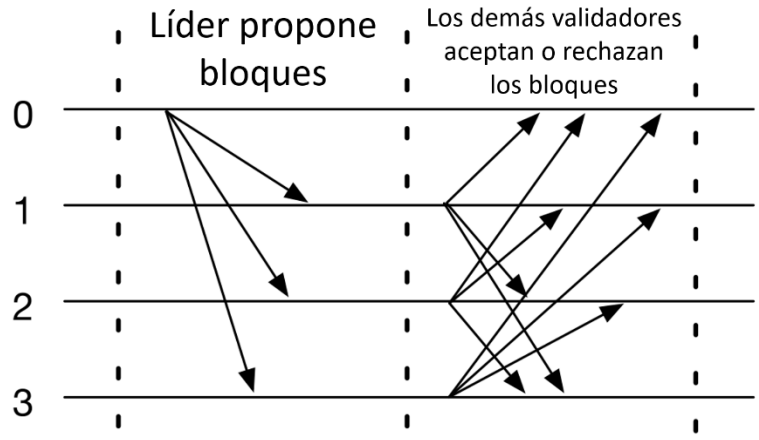

Figura 4. Algoritmo de consenso tomado de [8]

Cuando se hayan terminado todos los *steps*, los validadores activos e inactivos entran en un periodo de sincronización. El periodo de sincronización dura lo mismo que un *step* y en este tramo de tiempo los validadores activos les envían a los validadores inactivos los nuevos bloques insertados a las sub-blockchains. Si existe un validador que recién ha entrado al sistema, este se sincroniza solicitando todos los bloques y transacciones a los validadores activos. Al final de este periodo de sincronización, todos los validadores (activos e inactivos) le confirman al servidor el estado actual de todas sus sub-blockchains enviando un hash que las representa, de tal forma que se seleccionan como concursantes del siguiente torneo a los validadores que enviaron el hash que notificó el 60% de los validadores activos. Después de seleccionar a los concursantes del siguiente torneo se repite el proceso.

El periodo de tiempo que va desde la realización de un torneo, hasta la etapa de sincronización se llama *age* o era. La Figura **5** representa de forma gráfica dos eras en las que participan seis validadores, en la primera era, el torneo lo ganan los validadores V1, V2 y V3 y en el segundo lo ganan los validadores V4, V5 y V6. Los *steps* S1, S2 y S3 representan el espacio de

tiempo en donde uno de los validadores activos propone un bloque y los demás lo validan. Los *steps* S4 representan el periodo de sincronización y los *steps* S0 son un espacio de tiempo en el que todos los validadores del sistema reciben los resultados del torneo anterior y se preparan para ser validadores activos o inactivos.

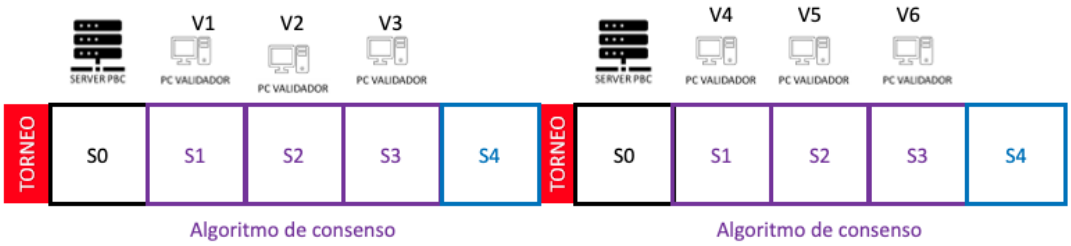

Figura 5. *steps* y *ages* en Pocket Ballot Chain

Para más información sobre el algoritmo de consenso y el torneo, por favor ir a las secciones 6.3 y 7.2 del SDD_voto_electronico.

4 Diagrama de componentes y Diagrama de despliegue

Debido a que el proyecto necesita una base de datos para guardar toda la información que no está relacionada con los votos, se utiliza un servidor que administra las consultas hacia esta. Adicionalmente, es necesario un cliente web que maneje la interfaz de usuario, realice solicitudes hacia el servidor y gestione la blockchain. Para representar ambas partes del sistema se realizó un diagrama de componentes que se encuentra en la Figura 6.

En la parte del cliente web PBC se encuentran los siguientes componentes:

- **Vista Web PBC:** este componente es el encargado de la parte visual de la página web del cliente.
- **Lógica P2P Web PBC:** este componente se encarga de manejar la lógica necesaria para gestionar la red *peer-to-peer* y la blockchain.
- **Servicio Web PBC:** este es el componente se especializa en ofrecer servicios a otros validadores.

- **Integración Web PBC:** este componente consume los servicios que provee el servidor y los validadores del sistema.

En la parte del servidor PBC se encuentran los siguientes componentes:

- **Negocio PBC:** este componente se encarga de procesar las solicitudes que le llegan al servidor por medio del Servicio PBC.
- **Servicio PBC:** este es el componente del servidor que se especializa en ofrecer servicios a todos los usuarios del sistema.
- **Integración Web PBC:** este componente permite la comunicación entre el servidor y la base de datos.

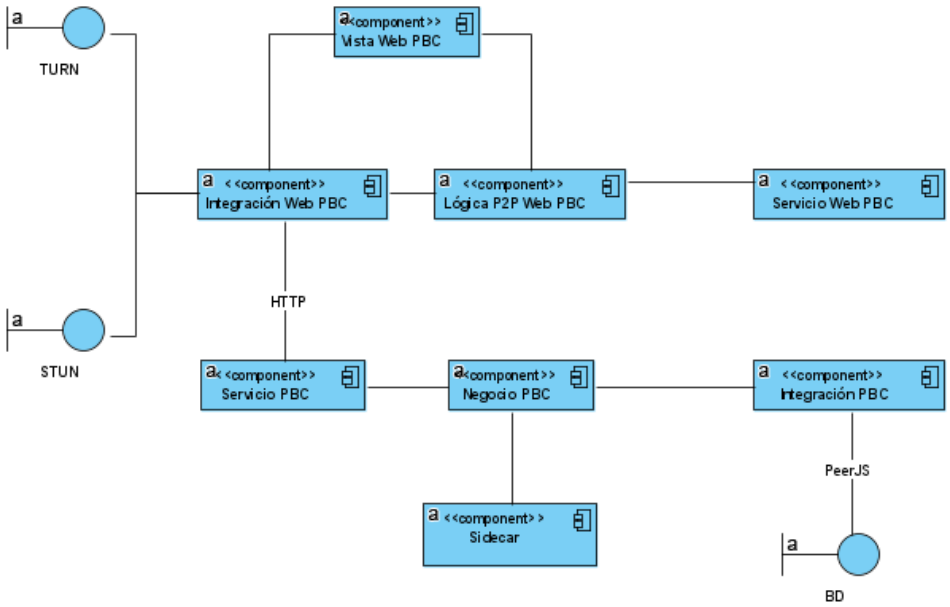

Figura 6. Diagrama de componentes

Además de los componentes, en el diagrama de la Figura 6 también se muestra una representación de los sistemas externos que utiliza Pocket Ballot Chain:

- **BD:** esta es la base de datos en MySQL que guarda toda la información del sistema a excepción de los votos que guarda la blockchain.

- **STUN/TURN:** estos son servicios externos que posibilitan la comunicación *peer-to-peer* en el sistema.

Para la explicación completa del diagrama de componentes, por favor dirigirse a la sección 6.1 del anexo SDD_voto_electronico.

El diagrama de despliegue muestra cómo se distribuyen los componentes en los nodos del sistema. Cabe resaltar que los componentes del cliente web (los que están alojados en los nodos PC Usuario) están replicados para representar la comunicación *peer-to-peer* del sistema durante el proceso de votación.

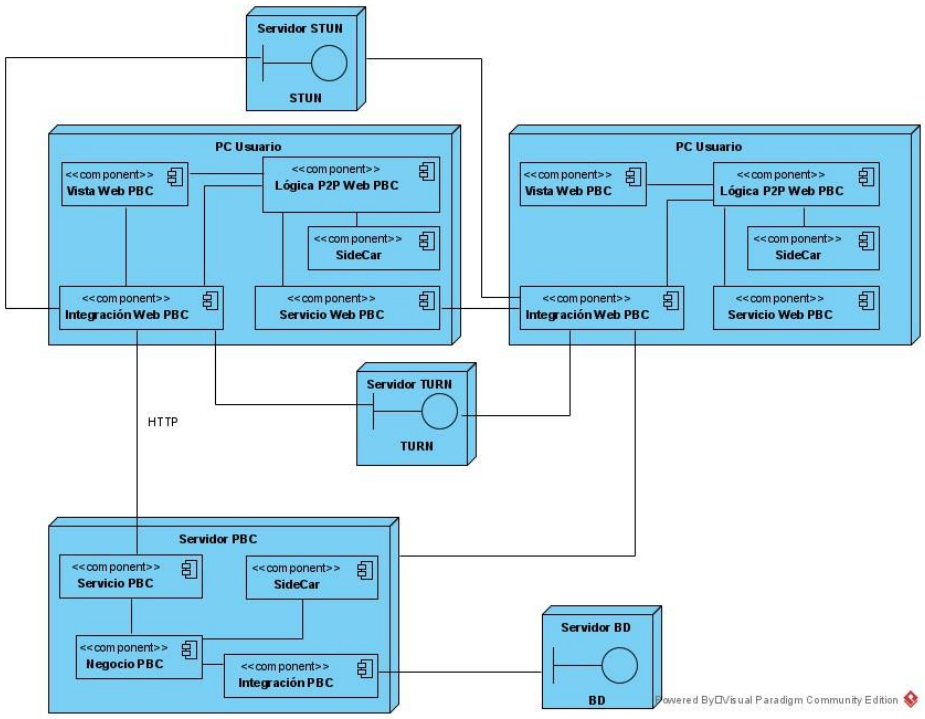

Figura 7. Diagrama de despliegue

Los nodos propios del sistema Pocket Ballot Chain son:

- **Servidor PBC:** este es el dispositivo que se encarga de alojar el servidor del sistema. Este nodo es accesible desde el internet y es el único que se puede comunicar con el nodo Servidor BD.
- **Servidor BD:** este dispositivo aloja la base de datos del sistema, físicamente se encuentra cerca del servidor PBC.
- **PC Usuario:** esto representa los computadores o dispositivos móviles mediante el cual se puede acceder al cliente del sistema. Es necesario que estos nodos contengan un navegador como Firefox, Google Chrome, Safari, Microsoft Edge u Opera con conexión a internet para poder hacer uso de los componentes Web PBC del sistema.

Por último, están los nodos que hospedan el STUN y el TURN que son parte fundamental de la comunicación entre *peers*. Estos nodos son externos al sistema Pocket Ballot Chain, pero cualquier dispositivo usuario se puede comunicar con estos para establecer las conexiones *peer-to-peer.*

Las conexiones *peer-to-peer* en la red de validadores se realiza por medio de la librería *PeerJS* la cual se basa en *WebRTC.* Para realizar esta comunicación es necesario que los validadores importen la librería *PeerJS client*. A partir de esta se crea un objeto *Peer,* sin embargo, para crearlo es necesario contar con un *PeerServer. PeerJS* ofrece un servidor gratuito, no obstante, también provee las instrucciones para ejecutar un servidor propio [27]. Gracias a esto, para este proyecto se implementó un *PeerServer* propio, el cual es el servidor PBC.

El *PeerServer* es el encargado de asignar un *Peer ID* a cada *peer* en la red, en este caso a los validadores. Estos ID pueden ser generados aleatoriamente o ser especificados, siempre que sean únicos. En el momento en que un *peer* quiere conectarse con otro *peer* es necesario que este conozca su *Peer ID* [27]. En el caso de PBC este intercambio de *Peer ID* se da cuando un torneo finaliza y se envía a los ganadores los *Peer ID* de los validadores activos e inactivos.

Cuando se solicita la conexión con otro *peer,* esta solicitud pasa por el *PeerServer,* el cual conoce a todos los *peers.* Esta primera comunicación se conoce como *signaling,* en la cual se coordina cómo se hará la comunicación [29].

PeerJS permite la comunicación entre *peers* incluso si están en distintas redes privadas, para que esto sea posible se hace uso de los servidores STUN y TURN. El servidor STUN permite encontrar la dirección IP pública de una máquina, el tipo de NAT y el puerto asociado con el puerto local a través de la NAT. El servidor TURN es un respaldo en caso de que el STUN falle, además retransmite los mensajes si la conexión *peer-to-peer* falla [29].

Para una explicación más completa de la vista física del sistema, que incluye los requisitos mínimos de hardware y software de cada nodo, por favor dirigirse a la sección 6.2 del SDD_voto_electronico.

VI DESARROLLO DE LA SOLUCIÓN

1 Metodología

1.1 SCRUM

En la versión final de la propuesta de proyecto de grado se determinó que la metodología principal de desarrollo y documentación seria manejada con una versión modificada de *Scrum*. Entre las modificaciones realizadas resalta una mayor libertad con respecto al tiempo de cada uno de los eventos principales de la metodología y la ausencia de los *Daily Scrum*. A continuación, se expone el papel de cada uno de los eventos de *Scrum* durante el desarrollo del proyecto:

Sprint Review: esta reunión se realizaba todos los martes junto con nuestro principal *stakeholder:* Rafael Vicente Páez. En estos espacios se mostró el avance en los documentos y la implementación del proyecto.

Sprint Planning: después de cada *Sprint Review* se tomaba un espacio para revisar: las tareas que habían sido resueltas, las tareas nuevas que se integran al *Product Backlog* (a partir de la retroalimentación del *Sprint Review)* y cuales se iban a desarrollar en el siguiente *Sprint*.

Sprint Retrospective: se realizó una reunión todos los sábados para evaluar el estado actual de la implementación del proyecto. Sin embargo, por la disponibilidad de tiempo de los integrantes, este evento se realizó durante el transcurso de los *sprints* y no al comienzo de estos.

Finalmente, cabe resaltar que se realizaron reuniones esporádicas entre los integrantes del *scrum team* con la finalidad de realizar la implementación y la documentación del proyecto.

2 Medición del trabajo y artefactos

En el *Software Project Management Plan* (SPMP) el grupo especificó los procesos, herramientas y artefactos que utilizaría para el desarrollo del proyecto. Entre estos se incluía métodos y herramientas de estimación, planes de trabajo, descomposición de actividades, medidas del proyecto, reporte del progreso, mecanismos de control para el ambiente de trabajo, manejo

de versiones y control sobre los cambios. Para ver los resultados de estos, consulte el documento anexo de *Postmortem*, el cual contiene tanto los artefactos como el informe final que los analiza para dar un cierre al proyecto. analiza para dar un cierre al proyecto.

En el *postmortem* se menciona como conclusión, que en general el grupo de trabajo hizo uso de la mayoría de los procesos y artefactos que especifico en el SPMP. Sin embargo, se encontró que aquellos procesos, actividades y artefactos que no fueron usados, se debió a la dificultad innecesaria que añadían al proyecto. Tómese como ejemplo el control de cambios de la documentación. Varios de los cambios venían del *Scrum Review,* donde el director de trabajo de grado hacía sus recomendaciones para el grupo. Estos cambios en lugar de llevarlos a diferentes tableros de Trello como se especifica en el SPMP, se decidió únicamente documentarlos en las actas de reunión, para ser discutidos en el *Scrum Planning* por los miembros del grupo.

Con respecto a las estimaciones del trabajo, se concluyó que las estimaciones del grupo no fueron acertadas en varios aspectos, especialmente en relación con la implementación de los requisitos relacionados al funcionamiento de la blockchain del sistema. El grupo identificó que esto se debió a no tener en cuenta algunos problemas y casos alternos para la red *peer-to-peer* de validadores. No obstante, cuando hubo retrasos nunca se pasó de una semana, y los objetivos del proyecto fueron logrados.

3 Producto final

3.1 Cliente web

Autenticación

Inicio de sesión: en el Apéndice Figura 14 se muestra la primera pantalla que todo usuario no autenticado puede ver, el inicio de sesión. El método de autenticación en Pocket Ballot Chain es por medio de un usuario y contraseña registrado previamente en el sistema. En caso de que un usuario no posea ninguna cuenta asociada al sistema, está la opción de “registrarse” que lo redirige a la pantalla de registro.

Registro: esta pantalla (que también se muestra en el Apéndice Figura 14) contiene el formulario que los usuarios necesitan diligenciar para registrarse en el sistema. Es necesario digitar un correo, un nombre de usuario (único en el sistema) y una contraseña que debe ser confirmada en el mismo formulario.

Votación

Lista de votaciones: el Apéndice Figura 15 y Apéndice Figura 16 muestran la pantalla de la lista de votaciones del usuario. En el Apéndice Figura 15 se muestran las votaciones que ha creado el usuario y en el Apéndice Figura 16 se muestran las votaciones en las que el usuario es participante. En esta pantalla es donde se crean votaciones, se vota y se solicitan los resultados.

Crear votación: las pantallas que se muestran a continuación permiten al usuario crear votaciones. En la Figura 17 se muestra la pantalla que permite al usuario seleccionar el tipo de votación a crear, en el caso de la votación popular se le solicitara al usuario ingresar la cantidad de votos para cada votante.

Luego de seleccionar el tipo de votación se le presentará al usuario una pantalla donde podrá colocar todos los detalles de la votación. Esta pantalla se puede visualizar en la Figura 18. Esta información comprende el título de la votación, una descripción, las opciones por las cuales se podrá votar, los participantes, fecha de inicio y fecha de finalización.

Votación popular: en la Figura 19 se muestra la pantalla en la que un votante podrá realizar su voto en una votación de tipo popular. En esta pantalla se muestra una serie de opciones o candidatos en los que un votante puede distribuir libremente la cantidad de votos que el sistema le provea. En caso de distribuir mal los votos en esta pantalla, se pueden restaurar y volver a distribuir a voluntad.

Votación ranking: en la Figura 20 se muestra la pantalla en donde los usuarios pueden emitir su voto de una votación de tipo ranking. En este caso, el usuario debe listar las opciones en orden descendente, siendo la primera posición su preferida y la última la que menos prefiere.

Votación clasificación: en esta pantalla, que se encuentra en la Figura 21 se puede realizar el voto de una votación de tipo clasificación. El usuario tiene dos columnas donde clasificar las opciones de votación: una de opciones aceptadas y otra de opciones rechazadas. El usuario puede mover las opciones de una lista a otra por medio de las flechas que acompañan a cada opción.

Resultados de votación: en las Figura 22, Figura 24 y Figura 25 se muestran la pantalla de resultado para los tres tipos de votaciones (popular, ranking y clasificación). En estas se muestra la información de la votación, los resultados de esta y las personas que fueron invitadas, y si estas votaron o no.

Validación

Validadores: las pantallas que se muestran a continuación corresponden a aquellas que brindan funcionalidad a los usuarios que quieren ser validadores. En la Figura 26 se muestra la pantalla de inicio del cliente web. En esta, un usuario indica que quiere ser validador y lo lleva a la siguiente pantalla de la Figura 26 donde se muestra las ventajas de ser validador en el sistema.

Una vez los usuarios indican que quieren ser validadores en la pantalla de la Figura 26, el cliente web se dirige a la página de la Figura 27. El usuario permanecerá en esta pantalla mientras quiera validar, y el sistema mostrará si este validador es activo o no.

3.2 Servidor y base de datos

A continuación, se muestran algunos de los procesos en los que el servidor cumple un papel de gran importancia, desde los logs que se registran en un estado de ejecución regular.

La comunicación con el servidor se logra a través de tres puertos, cada uno para un propósito diferente. Entre los cuales se encuentran: la recepción de peticiones REST de los clientes (puerto 3000), la comunicación de doble vía entre el servidor y los clientes (puerto 4000), y las solicitudes de conexiones P2P entre los clientes (puerto 5000). En la Figura 8, se evidencia lo anteriormente dicho.

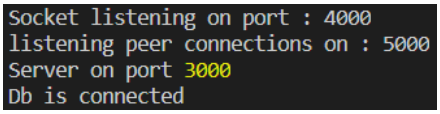

Figura **8**. Puertos habilitados en el servidor

Al finalizar cada era, primero se valida cual instancia de la blockchain es la que más se comparte entre el conjunto de validadores. De esta forma, en la Figura 9 se evidencia el hash de la blockchain local de cada validador. Al final, la blockchain que más se repite será la aceptada para todos los validadores activos en la siguiente era.

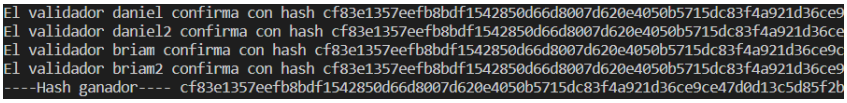

Figura **9**. Ejemplo de validación de blockchain

El servidor elige los validadores activos a partir del torneo. Luego, el servidor imprime en pantalla el estado de los validadores (activo o inactivo). Lo anterior se puede visualizar en la Figura 10, en la cual se muestra un validador activo, dos inactivos y todos los validadores disponibles que entran en el siguiente torneo.

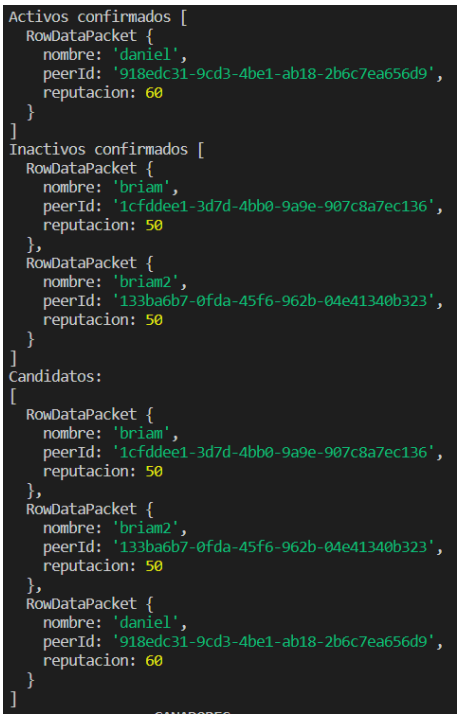

Figura **10**. Ejemplo de estados tras finalizar la era

Después de que finaliza un torneo, se anuncian los ganadores a todos los validadores, como se evidencia en la Figura 11.

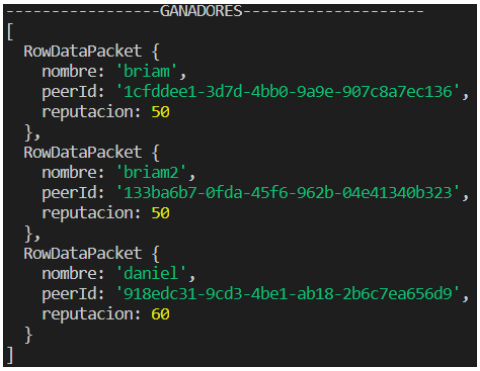

Figura 11. Ejemplo de validadores tras finalizar torneo

3.3 Seguridad

En el apartado de seguridad se recalca la utilización tanto de cifrado como de firmas digitales en las comunicaciones que involucran los votos para asegurar el origen y fiabilidad de la información. Además de esto, se cuenta con el cifrado de información sensible como la contraseña del usuario.

En la parte del despliegue se decidió utilizar *Apache* para lanzar la aplicación en la red, además de esto, se cuenta con una dirección IP obtenida desde la plataforma de *NO-IP* la cual permite adquirir dominios y servidores DNS de manera gratuita. El dominio utilizado para el despliegue es pocketballotchain.webhop.me. También, se utilizó la herramienta *ZeroSSL* que permite generar certificados SSL gratuitos, esto nos permite agregar esta seguridad extra en la capa de transporte (TLS) y en la capa de socket seguro (SSL). Como se puede ver en la Figura 12 se aprecia la vigencia del certificado SSL en la aplicación desarrollada.

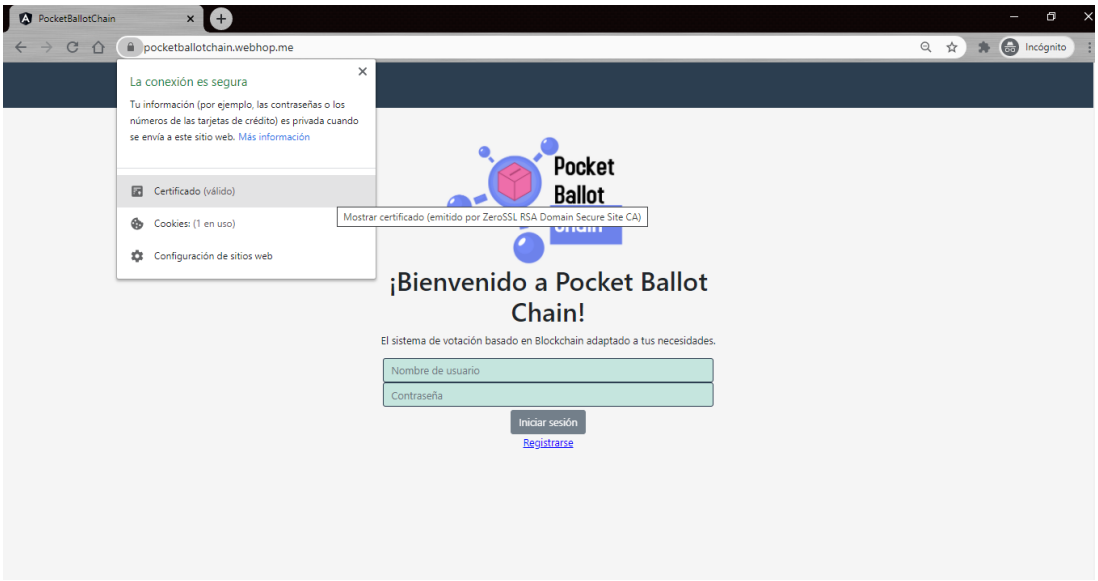

Figura 12. Certificado digital

VII RESULTADOS

Con la finalidad de comprobar la calidad del sistema desarrollado, se realizaron cuatro tipos de pruebas: pruebas de estrés, pruebas funcionales, pruebas de seguridad y pruebas de integración. Para las primeras tres, se creó una red de validadores por medio de cada una de las máquinas de los integrantes del grupo de desarrollo, cada validador estaba en una red privada diferente y hacia uso de internet para comunicarse entre sí. Por otro lado, las pruebas de integración se realizaron en un entorno local por medio de dos máquinas físicas. En la Tabla 7, se muestra los detalles de los computadores utilizados.

Tabla 7. Hardware del equipo de desarrollo

	PC 1 (portátil)	PC 2 (portátil)	PC 3 (portátil)	PC 4 (portátil)
Procesador	Intel i5-4300U 1.90 GHz 2.49 GHz	AMD Ryzen 5 4600H 3.0 GHz 4.0 GHz	Intel i5-4300U 1.90 GHz 2.49 GHz	Intel i5-8250U 1.6 GHz 3.4 GHz
RAM	8 GB	8 GB	8 GB	4 GB
Sistema Operativo	Windows 10 Professional	Windows 10	Windows 10 Ultimate Edition	Windows 10
Arquitectura	64 bits	64 bits	64 bits	64 bits
Velocidad de conexión a internet	6 Mbps	11 Mbps	30 Mbps	47 Mbps

1 Pruebas de integración

Las pruebas de integración del proyecto se encargaron de comprobar la correcta comunicación entre los dos componentes más importantes del sistema: el *backend* y el *frontend*. Estas pruebas se realizaron mediante el envío de solicitudes hacia el servidor utilizando la herramienta *Postman,* que permite realizar peticiones REST. Esta se usó para programar y automatizar las pruebas en cada uno de los *end-points* del servidor. El resultado de dichas pruebas se puede encontrar en los anexos Pruebas de integración.

En los anexos se pueden observar cada uno de los casos de prueba planteados por el grupo para las pruebas de integración. También se puede evidenciar los pantallazos que sirven como evidencia de la realización de cada prueba. En cada evidencia se muestra la URL correspondiente al *end-point* probado y su respuesta. Se considera que cada caso de prueba fue exitoso porque las respuestas esperadas coinciden con la respuesta que muestra *Postman*.

2 Pruebas de estrés

Las pruebas de estrés que se realizaron después del desarrollo del software se centraron en probar la capacidad de los validadores de procesar y guardar las transacciones enviadas por los usuarios del sistema. Para esto, se programó un script que enviaba cierta cantidad de votos seguidos. Estas pruebas se realizaron solamente con los validadores que están por defecto (5 validadores) y pueden variar dependiendo de: la potencia de las maquinas, la calidad de la red y la cantidad de validadores (que en el sistema pueden llegar a variar). Los resultados estas pruebas se encuentran en los anexos Pruebas de estrés.

A partir estos resultados se concluye que en un ambiente en donde solamente se encuentran los validadores por defecto, estos pueden procesar correctamente 100 transacciones en un periodo de tiempo menor a un segundo.

En la Figura 13 se presenta la gráfica que representa la respuesta de los validadores a medida que aumentan las transacciones que se les envía en un periodo de tiempo menor o igual a un segundo. Cada punto representa una prueba realizada en donde se enviaron la cantidad de transacciones correspondientes al eje horizontal de la gráfica. El valor que toma cada punto

en el eje vertical puede ser 1 (si el sistema funciono con normalidad en esas pruebas) y 0 (si alguna de las transacciones no fue guardada). Teniendo en cuenta lo anterior, se puede tomar como ejemplo el tercer punto de la tabla, el cual corresponde a una prueba en donde se enviaron 20 transacciones y los validadores pudieron procesarlas correctamente. Entonces a partir de la gráfica se concluye que los validadores por defecto pueden procesar por lo menos cien transacciones en menos de un segundo.

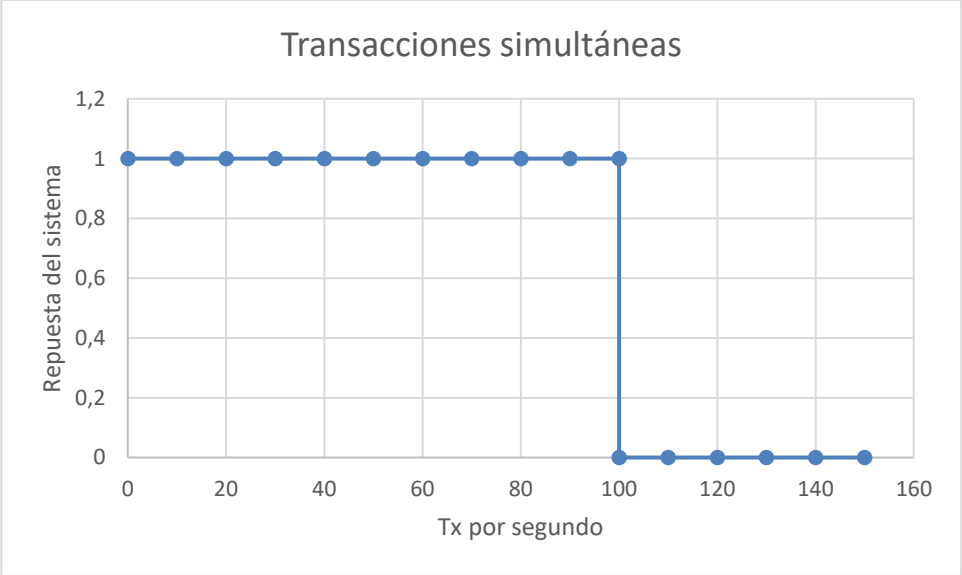

Figura 13. Pruebas de estrés al servidor con transacciones simultáneas

El segundo tipo de pruebas de estrés se realizaron probando la capacidad individual de las máquinas utilizadas, esto con el objetivo de probar su capacidad en el peor de los casos donde sólo una funcione correctamente. Como resultado de estas pruebas se obtuvo que el máximo de validadores que soporta una sola máquina es de ocho validadores. Esta restricción no se debió al consumo de recursos sino a la cantidad de validadores que se pueden ejecutar con los navegadores Chrome, Mozilla, Edge y Opera en su versión estándar e incógnito. De estas pruebas también se concluyó que el cliente web lo que más requiere es memoria RAM. Este resultado es de esperar dado a que el cliente web cuando es validador almacena: transacciones en cola, bloques de cada sub-blockchain, información de las votaciones y mensajes de los demás validadores. Las evidencias de estas pruebas se encuentran en el anexo de Pruebas de

estrés donde se muestra el comportamiento de las máquinas por medio del *task manager* del sistema operativo *Windows.*

3 Pruebas funcionales

Las pruebas funcionales se hicieron desde el *frontend* y su finalidad es probar cada uno de los requisitos funcionales del proyecto ver sección 7.1 del anexo SRS. Estas pruebas se realizaron con el sistema **desplegado,** es decir, que no fue en una red LAN, sino que cada integrante se encontraba en su propia red privada y se comunicaron vía internet. La documentación y los resultados de cada *test-case* se encuentran en el anexo Pruebas funcionales. En la evidencia presentada se reflejan los resultados esperado para de cada uno de los *test-cases.* Y por lo tanto se puede concluir que todos los requisitos funcionales se cumplieron. Además, también se evidencia el correcto funcionamiento de la aplicación web en caminos alternos, por ejemplo, cuando los datos ingresados por el usuario son incorrectos.

4 Pruebas de seguridad

La principal prueba de seguridad consiste en realizar un voto desde el cliente, si este voto es correcto se le asignará un seudónimo y se introducirá en la blockchain, luego se modificará la base de datos para que le asigne el mismo seudónimo a otro voto, cumpliendo estas condiciones los validadores activos de la base de datos deben notificar que la votación fue corrompida desde la base de datos y esta votación no debería ser válida. El objetivo de esta prueba es validar el correcto funcionamiento de los seudónimos y si la blockchain es capaz de notificar este error. La prueba se encuentra documentada en el anexo Pruebas de seguridad. Como se puede observar en el anexo, la pantalla de cada validador notifica al usuario de que la base de datos fue corrupta, por lo que se considera que esta prueba fue exitosa.

VIII CONCLUSIONES

Durante la planeación de este proyecto se realizó una investigación sobre los sistemas de votación electrónicos. Se identificaron variedad de alternativas en el mercado, de las cuales la mayoría presentaba falta de confiabilidad y trazabilidad. Este proyecto nace con el objetivo de presentar una alternativa que solucione estas falencias.

Para mejorar los niveles de confiabilidad en el almacenamiento de votos, se utilizó la tecnología blockchain. Esta tecnología ofrece las propiedades de: inmutabilidad, trazabilidad, descentralización y validación. Estas cualidades son clave en un sistema de votación electrónico y por ello algunos sistemas de votación electrónico lo implementan.

Con base en lo anterior, se planteó que el principal objetivo del sistema implementado fuese desarrollar un sistema de votación electrónico basado en blockchain que permitiera crear tres tipos de votaciones. Este objetivo se logró completamente, dado que se desarrolló la aplicación web Pocket Ballot Chain, la cual permite crear votaciones de tipo popular, clasificación y ranking. Además, el sistema propuesto garantiza el anonimato de los votantes durante las elecciones, asegurando que ninguna parte del proceso de votación se pueda vincular un voto con su votante. También, el sistema cuenta con mecanismos de seguridad como el cifrado asimétrico, para él envió de votos; firmas digitales, para asegurar el origen y autenticidad de cada mensaje, y canales seguros por medio de TLS y SSL.

Pocket Ballot Chain permite la participación de los usuarios durante todo el proceso de construcción y validación de la blockchain, aportando transparencia y descentralización al sistema. Con respecto a la trazabilidad, se implementó un sistema de seudónimos que permite verificar la validez de los votos contrastando la información de la blockchain con la base de datos.

Con respecto a los objetivos específicos del proyecto, todos fueron logrados. Durante el levantamiento de requisitos, se establecieron las principales funcionalidades y atributos de calidad del sistema. En el diseño de la arquitectura del sistema, se establecieron los principales

componentes y nodos del sistema, y se dieron las primeras pautas para el desarrollo por medio de la especificación de los procedimientos más importantes. Una vez planteada la arquitectura, se pasó a la fase de implementación del sistema de votación propuesto, el cual fue desplegado en el siguiente enlace: pocketballotchain.webhop.me. Por último, se realizaron las pruebas y ajustes del sistema para cumplir a cabalidad el diseño y los objetivos propuestos del proyecto.

Una vez finalizado el proyecto, se analizó el producto final con respecto a las principales alternativas disponibles en el mercado y se logra evidenciar como Pocket Ballot Chain presentan ciertas ventajas frente a sus competidores, se analizaron las siguientes características:

1. Cuenta los votos de manera descentralizada garantizando que no es posible monopolizar el conteo de votos.
2. Brinda transparencia a los usuarios permitiéndoles participar en el proceso de validación.
3. Fácil de usar, con diversos tipos de votaciones y con respuestas inmediatas una vez terminada la votación.
4. Almacena votos y votantes y además desvincula cualquier relación entre ellos.
5. Auditable, cuenta con un mecanismo capaz de identificar si los resultados de la votación concuerdan con la información, tanto en el sistema como en la blockchain, permitiendo validar la veracidad de una votación.

Una vez analizados los diversos sistemas se realizó la Tabla 8, en donde se comparan las ventajas de Pocket Ballot Chain con las principales alternativas del mercado.

Tabla 8. Comparación de la solución con respecto a las alternativas

	1	2	3	4	5
P*olys*	Si, utiliza un sistema blockchain para garantizar esto	No, su blockchain es manejada por una sola organizacion	Si, cuenta con varios tipos de votaciones	No se cuenta con información al respecto	No, La plataforma no especifica una manera de lograrlo
eBallot	Si, utiliza un sistema blockchain para garantizar esto	No, su blockchain es manejada por una sola organización	SI, Su sistema es muy personalizable y fácil de usar	No, Es uno de los principales problemas de la plataforma	Si, la plataforma incorpora un mecanismo de auditabilidad
Simply voting	N*o, el sistema es centralizado*	N*o, el sistema es centralizado*	Si, Su nivel de personalización en muy bueno	No se cuenta con información al respecto	No, no tiene un sistema para hacer auditorias
BallotChain	S*i, el sistema usa un blockchain para garantizar esto*	No se cuenta con información al respecto	Si, Su nivel de personalización en muy bueno	No se cuenta con información al respecto	No, no tiene un sistema para hacer auditorias
Proof of Vote	N*o, El sistema es centralizado*	N*o, El sistema es centralizado*	No, El sistema prioriza la seguridad ante la usabilidad	Si, El sistema no vincula votos y votantes al almacenarlos, a pesar de esto si los vincula mientras se realiza la votación.	Si, la plataforma presta un sistema de auditorias

Sistema Propuesto (Pocket Ballot Chain)	SI, *el sistema usa un blockchain para garantizar esto*	SI, el sistema permite contar los votos a la comunidad de votantes haciendo el proceso transparente	SI, Presenta varios tipos de votación y resultados instantáneos una vez finalizan las votaciones	Si, el sistema no vincula los votos con los votantes en ningún punto de la votación ni posteriormente a la mimas.	Si, gracias a la utilización de una base de datos y una blockchain se cuenta con un sistema para realizar auditorías a las votaciones

1 Análisis de impacto del proyecto

En el corto plazo, el proyecto ofrece un sistema de votación electrónico basado en Blockchain, en donde se busca asegurar la privacidad y la exactitud en los procesos de votación. De esta manera, se abordan dos características de gran importancia para la aceptación de los sistemas de voto electrónico[30]. Lo cual provoca un incremento en la participación y en el número de votaciones que se llevan a cabo de esta forma, sobre todo en aquellos contextos donde se presenta un alto grado de desconfianza en los votantes y organizadores [30].

A mediano plazo, tras un incremento en el número de votaciones electrónicas, se espera un aumento del interés en el desarrollo de soluciones que hagan uso de la tecnología Blockchain. También, se espera que este sistema pueda contribuir al desarrollo de soluciones para otros sectores (e. g. banca, salud, agro, etc.) siempre y cuando estas soluciones requieran de mecanismos similares a los diseñados para este proyecto. Por ejemplo, el envío anónimo de transacciones sin intermediarios.

A largo plazo, luego de una normalización en el uso de la votación electrónica basada en Blockchain, y el uso de esta tecnología para apoyar soluciones tecnológicas en otros sectores, se

continuaría con votaciones gubernamentales en países como Colombia. Además, estas soluciones se pueden complementar con: cedula electrónica, sistemas de autenticación biométrica, e incluso asistentes virtuales.

2 Trabajo Futuro

Se debe tener en cuenta que el desarrollo del proyecto se realizó en un tiempo limitado en el cual el alcance debía ser bien definido para lograr los objetivos planteados, a pesar de esto, el proyecto puede expandirse a futuro con nuevas funcionalidades. Algunas de las funcionalidades que se pensaron para este proyecto, pero que no se incluyeron en el diseño ni en el desarrollo de este son:

1. **Método para realizar auditorías accesibles al usuario:** actualmente el proyecto permite realizar auditorías, esto se lleva a cabo por una persona con acceso al servidor y la base de datos del sistema, desde allí se puede visualizar la información sobre los seudónimos almacenados en la blockchain y se puede contrastar con la base de datos. A futuro se puede implementar una interfaz en el *front-end* que les permita a determinados usuarios ver la información relevante de la base de datos y de la blockchain, y realizar la auditoria de una manera accesible.
2. **Delegar votos a un representante:** en algunos procesos de votación puede ser interesante que un usuario delegue su derecho a votar a otra persona. Este proceso es común en contextos donde se deben realizar votaciones electrónicas de forma presencial (como en las juntas de accionistas) y no todos los votantes pueden asistir. Sin embargo, para llevar a cabo este proceso se debe diseñar un mecanismo que permita garantizar que el servidor no pueda manipular el representante seleccionado, sin perder la privacidad de los votantes que ofrece actualmente el sistema. Por ejemplo, un votante podría pasar un código secreto a su representante, el cual sería guardado en la blockchain y que sólo sería conocido por ellos, de esta forma si el servidor modifica el representante, en la blockchain habría evidencia de esto.

3. **Modelo de negocio y sistema de recompensas para los validadores:** llegados a esto punto es bien conocido que el sistema desarrollado se basa en una arquitectura distribuida basada en blockchain, en donde los validadores se encargan de agregar los bloques a esta, por ello es importante que en todo momento se encuentren validadores activos en el sistema. Para incentivar a los usuarios a ser validadores se debe plantear un sistema de recompensas. Uno ejemplo de sistema de recompensas podría corresponder a la implementación de un modelo de negocios para el sistema desarrollado, el cual no solo soluciona el problema de los incentivos anteriormente mencionados, sino que lograría transformar el proyecto a futuro en una alternativa comercial.
4. **Proyección a mayor escala:** como se vio en las pruebas de estrés el sistema actualmente cuenta con la capacidad de procesar hasta 100 votos simultáneamente, esto es una cifra significativa para los objetivos planteados, a pesar de esto se considera que el proyecto es fácilmente escalable tanto con estrategias de *scale-up* como de *scale-out.* Si estas estrategias se aplicaran, se podría plantear la utilización de este sistema en ámbitos mucho mayores como votaciones a niveles regionales o en elecciones gubernamentales. Sin embargo, también sería necesario tener en cuenta nuevos tipos de votaciones o restricciones legales.
5. **Ampliar las pruebas de seguridad**: El sistema Pocket Ballot Chain cuenta con varias medidas de seguridad. A pesar de esto, las pruebas realizadas al sistema no son equivalentes a la seguridad ofrecida, por ello para el trabajo futuro se recomienda realizar pruebas de seguridad extra. Se recomienda revisar los estándares de verificación de seguridad OWASP. En donde se establecen diez de las vulnerabilidades más comunes en sitios web de internet, las cuales son: inyección SQL, Secuencia de comandos en sitios cruzados, perdida de datos de sesión, referencias inseguras a objetos, falsificación de peticiones, almacenamiento criptográfico inseguro, mala configuración de seguridad, controles de acceso a la URL, mala protección en la capa de transporte y redirecciones no válidos [31][32].

De las vulnerabilidades anteriormente mencionadas el sistema cuenta con mecanismos para solventar la mayoría de estos problemas. Sin embargo, no se ejecutaron pruebas dado el limitado tiempo de ejecución del proyecto [31][32].

IX REFERENCIAS

[1] Kung-E Cheng and Fadi P. Deek, “Voting Methods and Information Exchange in Group Support Systems.,” 2006. [Online]. Available: https://www.researchgate.net/publication/220889747_Voting_Methods_and_Information_Exchange_in_Group_Support_Systems. [Accessed: 23-Apr-2020].

[2] IDEA, *Introducing Electronic Voting: Essential Considerations*. 2011.

[3] M. Ilaria Lunesu, F. Eros Pani, A. Pinna, and F. Fusco, “Crypto-voting, a Blockchain based e-Voting System,” *researchgate.net*, 2018, doi: 10.5220/0006962102230227.

[4] C. A. Ribon, J. M. Leon, and O. F. Corredor, “Design of an Electronic Voting System Using a Blockchain Network.”

[5] B. Singhal, G. Dhameja, and P. S. Panda, *Beginning Blockchain : a beginner’s guide to building Blockchain solutions*. New York, NY : Apress, 2018.

[6] U. N. A. O. D. E. Renovar, L. A. Realizacion, and D. E. L. A. S. Elecciones, “La pandemia de covid-19, una oportunidad de renovar la realizacion de las elecciones *,” pp. 135–143, 2020.

[7] A. Lewis, “A gentle introduction to blockchain Technology,” *Bits Blocks*, pp. 1–13, 2015, doi: 10.1017/CBO9781107415324.004.

[8] S. De Angelis, G. Zanfino, L. Aniello, F. Lombardi, and V. Sassone, “Blockchain and cybersecurity : a taxonomic approach Introduction,” no. October, pp. 1–22, 2019.

[9] S. S. Al-Riyami and K. G. Paterson, “Certificateless public key cryptography,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2894, pp. 452–473, 2003, doi: 10.1007/978-3-540-40061-5_29.

[10] W. Zheng, Z. Zheng, X. Chen, K. Dai, P. Li, and R. Chen, “NutBaaS: A Blockchain-As-A-Service Platform,” *IEEE Access*, vol. 7, pp. 134422–134433, 2019, doi:

10.1109/ACCESS.2019.2941905.

[11] K. Schwaber and J. Sutherland, “The Scrum Guide: The Definitive The Rules of the Game,” *Scrum.Org and ScrumInc*, no. November, p. 19, 2017, doi: 10.1053/j.jrn.2009.08.012.

[12] M. L. Heilig, “United States Patent Office,” *ACM SIGGRAPH Comput. Graph.*, vol. 28, no. 2, pp. 131–134, 1994, doi: 10.1145/178951.178972.

[13] C. of Europe, *E-voting handbook: Key steps in the implementation of e-enabled elections*. 2010.

[14] “Polys Help Center.” [Online]. Available: https://docs.polys.me/en/. [Accessed: 23-Apr-2020].

[15] “Capabilities | eBallot.” [Online]. Available: https://www.eballot.com/capabilities. [Accessed: 23-Apr-2020].

[16] SimplyVoting, “Election Checklist,” vol. 1, no. 800. .

[17] P. M. Becker, Matt, Lauren Chandler, Patrick Hayes, Wesley Hedrick, Kurtis Jensen, Srini Kandikattu, “Proof of Vote: An end-to-end verifiable digital voting protocol using distributed ledger technology (blockchain),” 2018.

[18] B. Bruegge and E. Riedel, *A geographic environmental modeling system: Towards an object-oriented framework*, vol. 821 LNCS. 1994.

[19] I. Sommerville *et al.*, *Ingeniería de Software*. 2017.

[20] P. L. Ferdinandi, *A requirements pattern: succeeding in the Internet economy*. Addison-Wesley, 2002.

[21] Google, “Introduction to Angular concepts,” *Documentation*. 2020.

[22] “Github About.” [Online]. Available: https://github.com/about.

[23] “Git About.” [Online]. Available: https://git-scm.com/about.

[24] L. Alberto, C. Santillán, M. Gibert, G. Óscar, and P. Mora, “Bases de datos en MySQL.”

[25] B. Dayley, *Node.js, MongoDB, and AngularJS Web Development - Brad Dayley - Google Libros*. 2014.

[26] T. Cadenhead, *Socket.IO Cookbook - Tyson Cadenhead - Google Libros*. 2015.

[27] “PeerJS - Simple peer-to-peer with WebRTC.” [Online]. Available: https://peerjs.com/index.html. [Accessed: 21-Nov-2020].

[28] “js-sha512 - npm.” [Online]. Available: https://www.npmjs.com/package/js-sha512. [Accessed: 21-Nov-2020].

[29] “WebRTC in the real world: STUN, TURN and signaling - HTML5 Rocks.” .

[30] Y. Yao and L. Murphy, “Remote electronic voting systems: an exploration of voters’ perceptions and intention to use,” *Eur. J. Inf. Syst.*, vol. 16, no. 2, pp. 106–120, Apr. 2007, doi: 10.1057/palgrave.ejis.3000672.

[31] OWASP Fundation, “OWASP Application Security Verification Standard.” [Online]. Available: https://owasp.org/www-project-application-security-verification-standard/. [Accessed: 02-Dec-2020].

[32] OWASP Fundation, “Application Security Verification Standard 4.0 Final,” Mar. 2019.

X ANEXOS

Anexo 1: SPMP

Anexo 2: SRS

Anexo 3: SDD

Anexo 4: Pruebas

Anexo 5: Manual de usuario

Anexo 6: Postmortem

XI APÉNDICES

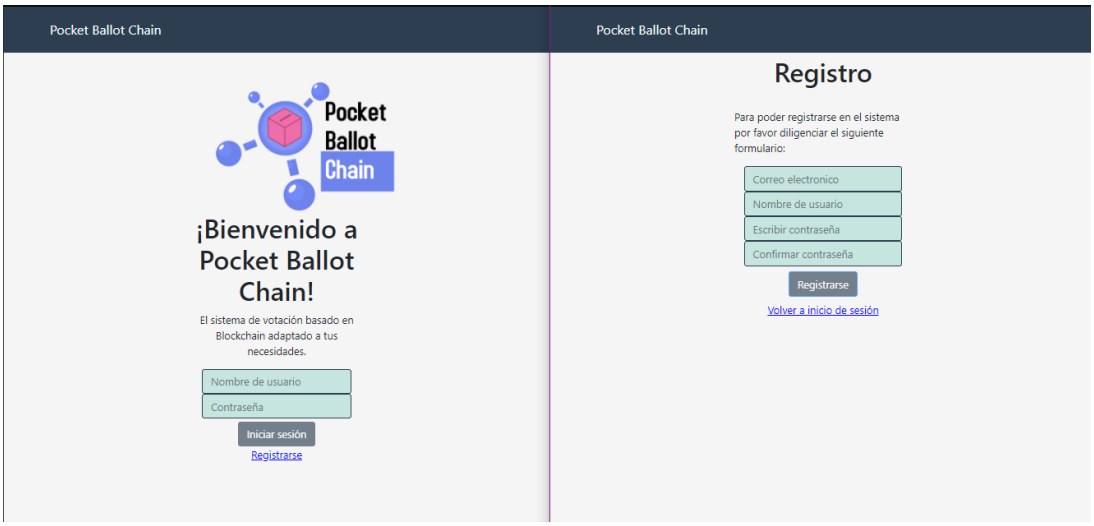

Figura 14. Pantalla inicio de sesión y pantalla registro

Figura 15. Pantalla mis votaciones, propias

Figura 16. Pantalla mis votaciones, participante

Figura 17. Pantalla selección tipo de votación

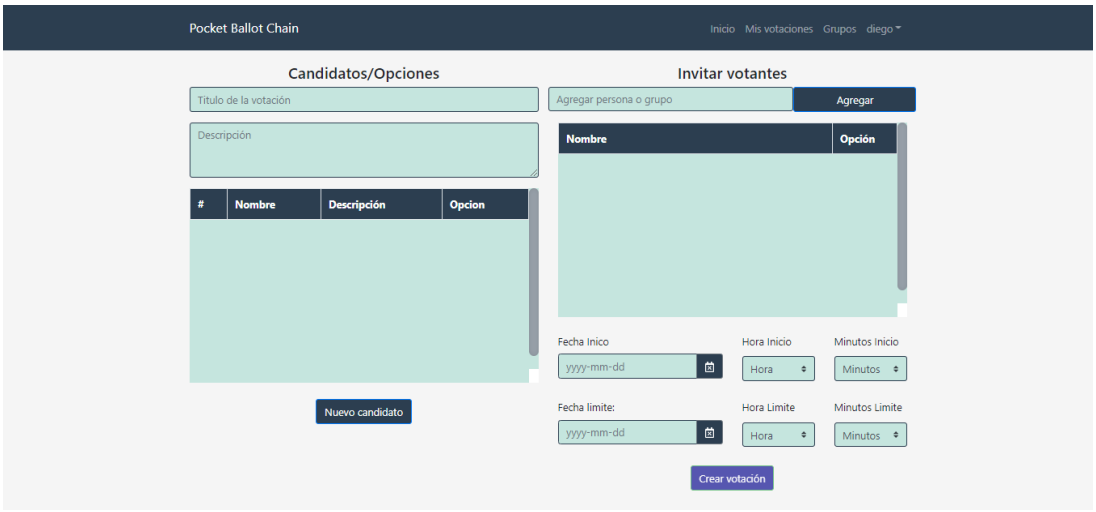

Figura 18. Pantalla crear votación

Figura 19. Pantalla votación popular

Figura 20. Pantalla de votación por ranking

Figura 21. Pantalla votación por clasificación

Figura 22. Pantalla de resultados de votación popular

Figura 23. Pantalla de resultados de votación popular (votantes invitados)

Figura 24. Pantalla de resultados de votación por ranking

Figura 25. Pantalla de resultados de votación clasificación

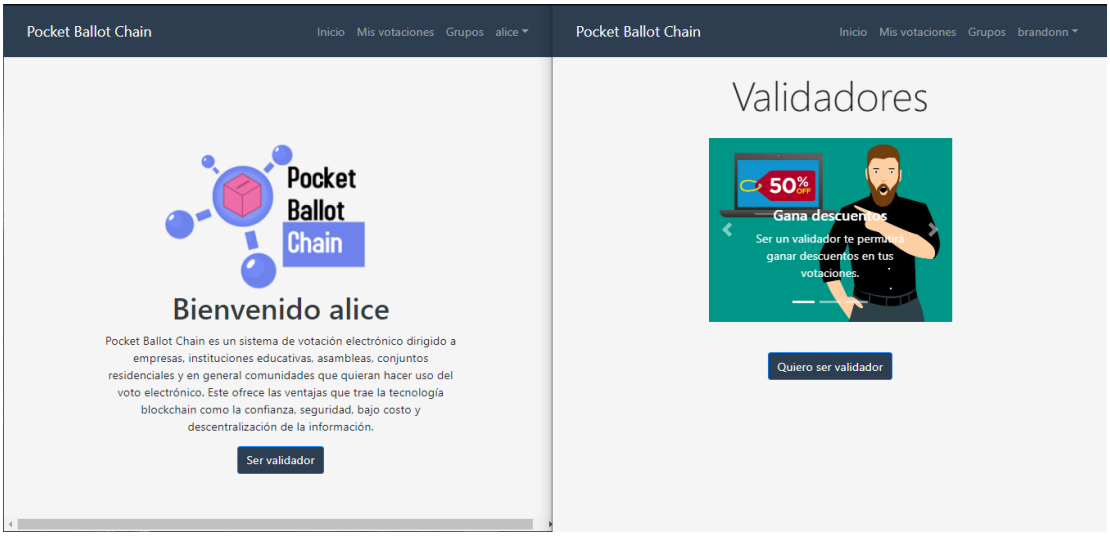

Figura 26. Pantalla de inicio y pantalla para postularse como validador

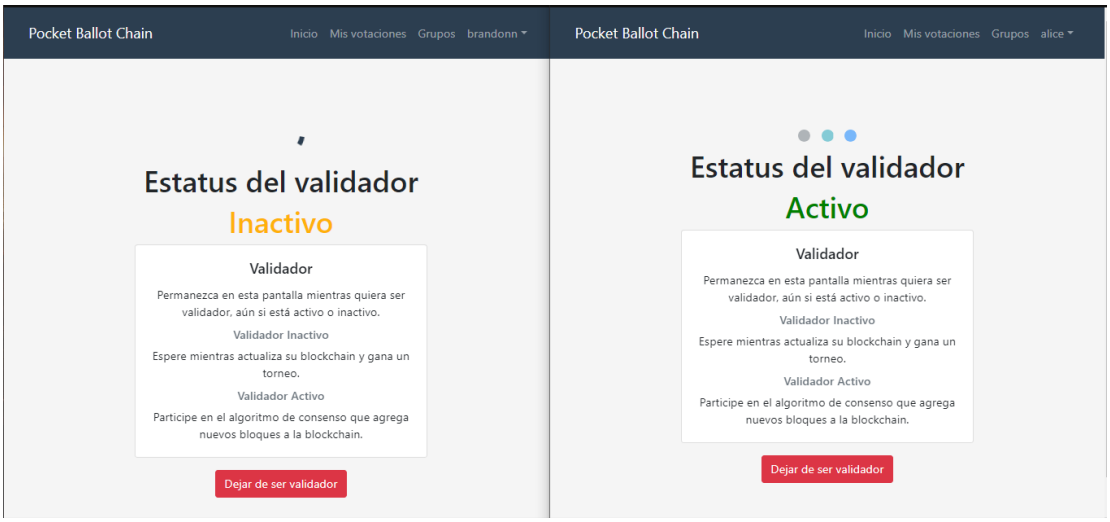

Figura 27. Pantalla de validación