

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Centro de Computación Paralela y Distribuida

Automatización del Subproceso de Votación para Elecciones Estudiantiles de la UCV usando la Tecnología Blockchain NEM

Trabajo Especial de Grado presentado ante la Ilustre

Universidad Central de Venezuela

por el Bachiller

Jose Francisco Iannini Rojas

C.I. 24.276.962

Tutor: Prof. Robinson Samuel Rivas Suárez

Caracas, 24 de mayo de 2019

ACTA

Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado titulado **Automatización del subproceso de votación para elecciones estudiantiles de la UCV usando la tecnología blockchain NEM**, presentado por el Bachiller José Francisco Iannini Rojas (C.I. V-24.276.962), a los fines de optar al título de Licenciado en Computación, dejamos constancia de lo siguiente:

Leído como fue dicho trabajo, por cada uno de los miembros del jurado, se fijó el día 24 de mayo de 2019, a las 11:00 am horas, para que el autor lo defendiera en forma pública, lo que este hizo en SALA 1 de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondió a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobar el trabajo con la nota de 18 puntos.

En fe de lo cual se levanta la presente Acta, en Caracas el día 24 de mayo de 2019.

Prof. Rivas Suárez, Robinson S
Tutor Firmante

Prof. Morales Bezeira, Ana V
Jurado Principal

Prof. Astor Romero, Miguel A
Jurado Suplente

REPUBLICA DE VENEZUELA
Facultad de Ciencias
Escuela de Computación
Universidad Central - Caracas



Universidad Central de Venezuela

Facultad de Ciencias

Escuela de Computación

Centro de Computación Paralela y Distribuida

Automatizar el Subproceso de Votación en elecciones de la Universidad Central de Venezuela con la Tecnología Blockchain

Autor:

José Francisco Iannini Rojas, C.I. 24.276.962, Iannini.rojas@gmail.com

Tutor:

Prof. Robinson Rivas, Robinson.rivas@ciens.ucv.ve

Fecha: 24/05/2019

RESUMEN

En los últimos años las criptomonedas han dado de que hablar en todo el mundo, un sistema financiero descentralizado en el que se garantiza la seguridad, integridad y equilibrio de los estados de cuenta, permitiendo realizar transacciones financieras sin necesidad de un ente regulador. Más allá de la revolución en el sector financiero que puede traer este sistema, el verdadero potencial se encuentra en la tecnología que lo sustenta "Cadena de Bloques" (*Blockchain*). En pocas palabras *blockchain* es un libro de contabilidad digital que se distribuye entre varias ubicaciones para garantizar la seguridad y facilidad de acceso a nivel mundial, permitiendo a consumidores y proveedores conectarse directamente, eliminando la necesidad de un tercero. La tecnología puede funcionar para casi cualquier tipo de transacción que involucre valor, incluidos dinero, bienes y propiedades. En esta ocasión el caso de uso será el voto electrónico, para eliminar la centralización del sistema electoral en la Universidad Central de Venezuela, logrando así un sistema más transparente, seguro y eficiente.

Palabras Clave: Votación con Blockchain, sistema electrónico de votación, Blockchain, auditoría en votación, comisión electoral UCV, proceso electoral.

Índice

Ilustraciones

Introducción

En un sistema democrático el sufragio es la piedra angular en un proceso de elección, la población tiene el derecho de poder expresar su voluntad de elegir y el Estado tiene el deber de garantizar que este proceso sea universal, libre, igual, directo y secreto. Sin embargo, cumplir con esas premisas conlleva a un enorme esfuerzo por parte del estado, tanto económico, ya que se debe invertir una gran cantidad de dinero para que el proceso electoral se pueda realizar, como logístico, movilizar e instruir a una gran cantidad de personas para que sean garantes del correcto cumplimiento. Y en ocasiones les es imposible llevar a cabo un proceso electoral con normalidad [1].

Los sistemas de votación electrónica han sido de interés en los últimos años, el primero en implementarlo fue Estonia [2]. Este método de votación se estructura en técnicas criptográficas, puede aumentar la eficacia y eficiencia del proceso de votación realizando un conteo automático y de forma anónima. En comparación con la votación tradicional, el voto electrónico es un sistema más económico que aborda la transparencia e imparcialidad [3]. Se han propuesto varios métodos para sistemas de votación transparentes [4] [5] [6], pero estos no se han implementado a gran escala. Se ha llevado a cabo varios programas [2] [7], aunque los sistemas de votación electrónica han estado plagados de problemas de seguridad y controversia [8] [9] [7].

A pesar de estas preocupaciones, el voto electrónico y remoto continúa desarrollándose. A medida que una mayor cantidad de la población utiliza Internet regularmente, el voto a distancia y electrónico se convierte en un incentivo para una mayor participación en la democracia. En este trabajo se discute los criterios de votación electrónica, y cómo se puede usar blockchain como un método transparente y rentable para administrar y verificar las transacciones en la votación a gran escala.

Capítulo I - Planteamiento del Problema

1. Situación Actual

La Universidad Central de Venezuela en el derecho pleno que le otorga su autonomía, tiene la potestad de realizar elecciones justas, libres y transparentes. Actualmente varios de los subprocesos que conforman el evento electoral son realizados de forma manual, lo que conlleva a un desarrollo ineficiente del proceso, considerando que en la actualidad existen diversas tecnologías que podrían ayudar a incrementarlo y mejorarlo, así como también disminuir los costos en insumos y recurso humano.

Uno de los subprocesos del sistema de elecciones que actualmente se realiza de forma manual es el de votación, en el cual la comisión electoral imprime planillas con los nombres de los candidatos postulados, seguidamente el elector marca con un símbolo preestablecido el o los candidatos de su preferencia sobre la planilla, estas son almacenadas en un contenedor debidamente sellado e identificado y resguardado por un grupo de personas calificadas por la comisión electoral. Luego de que toda la población ejerza su derecho al voto, el contenedor es transportado hacia el centro de totalización donde los votos serán contados con una máquina lectora de planillas.

El formato manual con el que se realiza la votación podría presentar una serie de inconvenientes en el proceso. El resguardo de los contenedores donde se depositan los votos es uno de los problemas más graves, ya que, si se llegase a extraviar el contenedor o algún voto, esta información se perdería puesto que no existe un respaldo. Las máquinas lectoras que cuentan los votos representan un enorme gasto para la Universidad, ya que estas deben ser alquiladas a entes externos para su funcionamiento, además no se garantiza que todos los votos sean contados por las maquinas ya que pueden presentar inconvenientes. Las planillas también representan un gasto para la universidad, la crisis económica que vive el país y la universidad hace casi imposible contar con presupuesto o establecimientos donde se pueda imprimir las planillas. Contar con el apoyo de recurso humano para ser garante del proceso también representa un problema para la comisión electoral, puesto que se deben encontrar personas responsables y comprometidas dispuestas a recibir entrenamiento para los comicios.

2. Solución Propuesta

Las elecciones juegan un papel fundamental en la sociedad, desde elegir una junta administrativa en una compañía, hasta elegir el presidente de un país. La Universidad Central de Venezuela entra en ese ámbito, ya que se realizan muchas elecciones en ella, desde elegir representación estudiantil, profesoral, directores de escuelas, decanos de facultad hasta rector de universidad, entro otros.

El festival electoral utiliza un sistema centralizado donde la universidad es el único ente encargado del proceso empleando un anticuado voto manual para llevarse a cabo. Es necesario dar un paso al frente y renovar el sistema de votación con uno más seguro, escalable, eficiente y eficaz.

Este trabajo plantea la implementación de un sistema de voto electrónico sustentando en la tecnología *blockchain*, esta no es más que un libro de registro compartido con todos los usuarios de una red, inmutable que contiene la historia completa de todas las transacciones que se han ejecutado. Se puede sustentar un sistema de voto electrónico con esta tecnología ya que garantiza el anonimato y la seguridad con algoritmos criptográficos. Es posible ejercer el derecho al voto desde cualquier dispositivo conectado a internet, logrando aumentar la participación del padrón electoral.

Un sistema de voto electrónico con tecnología *blockchain* lograría disminuir los problemas que conllevan una votación con sistema manual, logrando incrementar la confianza en el electorado.

3. Alcance

El alcance del sistema propuesto en este trabajo especial de grado se basa en crear un sistema de votación utilizando la tecnología *blockchain*, asegurando su funcionalidad y operatividad, en el que se podrá crear eventos electorales con sus respectivas elecciones, registrar padrón electoral, registrar planchas y candidatos, realizar el acto de sufragar y finalmente obtener los resultados para cada persona postulada.

4. Objetivo General

Desarrollar un sistema de voto electrónico con la tecnología *blockchain* para automatizar el subproceso de votación de elecciones de la Universidad Central de Venezuela.

5. Objetivos Específicos

En esta sección se plantean los objetivos necesarios para lograr cumplir la meta del trabajo de investigación.

5.1. Analizar el sistema de votación de la Universidad Central de Venezuela.

5.2. Diseñar una propuesta de arquitectura de voto electronico.

5.3. Relacionar la tecnología *blockchain* con el voto electrónico.

5.4. Implementar sistema de votación electrónico con base en la tecnología *blockchain NEM.*

5.5. Desplegar sistema de votación en ambiente de producción.

5.6. Realizar pruebas de funcionamiento.

Capitulo II – Marco Teórico

En este capítulo se conceptualiza las bases teóricas para llevar a cabo este trabajo de investigación. Se fundamentará en 3 secciones descritas a continuación.

Una primera sección en la cual se hace una descripción general de la Comisión Electoral de la Universidad Central de Venezuela (UCV), se tocarán aspectos como su historia, su ubicación en la estructura organizativa de la UCV y sus objetivos principales. Además, se describirá los procesos involucrados en la Comisión Electoral, esto con la finalidad de que sirva como base para la justificación de este trabajo. Se profundizará en el proceso de votación el cual es el tema central de esta investigación. A continuación, se detallará la tecnología *blockchain*, historia de esta tecnología, así como su definición, funcionamiento, tipos y casos de uso. El último punto abarcara los sistemas de votación electrónicas. Antecedentes, cómo funciona, su arquitectura, ventajas y desventajas de su uso. Luego, se estudiará el uso de *blockchain* en estos sistemas con el fin de lograr una base teórica para el sistema a desarrollar.

1. Elecciones UCV

1.1. Comisión Electoral

1.1.1. Perfil de la Comisión Electoral

La Comisión Electoral de la Universidad Central de Venezuela es un ente adscrito al Consejo Universitario y fue creado a través de la Ley de Universidades de fecha 5 de diciembre de 1958. Esta Comisión "tiene a su cargo la organización y gestión de los diversos procesos de elecciones universitarias" [10] que se desarrollan en la UCV. Como su Visión lo describe:

> "Esta Comisión tiene como alcance prospectivo estar a la vanguardia de nuevas tecnologías en materia Electoral, de manera que las elecciones realizadas por la Institución, se ajusten a las normas pautadas para su ejecución y garanticen validez, confiabilidad y exactitud en sus resultados. Esto porque entre los propósitos de esta Comisión está el de generar confianza en el sistema democrático y que los electores crean en los procesos electorales organizados y ejecutados en esta Dependencia." [10].

1.1.2. **Estructura Organizativa**

La comisión electoral estará integrada por tres profesores de diferentes Facultades designados por el Consejo Universitario, un alumno regular designado por los representantes de los alumnos ante los Consejos de Facultad y un egresado designado por los representantes de los egresados ante los Consejos de Facultad. Los suplentes serán designados en la misma forma y oportunidad de los principales y serán convocados, cuando fuere necesario en el orden en que aparecen en las respectivas listas [10]. Ilustración 1 Estructura Organizativa de la Comisión Electoral UCV la Ilustración 1 se detalla la estructura organizativa de la Comisión Electoral.

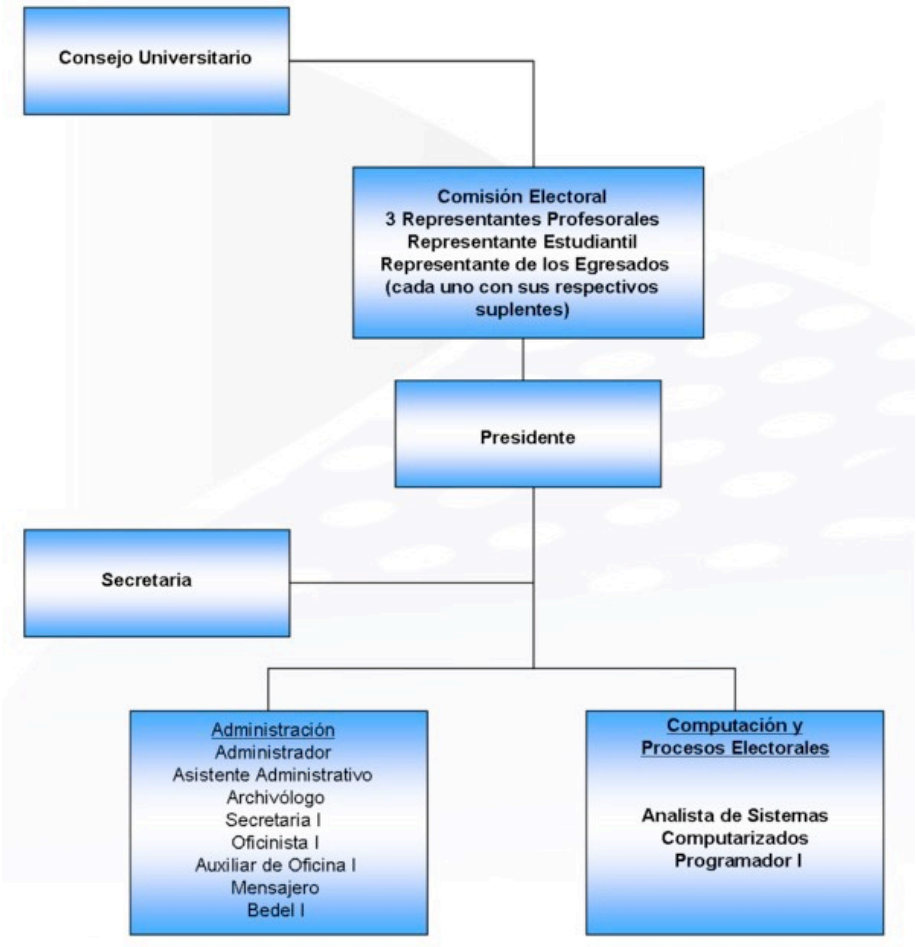


Ilustración 1 Estructura Organizativa de la Comisión Electoral UCV

Además, de la Comisión Electoral, existen dos organismos electorales que se involucran directamente para dar fiel cumplimiento al Reglamento de Elecciones Universitarias (REU), son las Subcomisiones Electorales y las Mesas Electorales.

1.1.3. Subcomisiones Electorales

Las Subcomisiones Electorales son los entes designados por la Comisión Electoral para realizar tareas inherentes a cada facultad. Cada una de estas "estará integrada por dos profesores miembros de la respectiva Asamblea de Facultad y por un estudiante regular de la misma Facultad. En la misma oportunidad la Comisión Electoral nombrará los respectivos suplentes, quienes deberán reunir las condiciones exigidas para los principales." (Art. 13 de REU) *[11]*.

Las Subcomisiones Electorales durarán un año en sus funciones y deberá elegir de su seno un Presidente y designará de fuera de su seno un Secretario, quien debe ser miembro del Personal Docente y de Investigación de la Universidad.

El Art. 16 del REU *[11]* contempla que las atribuciones de cada Subcomisión Electoral son:

1. Nombrar y remover a los integrantes de las mesas electorales e informar, tanto al Decano como a la Comisión Electoral sobre dichos nombramientos y remociones.

2. Distribuir entre las mesas electorales el material requerido para las elecciones, y, en caso de que éste no hubiera sido remitido oportunamente, reclamar a la Comisión Electoral.

3. Proponer a la Comisión Electoral los lugares de votación.

4. Levantar las actas de totalización de la respectiva Facultad, según lo dispuesto en el Capítulo V de este Reglamento.

5. Recabar de las mesas electorales las actas de votación y escrutinio y remitirlas a la Comisión Electoral.

6. Informar a la Comisión Electoral de la marcha del proceso electoral y, en especial, de las irregularidades que se hubieran observado.

7. Cumplir con las disposiciones que emanen de la Comisión Electoral.

1.1.4. Mesas Electorales

Las Mesas Electorales son entes designados por cada Subcomisión para velar por el correcto desarrollo del proceso de votación. Cada subcomisión tiene el deber de nombrar tantas mesas electorales sean necesarias para cada facultad y estarán integradas por tres profesores miembros del claustro y un secretario, quien debe ser miembro del personal docente y de investigación de la facultad. Solo en el caso de elecciones de representantes estudiantiles ante el cogobierno universitario, las mesas electorales se integrarán por dos miembros ordinarios del personal docente y de investigación, un estudiante regular y un secretario. Al momento de su instalación, cada mesa electoral elegirá de su seno a un presidente.

El Art. 22 del REU *[11]* contempla que las atribuciones de cada Mesa Electoral son:

1. Presenciar la votación correspondiente con estricta sujeción a lo establecido en el Capítulo III de este Reglamento.

2. Reclamar de la Subcomisión Electoral el material requerido para las elecciones, cuando éste no le hubiera sido entregado oportunamente.

3. Velar por el secreto del voto. A tal efecto, deberá comprobar que el sitio dispuesto para la votación garantiza suficientemente dicho secreto.

4. Colocar en sitio visible en el local de la votación y en el día fijado para ella los nombres de los candidatos y de las listas inscritas, con la especificación de los candidatos que las integran.

5. Cuidar del mantenimiento del orden en el lugar de las votaciones, en colaboración con las autoridades respectivas, quienes tomarán las medidas necesarias para garantizarlo, de conformidad con lo dispuesto en el Artículo 6o de la Ley de Universidades.

6. Realizar el escrutinio con estricta sujeción a lo dispuesto en los Capítulos IV y V de este Reglamento.

7. Levantar las actas de votación y de escrutinios, según lo dispuesto en los Capítulos III y V de este Reglamento.

8. Informar a la Subcomisión Electoral de la marcha del proceso electoral y en especial de las irregularidades que se hubieran observado.

9. Cumplir con las disposiciones que emanen de la Comisión Electoral y de la Subcomisión respectiva.

1.1.5. Objetivos

El propósito de esta Comisión está en generar confianza en el sistema democrático y que los electores crean en los procesos electorales organizados y ejecutados en esta dependencia, alcanzando los siguientes objetivos [10]:

- Tomar las medidas conducentes a la eficaz organización y desarrollo de los procesos electorales.
- Garantizar la alternabilidad en los diferentes cargos de las Autoridades Universitarias, de los Decanos, Representantes Profesorales, Estudiantiles y Egresados a los distintos organismos de Cogobierno, una vez vencidos sus respectivos períodos.
- Promover la participación democrática de la comunidad profesoral y estudiantil en los procesos electorales.

1.1.6. Funciones

- Mantener actualizados los siguientes registros electorales: Claustro Universitario, Asambleas de Facultades, Estudiantiles por Facultades y Escuelas, Egresados, Registro para elección de autoridades interinas y de los representantes estudiantiles ante los Consejos de Facultades.
- Convocar a cada uno de los procesos electorales antes de que estén vencidos los lapsos de duración de los cargos y representaciones.

- Planificar, coordinar y dirigir la ejecución de cada uno de los siguientes procesos electorales:
    - Autoridades Universitarias.
    - Decanos.
    - Representantes Profesorales ante el Consejo Universitario, Facultades y Escuelas.
    - Representantes Estudiantiles ante los organismos de Cogobierno Universitario.
    - Representantes de los Egresados ante los organismos de Cogobierno Universitario.
    - Representantes de Estudiantes y Egresados ante la Comisión Electoral.
- Elaborar informe de cada una de las elecciones realizadas y remitirlo al Consejo Universitario.
- Elaborar las credenciales para los distintos cargos de autoridades, de decanos, representantes profesorales, estudiantiles y egresados, a fin de remitirlo a las instancias correspondientes para su juramentación.

1.1.7. Atribuciones

Según el Artículo 9 del Reglamento de Elecciones Universitarias [11] "Son atribuciones de la Comisión Electoral:"

1. Tomar las medidas conducentes a la eficaz organización y desarrollo de los procesos electorales.
2. Elaborar y actualizar los registros electorales.
3. Oír y decidir en primera instancia administrativa, las impugnaciones que se formularen en relación con la composición de los registros electorales.
4. Nombrar y remover a los miembros de las Subcomisiones electorales e informar al Decano de la respectiva Facultad sobre dichos nombramientos o remociones.
5. Asegurarse del cumplimiento de sus respectivas funciones por parte de las Subcomisiones Electorales y, en casos de urgencia, asumir directamente el conocimiento de materias correspondientes a dichas Subcomisiones, cuando éstas, por cualquier causa, no hubieren resuelto sobre las mismas.

6. Remitir al Consejo Universitario, cada vez que sea necesario, las listas de los mandatos por finalizar y proponer las fechas de las respectivas elecciones, a fin de proceder a la correspondiente convocatoria.
7. Convocar a elecciones universitarias en los plazos legales.
8. Hacer las convocatorias, participaciones y avisos relativos al proceso electoral.
9. Admitir los candidatos propuestos para las elecciones, previa comprobación de que reúnen las condiciones requeridas para el cargo o representación correspondiente, y de que la postulación se ha hecho de conformidad con la Ley y los Reglamentos.
10. Fijar, previa consulta con la autoridad universitaria respectiva, los locales destinados para las votaciones. El consejo Universitario será informado de los locales escogidos.
11. Recibir las actas de votación, de escrutinios y de totalización por Facultades y, en el caso de que así fuere necesario, hacer los cómputos pertinentes y elaborar el acta de totalización final.
12. Proclamar a los candidatos electos siguiendo, si fuere el caso, el procedimiento establecido en el artículo 171 de la Ley de Universidades.
13. Denunciar ante el Consejo Universitario las irregularidades observadas y transmitir aquéllas que le hayan sido denunciadas.
14. Calificar la propaganda electoral y, si fuere necesario, ordenar el retiro de la propaganda inapropiada.
15. Preparar y distribuir con la debida anticipación el material necesario para los procesos electorales.
16. Extender las credenciales a los testigos electorales con expresión de la mesa y la elección correspondiente, así como la plancha o candidato que representa.
17. Elaborar para cada proceso electoral, las instrucciones correspondientes para las mesas electorales.
18. Organizar y conservar su archivo y los libros, actas y demás recaudos referentes a los procesos electorales.
19. Llevar los protocolos de sus reuniones y archivarlos debidamente.
20. Solicitar a las directivas de los Colegios o Asociaciones Profesionales la designación de los representantes de los egresados, una vez concluidos los lapsos de sus representantes y previa participación al Consejo Universitario.

21. Otorgar la credencial de reconocimiento a los representantes de los egresados, designados por los colegios y asociaciones, una vez verificado el cumplimiento de los requisitos exigidos.
22. Comunicar a las instancias universitarias pertinentes los representantes de los egresados reconocidos por la Comisión para los respectivos organismos de Cogobierno.
23. Convocar, organizar y efectuar la elección para elegir el representante de los egresados ante los Consejos de las Facultades que tienen más de un Colegio o Asociación de egresados.
24. Convocar, organizar y efectuar la elección para elegir el representante de los egresados ante el Consejo Universitario.
25. Convocar, organizar y efectuar la elección para elegir el representante de los estudiantes ante la Comisión Electoral entre los Representantes Estudiantiles a los Consejos de Facultad.

1.2. Procesos Involucrados en los Eventos Electorales

En la Comisión Electoral de la UCV intervienen varios procesos en la ejecución de todas las elecciones realizadas dentro de la Universidad. Dichas elecciones están referidas en la Ley de Universidades y en el Reglamento de Elecciones Universitarias de la UCV y se describen a continuación:

1. Elección de Autoridades Universitarias: se elige Rector, Vicerrector Académico, Vicerrector Administrativo y Secretario.
2. Elección de Autoridades de Facultad: se eligen los Decanos correspondientes a cada Facultad.
3. Elección de los Representantes de los Profesores ante el Cogobierno Universitario: se eligen los representantes profesorales al Consejo Universitario, Consejo de Facultad y Consejo de Escuela.
4. Elección de los Representantes de los Estudiantes ante el Cogobierno Universitario: se eligen los representantes estudiantiles al Consejo Universitario, Consejo de Facultad, Asamblea de Facultad y Consejo de Escuela.

5. Elección de los Representantes de los Egresados ante el Cogobierno Universitario: se eligen los representantes de los egresados al Claustro Universitario, Consejo Universitario, Consejo de Facultad, Asamblea de Facultad y Consejo de Escuela.
6. Elección de Candidatos al Consejo de Apelaciones.
7. Elección de los Representantes de los Estudiantes ante la Comisión Electoral.
8. Elección de los Representantes de los Egresados ante la Comisión Electoral.
9. Elección de los Representantes ante el Gobierno Estudiantil: para esta elección en particular, la Comisión Electoral presta su apoyo técnico para la realización de las mismas, las cuales se rigen con los mismos procesos establecidos para las elecciones del Cogobierno Universitario.

A nivel general, para cada una de las elecciones antes descritas, se sigue un mismo proceso el cual se puede dividir en varios subprocesos, los cuales se representan en la Ilustración 2.

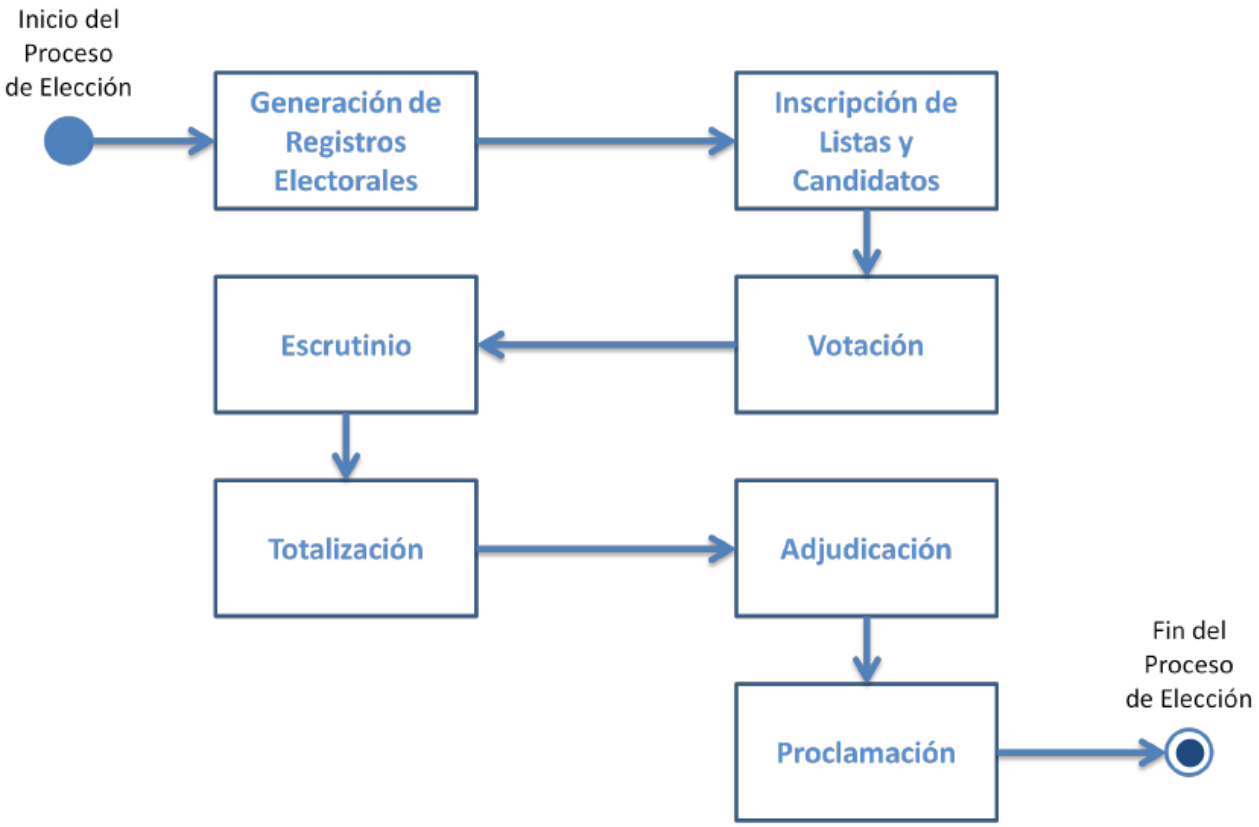


Ilustración 2 Flujo de Trabajo General

1.2.1. Generación de Registros Electorales

Este proceso se realiza básicamente para dar cumplimiento al Artículo 26 del REU [11] el cual dice que: "*Los Registros Electorales serán permanentes, de acuerdo a lo establecido en este Reglamento. Cualquier interesado podrá conocer de su composición en cualquier momento*". Estos registros electorales servirán de base para todos los procesos electorales que se realicen en la UCV y deben ser publicados, como mínimo, en treinta días continuos antes de la respectiva elección (Ilustración 3). A lo largo del Reglamento de Elecciones Universitarias se identifican varios puntos importantes con respecto a los registros electorales, entre los que podemos mencionar:

- No podrá votar quien no aparezca como elector en el Registro Electoral (Art. 25).
- Ningún elector puede aparecer dos veces en el Registro Electoral preparado para una misma elección. En caso de que, para una misma elección, un elector aparezca calificado para votar en dos Escuelas de una misma Facultad o en dos Facultades, ejercerá el derecho a voto solamente en la Escuela o Facultad en que hubiera ingresado primero. En igualdad de circunstancias, la Comisión Electoral dispondrá el lugar donde votará el elector. En caso de que, para una misma elección, un profesor estuviera además legitimado para votar en representación de otro de los sectores de la comunidad universitaria, ejercerá su derecho a voto sólo en su condición de profesor (Art. 32).

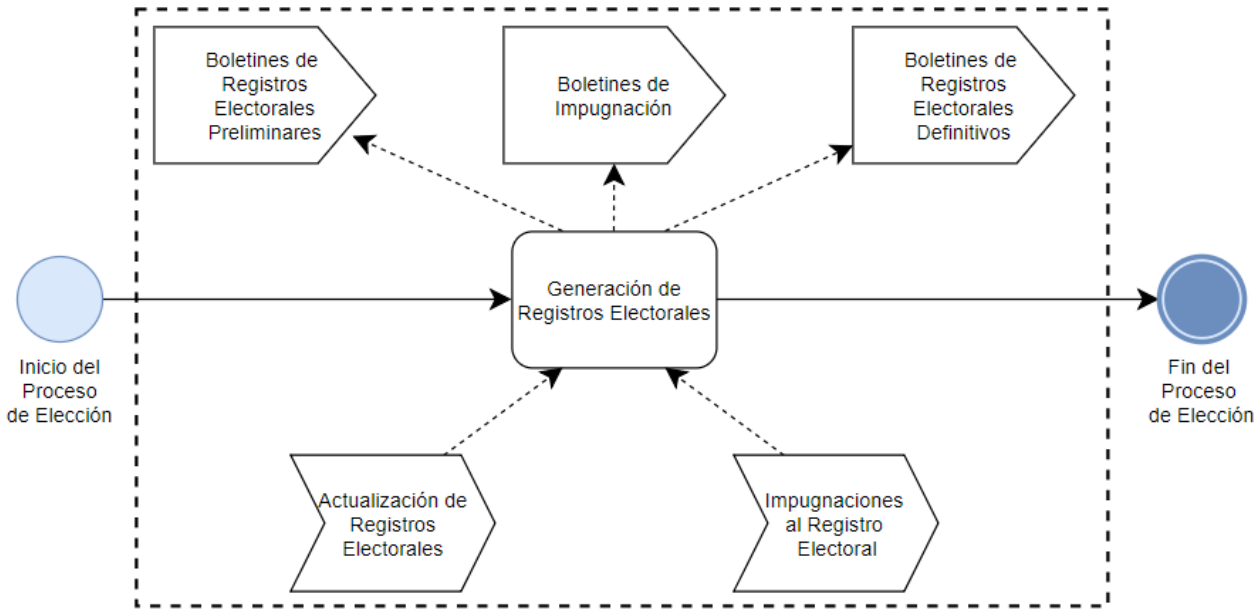


Ilustración 3 Detalle del Proceso de Generación de Registros Electorales

Este proceso comprende los subprocesos de Recopilación y Actualización de Registros Electorales, Generación de Registros Electorales Preliminares e Impugnaciones a los Registros Electorales.

- **Recopilación y Actualización de Registros Electorales**: en este subproceso se realiza, primordialmente, el conjunto de tareas necesarias para mantener permanentemente el registro electoral. Esto se hace solicitando a la Secretaría de la UCV, en el caso de los egresados, a los distintos Departamentos de Control de Estudios de todas las Facultades, en el caso de los estudiantes, y a los diferentes Decanatos de todas las Facultades, en el caso de los profesores, todos los datos necesarios para su actualización. Una vez recibidos dichos registros, la Comisión Electoral se encarga de depurarlos e ingresarlos en la base de datos correspondiente.
- **Generación de Registros Electorales Preliminares**: una vez actualizado el Registro Electoral, se genera el Registro Electoral Preliminar que consiste en filtrar la data recopilada en el apartado anterior, tomando como base las condiciones necesarias para ser elector dentro de la Universidad para la elección correspondiente. Las mencionadas condiciones se encuentran contempladas en la Ley de Universidades y en el Reglamento de Elecciones Universitarias de la UCV. Una vez aplicados dichos filtros, se genera un boletín informativo con el contenido del Registro Electoral Preliminar y se abre el lapso de Impugnaciones.
- **Impugnaciones a los Registros Electorales**: durante el lapso de Impugnaciones, cualquier miembro de la comunidad universitaria que posea la cualidad de elector puede objetar la composición del Registro Electoral Preliminar. Existen varios tipos de Impugnaciones:
    - **Por Inclusión**: esta se produce debido a la ausencia de algún elector en el Registro.
    - **Por Exclusión**: esta se genera dado que un elector considera que él u otro elector no debería estar incluido en el Registro.
    - **Por Cambio de Escuela**: se produce cuando un elector considera que, por cursar distintas carreras en paralelo, desea votar en otra Escuela diferente a la

que aparece en el Registro, o aparece inscrito en una escuela incorrecta. Este tipo de impugnación se realiza en dos pasos, el primero por exclusión de la escuela actual, y el segundo por inclusión en la escuela deseada.

- **Por Corrección de Datos**: se produce cuando existen errores en los datos personales del elector.

1.2.2. Inscripción de Listas y Candidatos

El proceso de inscripción de Listas y Candidatos, como se refleja en la Ilustración 4, se alimenta del registro electoral ya depurado correspondiente a la elección a la que sirve como base, cumpliendo con lo establecido en el Artículo 48 del Reglamento de Elecciones Universitarias [11]. Este proceso se conforma por dos tareas que se describen a continuación:

- **Inscripción de Listas y Candidatos**: es el evento por el cual pueden postularse aquellas personas que pertenezcan al Registro Electoral a los cargos a elegir en el evento electoral que organiza la Comisión Electoral. A ésta le corresponde la verificación de las condiciones necesarias para ser candidato.
- **Impugnaciones de Listas y Candidatos**: en este evento se le permite a cualquier persona que pertenezca al Registro Electoral objetar la inscripción de uno o más candidatos o de una plancha, en el evento que esté organizado por la Comisión Electoral.

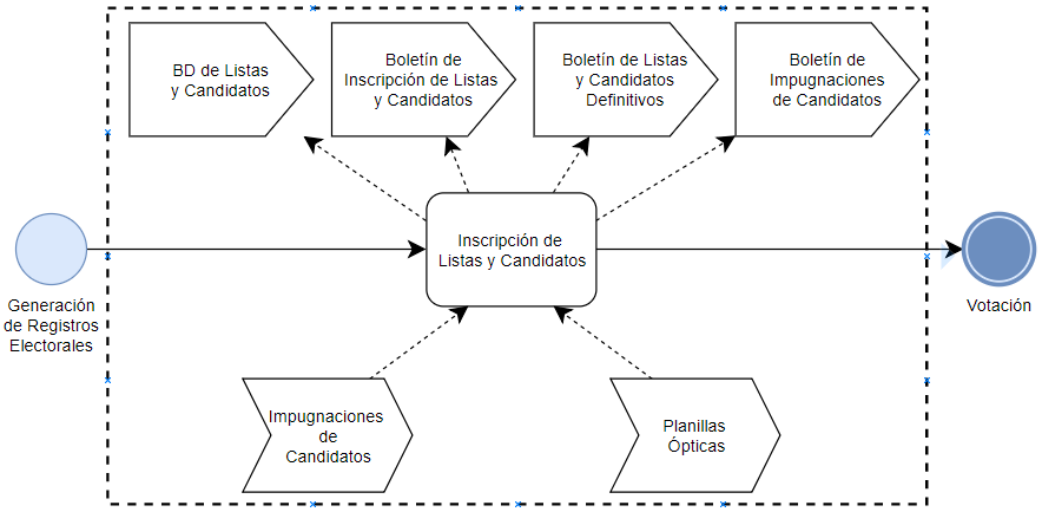


Ilustración 4 Detalle de Inscripción de Listas y Candidatos

1.2.3. Votación

El proceso de votación consiste en facilitar al elector las herramientas necesarias para seleccionar los candidatos de su preferencia durante un evento electoral. En este proceso están involucrados las Subcomisiones Electorales, en el rol de supervisora del proceso a nivel de Facultad, y los Miembros de Mesa, quienes juegan un papel crucial para la operatividad de dicho evento, ya que son los responsables de la ejecución del proceso de Votación en cada mesa que representan (Ilustración 5). El proceso de votación es manual, la comisión electoral imprime las planillas con los nombres de los candidatos y es entregada al elector al momento de ejercer el voto. Posteriormente estas planillas son leídas por máquinas de lectura óptica para el conteo de los votos.

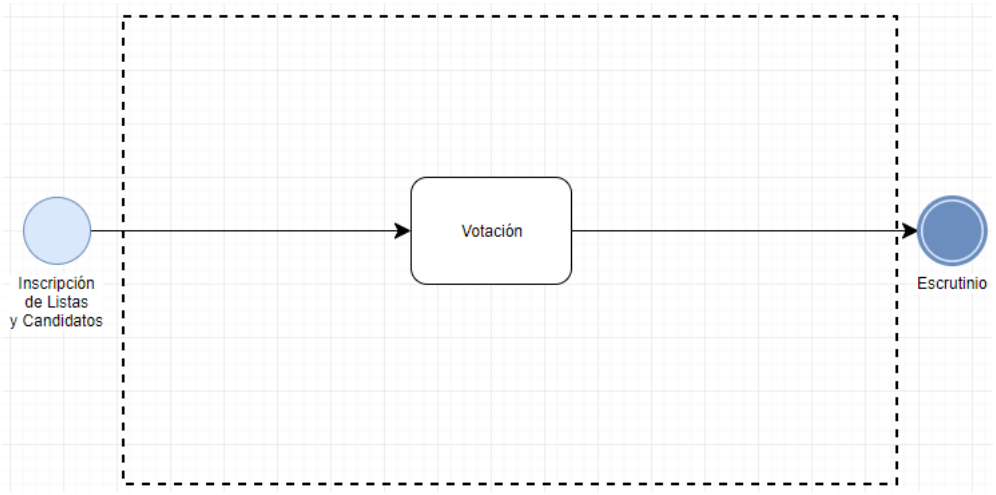


Ilustración 5 Detalle del Proceso de Votación

1.2.4. Escrutinio, Totalización y Adjudicación

Aunque los procesos de Escrutinio, Totalización y Adjudicación son procesos bien delimitados a la hora de su ejecución tienen mucha interoperabilidad entre ellos, por lo cual es necesario definirlos en paralelo.

El subproceso de escrutinio consiste en el conteo de los votos emitidos por los electores en el proceso de votación e inicia con el traslado del material electoral junto con las boletas a escrutar a las “Zona de Escrutinio” por parte de los miembros de mesa de cada escuela. Los miembros de mesa entregan todo el material utilizado durante el proceso de votación al

personal de la Comisión Electoral encargado de la Recepción del Material. La lectura de las boletas es realizada por "Estaciones de Máquinas de Lectura Óptica" (Estaciones MLO), los resultados serán enviados por lotes a un servidor local para realizar el subproceso de totalización. Posteriormente estos resultados de cada Zona de Escrutinio son transmitidos al servidor central a través de una Red Virtual Privada (VPN) para poder realizar el subproceso de adjudicación.

1.2.5. Proclamación

En este proceso, y una vez finalizado el lapso de impugnaciones de resultados, se ratifican los cargos adjudicados en el proceso anterior.

Un punto de extrema importancia en este proceso es la resolución de empates, ya que es aquí donde se toman las decisiones al respecto, basándose en el Art. 171 de la Ley de Universidades citado anteriormente.

2. Cadena de Bloques

La cadena de bloques es una base de datos que se halla distribuida, protegida criptográficamente y organizada en bloques de transacciones ordenados por marcas de tiempo y enlazados entre sí matemáticamente. Funciona como un libro para el registro de transacciones de cualquier índole, todos los participantes de la red guardan una copia del libro y estas no pueden ser alteradas o eliminadas sin el consentimiento de la comunidad, esta premisa se cumple mientras que participantes honestos controlen colectivamente más poder de procesamiento (CPU) que cualquier grupo de participantes atacantes en cooperación la red. La clave de esta tecnología es el consenso: si todos tenemos la misma información, esa información es verdad. Esta herramienta se vale de la combinación de diferentes tecnologías tales como la criptografía, los Arboles Merkle, llaves asimétricas y firmas digitales para lograr la continuidad en el tiempo y la inmutabilidad.

Esta tecnología se conceptualiza por primera vez en el paper Bitcoin: A Peer-to-Peer Electronic Cash System" publicado en el 2008. En este se especifica el sistema de criptomoneda Bitcoin. Un sistema financiero descentralizado basado en una red peer-to-peer. *Blockchain* es el componente

central de la criptomoneda, donde sirve como el libro público para todas las transacciones en la red. Mediante el uso de cadena de bloques, el bitcoin se convirtió en la primera moneda digital para resolver el problema de los gastos dobles sin requerir una autoridad de confianza y ha sido la inspiración para muchas aplicaciones adicionales.

Las cadenas de bloques pueden ser clasificadas de acuerdo al acceso al procesamiento de transacciones (çvs. permisivo) y acceso a datos (público vs. privado).

Según acceso al procesamiento de transacciones se tiene:

- Permisivo. Una cadena de bloques sin permisos es una cadena de bloques, en la que no hay restricciones sobre las identidades de los procesadores de transacciones (es decir, los usuarios que son elegibles para crear bloques de transacciones).
- Permisivo. Una cadena de bloques autorizada es una cadena de bloques, en la que el procesamiento de transacciones se realiza mediante una lista predefinida de sujetos con identidades conocidas.

También podemos encontrar que según el acceso a la data:

- Publica. Una cadena de bloques pública es una cadena de bloques, en la que no hay restricciones para leer los datos de la cadena de bloques (que aún pueden estar encriptados) y enviar transacciones para su inclusión en la cadena de bloques.
- Priva. Una cadena de bloques privada es una cadena de bloques, en la que el acceso directo a los datos de la cadena de bloques y la presentación de transacciones se limita a una lista predefinida de entidades.

2.1. Características

La tecnología *blockchain* posee una serie de características que la hacen ser aplicables en cualquier ámbito que se requiera tener un registro de datos distribuido de forma segura.

2.1.1. Descentralizado

Se trata de una red de pares, o red entre iguales, donde todos los nodos que forman la red tienen una copia de la cadena de bloques, por lo que no existe una copia oficial centralizada, se otorga calidad a los datos por la replicación masiva de la base de datos. Lo nodos se comportan como iguales entre sí, actuando a la vez como servidores y clientes del resto de nodos de la red, por lo que ningún usuario es de más confianza que cualquier otro.

2.1.2. Sistema abierto

Es abierto porque cualquier persona puede formar parte de la red solo tienen que descargar el programa. Luego podrá realizar movimientos y transacciones con monedas virtuales y acceder a los datos registrados en la cadena de bloques.

Los *peers* de la red pueden llegar a tener versiones diferentes de la base de datos, solo guardan la versión con la puntuación más alta que conocen. La cadena de bloques tiene un algoritmo especificado para marcar diferentes versiones de la cadena para que una con un valor más alto pueda ser seleccionada sobre otras, cuando se recibe una versión de puntuación más alta (usualmente la versión antigua más un solo bloque añadido) extienden o sobrescriben su propia base de datos y retransmiten la mejora a sus pares.

2.1.3. Seguridad

Los bloques que forman parte del *blockchain* son ordenados en la cadena por orden cronológico y tienen un código alfanumérico conocido como hash, que corresponde al bloque que los precede, gracias a ese hash todos están referenciados por el bloque que los creo, por lo que solo los bloques que contienen un código valido son introducidos en la cadena y replicados a todos los nodos. Es gracias a este método lo que hace virtualmente imposible modificar un bloque que ha sido introducido.

Por lo tanto, el *blockchain* nos permite llevar a cabo, una contabilidad pública de los movimientos realizados en la red de manera transparente, minimizando la posibilidad de fraude, no permitiendo la perdida de datos y con un sistema totalmente trazable. *Blockchain*

se basa en algoritmos criptográficas como firma digital, hashes y Árbol de Merkel para lograr sus mecanismos de seguridad.

2.1.4. Anonimato

El protocolo se utiliza para realizar transacciones entre direcciones (equivalentes a lo que sería una cuenta bancaria). Pero estas direcciones las generan los propios usuarios, por lo que no hay un registro centralizado que permita asignar una dirección a una persona concreta.

2.1.5. Consenso distribuido

Es la lógica que hace funcionar el sistema. En una red distribuida en la que diferentes nodos pueden estar haciendo operaciones simultáneas que deben propagarse por la red y en ocasiones colisionan entre sí, se establece un sistema claro para determinar qué transacciones son válidas y se incorporan a la cadena y cómo se resuelven los conflictos y colisiones.

2.2. Componentes

La tecnología está basada en cuatro fundamentos: el registro compartido de las transacciones (ledger), el consenso para verificar las transacciones, un contrato que determina las reglas de funcionamiento de las transacciones y finalmente la criptografía, que es el fundamento de todo. En esta sección veremos los componentes que hacen esto posible.

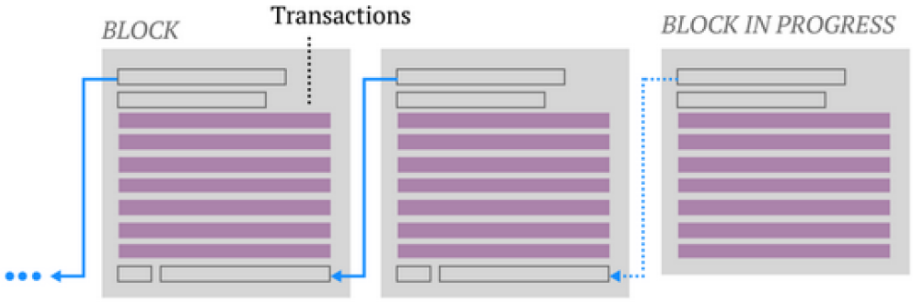
2.2.1. La cadena de bloques (*blockchain*)

Es el fundamento principal del sistema. Representa el registro donde se anotan todas las transacciones que se van realizando. Tiene toda la información de todos los elementos y todas las transacciones que se han llevado a cabo desde el inicio del sistema.

2.2.2. Bloques

Los bloques es la estructura en donde se empaquetan las transacciones realizadas que posteriormente son validadas por los mineros para incluirlas en la *blockchain* y distribuidas a todos los nodos que forman la red. Cada bloque que forma parte de la cadena (menos el primer bloque que inicia la cadena) está formado por:

1. Un código alfanumérico (hash) que enlaza con el bloque anterior
2. El "paquete" de transacciones que incluye
3. Otro código alfanumérico (hash) que enlazará con el siguiente bloque.

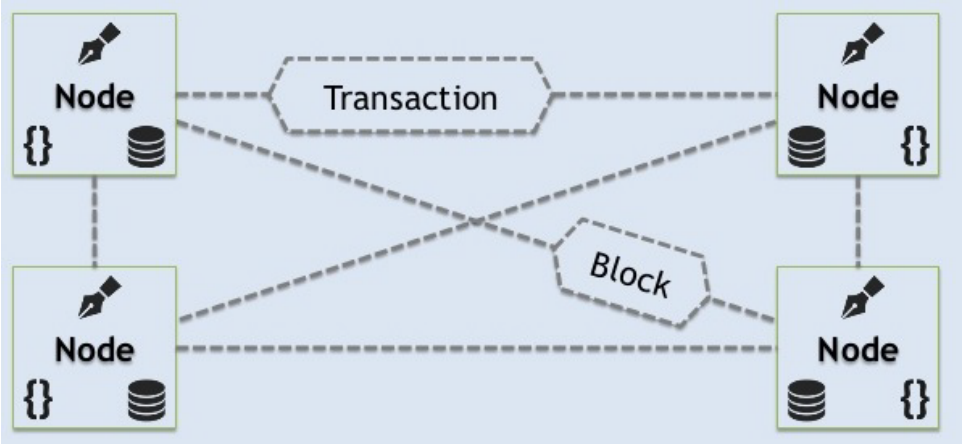


2.2.3. Nodos

Son computadoras personales o megacomputadoras, según la complejidad de la red, conectadas a la red utilizando un software que almacena y distribuye una copia actualizada en tiempo real del *blockchain*. Con independencia de la capacidad de cómputo, todos los nodos han de poseer el mismo software/protocolo para comunicarse entre sí. De otro modo no podrán conectarse ni formar parte de la red de una *blockchain*, sea ésta pública, privada o híbrida. Si en una *blockchain* pública estos nodos no tienen por qué identificarse, en una *blockchain* privada los nodos se conocen entre sí, pudiendo también ser iguales entre ellos.

Cada vez que un bloque se valida y se añade a la cadena, el cambio es comunicado a todos los nodos y este se añade a la copia que cada uno almacena. Algunos, conocidos como mining pools o grupos de minera, se encargan además de escuchar nuevas transacciones y

agruparlas en bloques para proponerlos como trabajo a los mineros, que luego de ser conformados son propagados a la red y añadidos a la cadena.



2.2.4. Mineros

Los mineros son ordenadores dedicados que aportan su poder computacional a la red para verificar las transacciones que se llevan a cabo. Son computadoras que se encargan de autorizar la adición de los bloques de transacción. Estos siguen los siguientes pasos:

1. Las nuevas transacciones se transmiten a todos los nodos
2. Cada nodo de la minera recoge nuevas transacciones en un bloque.
3. Cada nodo minero trabaja en la búsqueda de una prueba de trabajo para su
4. bloque.
5. Cuando un nodo de la minera encuentra una prueba de trabajo, este transmite
6. el bloque a todos los nodos.
7. Los demás nodos aceptan el bloque solo si todas las transacciones son validas
8. y no se hayan gastado.
9. Los nodos expresan su aceptación del bloque trabajando en la creación del próximo bloque en la cadena, utilizando el hash del bloque aceptado como el hash anterior.

Cada vez que alguien completa un bloque recibe una recompensa en forma de criptomoneda según sea la red y/o por cada transacción que se realiza.

2.2.5. Protocolo Estándar

Forma de software informático para que una red de ordenadores (nodos) pueda comunicarse entre sí. Existen protocolos muy conocidos, como el TCP/IP para internet o el SMPT para el intercambio de correos electrónicos. El protocolo de una *blockchain* funciona de la misma forma: otorga un estándar común para definir la comunicación entre los ordenadores participantes en la red.

2.2.6. Red entre Pares o P2P (Peer-to-Peer en inglés)

Se trata de una red de pares, o red entre iguales, donde todos los nodos que forman la red se comportan como iguales entre sí, actuando a la vez como servidores y clientes del resto de nodos de la red.

2.3. Funcionamiento

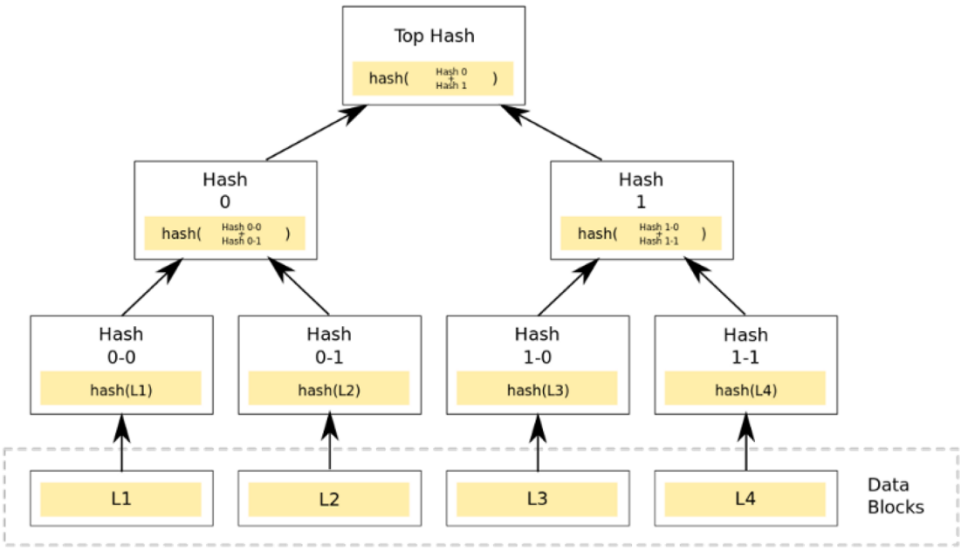
La *blockchain* consiste de una red P2P, compuesta por nodos. Es necesario que los nodos que componen la red estén sincronizados, manteniendo la consistencia de los datos con un algoritmo de consenso. El caso básico para explicar el funcionamiento de *blockchain* es el Bitcoin (primera *blockchain* conceptualizada).

En la red se siguen una serie de pasos para lograr gestionarla:

1. Transacciones nuevas son emitidas a todos los nodos.
2. Cada nodo recolecta nuevas transacciones en un bloque.
3. Cada nodo trabaja en encontrar una prueba-de-trabajo difícil para su bloque.
4. Cuando un nodo encuentra una prueba-de-trabajo, emite el bloque a todos los nodos.
5. Los nodos aceptan el bloque si todas las transacciones en el bloque son válidas y no se han gastado ya.
6. Los nodos expresan su aceptación del bloque al trabajar en crear el próximo bloque en la cadena, utilizando el hash del bloque aceptado como el hash previo.

Como se explicó anteriormente, la *blockchain* está compuesta por bloques, cada uno de estos contiene información sobre las transacciones de un periodo concreto, estas son almacenadas en una estructura denominada Merkle Tree (en honor a su creador: Ralph Merkle), también la información criptográfica del bloque precedente es decir, el código hash, y un número único llamado nonce , el cual es un valor arbitrario que puede utilizarse una sola vez, es generado por los mineros mediante la prueba de trabajo (Proof of Work o PoW) y sirve como método sencillo para autenticar un bloque en caso de una posible modificación o reutilización de su contenido, sin tener que volver a procesar toda la cadena, ahorrando así mucho trabajo computacional. De esta manera todos los bloques están enlazados cronológicamente.

El "Árbol de Merkel" es una estructura de árbol binario que almacena información en sus nodos hoja, por cada nodo hoja es generado un hash y es concatenado con su nodo hermano los cuales son agrupados para generar otro hash, este procedimiento se repite en todos los niveles del árbol hasta alcanzar un único bloque raíz que se conoce como "root hash". Esta estructura es utilizada debido a que el orden de acceso de un árbol es menor que usar una estructura secuencial, además se reduce el espacio utilizado en la cadena.



Esta estructura permite recorrer cualquier nodo del árbol para la verificación de que ningún dato ha sido falsificado o adulterado, debido a que el hash del último bloque es un resumen. Algún cambio a cualquiera de las transacciones (nodos hoja) requerir a propagar los cambios hasta la raíz del bloque, y las raíces de todos los bloques. Por lo tanto, si se conoce el último

hash, es posible obtener el resto del libro mayor desde una fuente no confiable y verificar que no ha cambiado.

Un argumento similar establece otra propiedad importante de la estructura de datos, se puede probar eficientemente que una transacción particular está incluida en el libro mayor. El usuario tendría que enviar un pequeño número de nodos en ese bloque de la transacción (este es el objetivo del arbol Merkle), así como una pequeña cantidad de información para cada siguiente bloque. La habilidad para probar inclusión de transacciones eficientemente es altamente deseable por rendimiento y escalabilidad.

Las cadenas de bloque también pueden utilizan otros esquemas de consenso, para serializar los cambios. Los métodos de consenso alternativos incluyen Proof of Stake y Proof of Burn.

2.4. Algoritmo de Consenso

Un algoritmo de consenso es un proceso informático utilizado para lograr un acuerdo sobre un único valor de datos entre procesos o sistemas distribuidos. Los algoritmos de consenso están diseñados para alcanzar la fiabilidad en una red que incluye múltiples nodos poco fiables. Resolver este problema, conocido como el problema del consenso, es importante en los sistemas de computación distribuida y multiagente. Para acomodar esta realidad, los algoritmos de consenso necesariamente asumen que algunos procesos y sistemas no estarán disponibles y que algunas comunicaciones se perderán. Como resultado, los algoritmos de consenso deben ser tolerantes a los fallos. Estos suelen asumir, por ejemplo, que tan solo una porción de los nodos responderá, pero requieren una respuesta de esa porción, un 51% como mínimo".

En *blockchain*, los algoritmos de consenso están diseñados para asegurar que las transacciones sean válidas y distribuidas entre una gran cantidad de participantes para así verificar la exactitud y la resistencia a través de la redundancia. Existen múltiples algoritmos de consenso, entre las que podemos encontrar:

- Prueba de Trabajo (Proof of Work – PoW)
- Prueba de Propiedad (Proof of Stake – PoS)
- Prueba de Importancia (Proof of Importance – PoI)

- Tolerancia Delegada ante Fallo Bizantino (Delegated Byzantine Fault Tolerance – dBFT)
- Prueba de Propiedad Delegada (Delegated Proof of Stake – dPoS)

2.4.1. Prueba de Trabajo (Proof of Work – PoW)

La prueba de trabajo es un algoritmo informático, utilizado para llegar a un acuerdo descentralizado, que determine cuál de los bloques se agregara a la cadena después de minado. La prueba de trabajo se lleva a cabo mediante la resolución de problemas matemáticos complejos y variables, donde los mineros intentan darle la solución correspondiente para lograr obtener una recompensa en la cadena de bloques.

Para minar un bloque de manera exitosa, es necesario ajustar el encabezado del bloque de tal manera que sea menor o igual que el objetivo (Hash). Los mineros llegan a este hash en particular, variando una pequeña porción del encabezado del bloque, llamado "nonce". Un nonce siempre comienza con "0" y se incrementa cada vez para obtener el hash requerido.

Este protocolo tiene como objetivo, evitar los ciberataques como los de denegación de servicio (DDoS) en los cuales se pretende agotar los recursos de un sistema informático, mediante el envío de múltiples solicitudes falsas.

2.4.2. Prueba de Propiedad (Proof of Stake – PoS)

La prueba de participación es un algoritmo para lograr un consenso descentralizado en donde importan son las cantidades de monedas almacenadas en sistema. Su creación fue el resultado de considerar el mecanismo de prueba de trabajo como un desperdicio de recursos, ya que los costos por el alto consumo eléctrico y el de los equipos necesitados para realizar la minería resultaban elevados.

Para la prueba de participación es más relevante la cantidad de monedas almacenadas en el sistema, lo que supone un interés por parte de la comunidad en que su rendimiento sea óptimo. Los cálculos no llevan tanta dificultad como en el protocolo anterior, basta con demostrar la posesión de un determinado porcentaje en las criptomonedas establecidas.

En este mecanismo los bloques no son minados por los usuarios, sino que se van acuñando según la participación predominante que tenga el usuario de la red y posteriormente se agregan a la cadena de bloques. Aunque el enfoque del mecanismo sea la participación, también se toman en cuenta otros factores como la selección aleatoria de bloques, la selección basada en la edad de monedas, los nodos principales, etc.

Un aspecto característico de la prueba de participación es que todos los bloques ya fueron previamente minados, es decir, se fija una cantidad en suministro de criptomonedas desde el principio, por lo que no se pueden extraer nuevos bloques.

2.4.3. Prueba de Importancia (Proof of Importance – PoI)

Es un algoritmo de consenso en el núcleo del software NEM. Cuanto mayor sea su importancia, mayor será su oportunidad de poder calcular un bloque (y recoger los honorarios dentro de ese bloque). POI ajusta su importancia dependiendo de cuántas transacciones hace un usuario, con quién las hace y otra serie de factores. Si no realiza ninguna transacción, el POI establecerá su importancia basándose únicamente en su saldo, es decir, un algoritmo POS (Proof Of Stake) tradicional.

El resultado de este mecanismo es que, aunque el titular no sea poseedor de la mayor cantidad de criptomonedas XEM, de igual forma obtiene importancia del mecanismo, si sus operaciones son rápidas y constantes.

2.4.4. Tolerancia Delegada ante Fallo Bizantino (Delegated Byzantine Fault Tolerance – dBFT)

En este método de consenso los nodos son denominados accionistas que votan por un nodo delegado, los accionistas solo pueden hacer transferencias o cambiar sus monedas, pero los delegados hacen la contabilidad, es decir, validan los bloques. Cuando se necesita una validación se elige a un delegado al azar y el 66% de los delegados restantes debe aprobar su trabajo, si no es aceptada se elige otro al azar y se reinicia el proceso. Los accionistas pueden cambiar a su delegado dependiendo de la comisión que cobre, promoviendo el bajo costo y el uso de la red.

Para que un nodo pueda ser delegado, debe tener una cantidad de monedas en su poder, en este sentido es similar a PoS, la diferencia radica en que los nodos tienen el mismo peso independientemente del capital que posean.

2.4.5. Prueba de Propiedad Delegada (Delegated Proof of Stake – dPoS)

En DPoS, aquellos que tienen el token de red pueden emitir votos para elegir a los productores de bloque; los votos se ponderan según la participación del votante, y los candidatos productores de bloque que reciben la mayor cantidad de votos son aquellos que producen bloques. Los usuarios también pueden delegar ("proxy") su poder de voto a otro usuario que puede votar en su nombre. DPoS es una democracia líquida y representativa con sufragio token holder. También se puede pensar en DPoS como una versión formalizada y digital de una jerarquía organizacional tradicional que opera de una manera completamente transparente.

2.5. Sistema de Votación Electrónica

Se considera un sistema de votación electrónica a la incorporación de recursos tecnológicos en cualquier subproceso del proceso de votación, bien sea en el registro de ciudadanos, la logística electoral, el ejercicio del voto, el escrutinio y/o la transmisión de resultados.

No existe una única forma de implementar voto electrónico, más bien podríamos decir que existen tres grandes tipos de sistemas a utilizar, que difieren no sólo en su implementación, sino y fundamentalmente en sus riesgos y beneficios. Los mecanismos más frecuentemente identificados como de voto electrónico se pueden agrupar en tres grandes conjuntos: a) los sistemas de recuento automático de votos mediante reconocimiento óptico de las marcas hechas en la boleta por parte de los ciudadanos, que son sistemas que hacen hincapié en el escrutinio electrónico; b) los sistemas de registro electrónico directo (RED, o DRE por su sigla en inglés) ejemplificados comúnmente con los denominados kioscos de votación o urnas electrónicas; c) los sistemas de votación a distancia a través de internet.

Las tecnologías del voto electrónico pueden acelerar el conteo de los votos y proveer una mejor accesibilidad para los votantes con algún tipo de discapacidad. Sin embargo, ha sido calificado como anticonstitucional en algunos países con el argumento de "no permitir la fiscalización del proceso" por personas sin conocimientos altamente especializados.

2.5.1. Sistemas Usados

Existen tres grandes grupos con lo que se puede implementar un sistema de votación electrónico.

2.5.1.1. Sistemas de Recuento Automáticos

Los primeros sistemas de esta clase datan del siglo XIX, cuando se comenzaron a implementar en la ciudad de Nueva York mediante tarjetas perforadas. Actualmente, la mayoría de los sistemas de este tipo se basan en el reconocimiento óptico de marcas hechas por el votante sobre la boleta, ya sea de forma directa o a través de una máquina de marcar boletas.

En principio, estos sistemas resuelven el problema más álgido de la incorporación de tecnología al sufragio: al mantener el principio de que la voluntad del elector se expresa en un trozo de papel anónimo, desacopla el acto de emisión de voto (que debe ser inauditable) del acto de escrutinio (que debe ser auditable en todos sus detalles). De esta manera es posible construir un sistema en el cual todos los resultados en los que la

informática está involucrada pueden ser auditados independientemente de los dispositivos usados y el software en sí, mediante el simple recurso de realizar un recuento manual.

Un elemento que no puede faltar en la aplicación de sistemas de recuento automático es la auditoría manual de los resultados arrojados por una porción estadísticamente significativa de las máquinas usadas, seleccionadas al azar luego del acto eleccionario. De lo contrario, una programación maliciosa del software de tabulación de votos podría alterar los resultados sin ser detectada.

2.5.1.2. Sistemas de Registro Electrónico Directo

Los Sistemas RED o DRE son aquellos que más se corresponden con el imaginario popular de las "urnas electrónicas". Los sistemas RED se caracterizan por realizar simultáneamente el registro y la tabulación del voto mediante un dispositivo informático que es operado directamente por el votante mediante un teclado, una botonera especial, o una pantalla táctil. A diferencia de los sistemas de recuento automático, en los que el soporte fundamental del voto es la boleta marcada por el ciudadano, en las máquinas RED el registro se realiza directamente en la memoria del dispositivo. Luego de la elección producen una tabulación de los datos de la votación almacenados en la memoria y una copia impresa. El sistema también puede proveer un medio para transmitir los votos o papeleta individuales o los totales de votos a una locación central para consolidar e informar los resultados desde las oficinas de la locación central. Estos sistemas usan un método de cómputo que cuenta las papeletas en el lugar de la votación. Típicamente, las papeletas se cuentan a medida que se van emitiendo y los resultados se imprimen luego del cierre de la votación.

Algunos sistemas de RED ofrecen además ayuda para personas con algún tipo de discapacidad, por ejemplo, mediante una interfaz de audio para superar las dificultades visuales. A diferencia de los sistemas de recuento automático, en los que el soporte fundamental del voto es la boleta marcada por el ciudadano, en las máquinas RED el registro se realiza directamente en la memoria del dispositivo.

2.5.1.3. Sistemas de Votación a Través de Internet

También conocidos como sistemas de votación a distancia, son mecanismos para emitir el sufragio desde una computadora común conectada a la red de redes, permitiendo que los sufragantes emitan su voluntad desde sus propios domicilios, desde puntos públicos de acceso, e incluso desde el extranjero. Existen variantes de estos sistemas que permiten emitir el voto no sólo desde una computadora personal, sino eventualmente también desde un teléfono celular o un sistema de televisión digital.

Uno de los desafíos más graves que enfrenta este tipo de sistemas es la identificación del votante, que es imprescindible para asegurar varias propiedades importantes del mecanismo, tales como evitar que alguien vote más de una vez o en nombre de otra persona, o que voten personas que no están habilitadas para hacerlo. Este problema suele resolverse mediante una clave unívoca y personal, que puede incluir elementos físicos de autenticación tales como la posesión de una tarjeta de identificación criptográfica o un generador de claves pseudoaleatorias.

Un problema adicional asociado a la identificación es que obligan a que la máquina que recibe el voto tenga conocimiento de quién lo está emitiendo. Esto ofrece un punto único de ataque para quien quiera violar el secreto del voto: basta con obtener la información almacenada en el servidor del sistema de votos para averiguar cómo votó cada persona que lo usó.

2.5.1.4. Votación Electrónica Empleando Blockchain

Los procesos electorales empleando algún componente tecnológico no son un tema nuevo y mucho menos popular entre los organismos electorales de los distintos países por sus "fallas" al garantizar los principios básicos de una elección. Sin embargo, ha sido considerado como un desarrollo prometedor que podría acelerar, simplificar y reducir el costo de las elecciones, e incluso podría conducir a un mayor número de votantes y al desarrollo de la democracia. Es un sistema que carece de un componente extra que transmita confianza y seguridad a los distintos usuarios. La tecnología *blockchain* puede ser este componente faltante en el sistema de votación electrónica para que por fin sea implementado en su totalidad.

Ya explicado en secciones anteriores, *blockchain* permite verificar, actualizar y mantener todos los datos de una red de forma descentralizada e independiente eliminando intermediarios, funcionando como un libro de registros digitales cifrados y compartido entre computadoras distribuidas. Todas las transacciones son transparentes y registradas permanentemente. *Blockchain* puede entenderse como un sistema en el cual los datos quedan cifrados en bloques de información y en donde perfectamente se podrían ir almacenando los votos de una elección.

Normalmente, los votos son registrados, administrados, contados y verificados por una autoridad central. El registro histórico no podría entonces cambiarse por alguna persona o porque otros votantes consideren que la decisión difiere de la suya. La votación electrónica basada en *blockchain* permitiría a los votantes tener más confianza en este tipo de sistema de votación, porque garantiza la anonimidad, seguridad, transparencia y descentralización en el proceso de votación, permitiéndoles tener un mayor control del registro del proceso.

Capitulo III – Marco Tecnológico

1. NEM (New Economy Movement)

NEM es un proyecto fundado en Singapur y lanzado en el año 2015. Consiste en una red peer to peer, en la que mediante una tecnología *blockchain* descentralizada, en la cual se pueden desarrollar nuevas aplicaciones, incluyendo tokens y criptomonedas. Este proyecto no es considerado como otro Altcoin, ya que su código fue creado desde cero con el objetivo de mejorar algunas deficiencias del códio de Bitcoin e incrementar sus capacidades en el sector empresarial.

La principal característica distintiva de NEM, es su sistema de activos inteligentes, el cual utiliza un método de contabilidad *blockchain*, compatible para gestionar una gran variedad de activos, como monedas, instrumentos financieros, cadenas de suministro, registros de propiedad y más.

NEM también aboga por un acceso más fácil gracias a la Prueba de Importancia (POI) en lugar de los mineros tradicionales que requieren una gran cantidad de poder de computación para extraer la moneda.

1.1. Activos Inteligentes

NEM está desarrollado en base a un potente sistema que te permite personalizar el uso de la *blockchain*. El "Sistema de Activos Inteligentes" otorga la capacidad de utilizar NEM como si fuese una *blockchain* personalizada para activos y aplicaciones.

En lugar de forzar a escribir desde cero tu propio código utilizando lenguajes de "*smart contract*" o utilizar métodos *off-blockchain* para definir activos personalizados para tu negocio NEM expone su funcionalidad a través de una potente interfaz API que se puede utilizar con cualquier lenguaje de programación. El código de lógica comercial existente se puede combinar fácilmente y utilizar la *blockchain* donde es más útil: en transacciones seguras y contabilidad.

Los Activos Inteligentes de NEM se crean utilizando cuatro partes estrechamente conectadas

1.1.1. Dirección

Las Direcciones NEM son contenedores de activos en la *blockchain* que puede representar un objeto único y actualizable. Las direcciones contienen Mosaics. Una dirección puede ser la cuenta de un usuario llena de monedas, un paquete que ha de ser enviado, la escritura de una casa o un documento que ha de ser notariado.

Los activos de direcciones se convierten realmente en inteligentes cuando se configuran con unas reglas especiales que definen como se van a controlar y relacionar entre sí, también en, cómo se pueden actualizar y transferir sus contenidos. Un tipo de regla, crucial, es el control multi firma (normalmente llamado "Multisig") que permite que la propiedad de los activos de Direcciones se pueda compartir de diferentes formas entre múltiples partes, todo en la *blockchain*.

1.1.2. Mosaic

Los Mosaics son activos fijos en la *blockchain* que pueden representar un conjunto de elementos múltiples *idénticos* que no cambian. Un Mosaic puede ser tan simple como un token, pero también puede representar un conjunto de activos más especializados como: puntos de recompensa, acciones, firmas, indicadores de estado, votos o incluso otras divisas. Cada Mosaic se define por una variedad de atributos como: nombre, descripción, cantidad, divisibilidad, transferibilidad y más.

1.1.3. Namespace

Los Namespaces permiten crear un sitio único, para los activos y negocios, en la *blockchain* de NEM. Un Namespace empieza con un nombre único que se escoge, similar al nombre de un dominio en internet. Una vez escogido el nombre, se tiene la capacidad de definir subdominios propios, así como activos. Esto hace que los activos sean confiables, fáciles de usar y únicos.

1.1.4. Transacciones

Las Transacciones son la forma en la que los Activos Inteligentes se ponen en acción. Las Transacciones permiten transferir Mosaics entre Direcciones, transferir o configurar la propiedad de las Direcciones (incluyendo el uso de reglas Multi firma), enviar mensajes y más. La *blockchain* de NEM incluye una función integrada de mantenimiento del tiempo impulsada por consenso, de modo que las transacciones se imprimen con una marca de tiempo de forma automática y precisa.

1.2. Arquitectura de NEM

La plataforma *blockchain* de NEM está construida a partir de una red de nodos que ejecutan el software de servidor del nodo central de NEM. En resumen, estos nodos proporcionan una plataforma potente, fácil de usar, estable y segura, donde se llevan a cabo, se buscan y se registran las transacciones de Activos inteligentes de manera inmutable, en el libro contable de la *blockchain*. Para hacer esto, los nodos, en esta red, cumplen dos funciones esenciales

1.2.1. Portal de Acceso al Servidor API

NEM proporciona a través de sus nodos un API para que las aplicaciones accedan a las funcionalidades de la *blockchain* y sus características, por lo que tu aplicación no necesita ejecutar ningún software complejo del nodo. Esto significa que la *blockchain* se puede usar para crear una variedad de soluciones de arquitectura con un código liviano en cualquier lenguaje de programación.

Algunos ejemplos de soluciones de arquitectura:

- **Acceso directo a aplicaciones móvil**: Se conecta una aplicación liviana directamente a las características de la *blockchain*

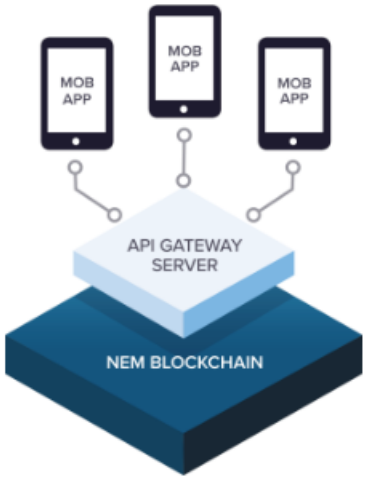


Ilustración 6 Arquitectura NEM - Acceso directo a aplicaciones Móvil

- **Modelo cliente/servidor**: Un portal de enlace administra el uso de la *blockchain* para una aplicación cliente o un servicio web.

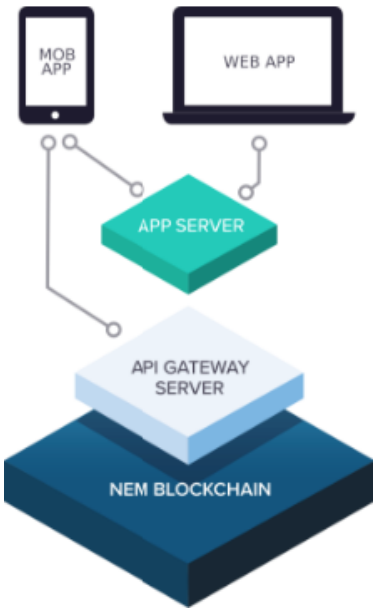


Ilustración 7 Arquitectura NEM - Modelo Cliente/Servidor

- **Integración de sistemas heredados**: Un portal del enlace al servidor vincula la lógica de contratos comerciales existentes, sistemas o bases de datos, con el libro contable de la *blockchain*

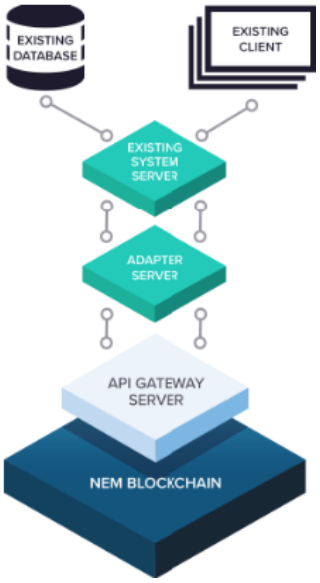


Ilustración 8 Arquitectura NEM - Integración de sistemas heredados

1.2.2. Red de Nodos

La red *blockchain* P2P de NEM se basa en nodos que colaboran entre sí, cada uno de ellos ejecutan software que verifica las transacciones, mantiene una base de datos, se sincroniza con otros nodos y mantiene la estabilidad y la fiabilidad para crear una red escalable, rápida y segura. Los nodos se controlan en función de su (pasado) comportamiento (métricas), lo que significa que la cantidad de trabajo que se realiza no es tan importante como la calidad del servicio. Teóricamente, los nodos que intenten manipular activamente los datos serán capturados y eliminados de la red *p2p*.

Existen una serie de características para que la red funcione.

1.2.3. Servidor de Infraestructura NEM (NIS)

Cada nodo está asociado con una única cuenta principal, que se utiliza para autenticar las respuestas de ese nodo. Esto evita que un atacante se haga pasar por un nodo sin tener su clave privada, incluso si puede falsificar la dirección IP del nodo.

Cada nodo NIS tiene la siguiente configuración:

- nis.nodeLimit: el número de otros nodos a los que el nodo local debe transmitir información
- nis.timeSyncNodeLimit: el número de otros nodos que el nodo local utiliza para sincronizar su sección de reloj.

1.2.4. Protocolo de nodo

Los nodos NIS se comunican entre ellos utilizando un formato binario patentado de forma predeterminada. Este formato minimiza el ancho de banda de la red al compactar las solicitudes y el procesamiento al reducir el costo de serialización y deserialización. De hecho, todas las API de NIS admiten solicitudes codificadas en el formato binario patentado de NEM o JSON. Para evitar los ataques basados en suplantación, los nodos NIS se comprometen en un apretón de manos de dos partes al comunicarse:

1. El nodo local envía los datos de solicitud y una carga útil aleatoria de 64 bytes al nodo remoto.
2. El nodo remoto firma la carga aleatoria y envía la firma junto con los datos de respuesta solicitados al nodo local.
3. El nodo local verifica la firma y solo procesa los datos de respuesta si puede verificar que el nodo remoto firmó la carga aleatoria.

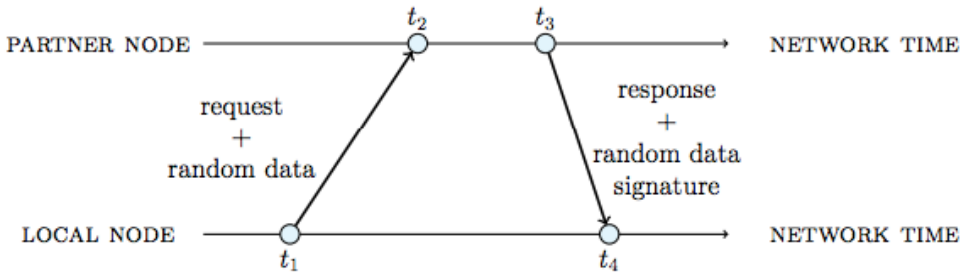


Ilustración 9 Comunicación entre nodo local y nodo asociado

1.2.5. Inicio del nodo

Cuando se lanza un nodo NIS, el nodo procesa la cadena de bloques y almacena algunos datos en la memoria para mejorar el rendimiento en línea. Una vez que finaliza el procesamiento, el nodo aún no está conectado a la red porque aún no se ha iniciado.

Un nodo no arrancado no está asociado con una cuenta. Esto evita que pueda firmar respuestas e impide que otros nodos puedan verificar su identidad.

Para iniciar un nodo, la clave privada de una cuenta NEM se debe suministrar al nodo. Esta cuenta es la cuenta principal asociada con el nodo. Se puede usar una cuenta delegada para arrancar un nodo con el fin de proteger mejor la clave privada de la cuenta real.

1.2.6. Descubrimiento de nodo

Una vez que se inicia un nodo, se conecta a la red NEM y comienza a compartir información con otros nodos. Inicialmente, el nodo solo conoce los nodos pre-confiables.

Con el tiempo, el nodo se da cuenta de más nodos en la red. Normalmente hay dos formas en que esto sucede: mediante un anuncio o una actualización.

1.2.6.1. Anuncio

Periódicamente, un nodo se anuncia a sus nodos asociados actuales e incluye su información de experiencia local en la solicitud. Si el socio no conoce el nodo, el compañero lo marca como activo y actualiza las experiencias del nodo. Si el socio conoce al nodo, el socio solo actualiza las experiencias del nodo, pero no cambia su estado.

1.2.6.2. Actualización

Periódicamente, un nodo le pide a sus socios actuales que proporcionen información actualizada sobre sí mismos y sobre el estado de la red.

Primero, el nodo solicita información de actualización sobre el nodo remoto. Si tiene éxito, el nodo local actualizará el punto final del control remoto (necesario para admitir direcciones IP dinámicas) y los metadatos. Este paso falla si ocurre cualquiera de los siguientes:

- El nodo remoto devuelve una solicitud válida de un nodo diferente
- El nodo remoto no es compatible con el nodo local (por ejemplo, el nodo remoto es testnet pero el nodo local es mainnet)

En caso de éxito, se solicita al nodo remoto que proporcione una lista de todos sus nodos asociados actuales. Para evitar que un nodo maligno proporcione información incorrecta sobre los nodos buenos y los haga aparecer en la lista negra, el nodo local intenta contactar directamente a cada nodo asociado denunciado. Es importante tener en cuenta que los metadatos de un nodo solo se actualizarán con los metadatos proporcionados por ese nodo.

1.2.6.3. Selección Nodo

Con el tiempo, como resultado del proceso de descubrimiento de nodos, el nodo local se dará cuenta de más nodos en la red. Eventualmente, la cantidad de nodos conocidos (incluidos los nodos bien conocidos y otros) será mucho mayor que la cantidad de nodos asociados.

Periódicamente, un nodo recalcula sus nodos asociados. Cuando se produce este nuevo cálculo, todos los nodos conocidos son elegibles para convertirse en un nodo asociado, incluidos los nodos que se han detectado como ocupados.

Los nodos conocidos se ponderan según sus valores de confianza y los nodos asociados se seleccionan aleatoriamente entre ellos. Los nodos con valores de confianza más grandes (que tienen más probabilidades de ser buenos nodos) tienen más probabilidades de ser elegidos como socios. Los nodos con los que se ha interactuado mínimamente suelen tener valores de confianza cercanos a 0. Con el fin de dar a estos nodos la oportunidad de demostrar su valía y generar confianza, reciben un pequeño impulso de confianza para que tengan la oportunidad de ser seleccionados y participar en la red. El 30% de la confianza se distribuye de manera uniforme entre los nodos con menos de 10 interacciones.

Para garantizar que la red esté conectada, se realiza un ajuste en el proceso aleatorio. Si el nodo local es un nodo bien conocido, también está conectado con todos los nodos

conocidos en línea. Si el nodo local no es un nodo bien conocido, también está conectado con un nodo conocido, en línea y al azar. Esto asegura que todos los nodos se asocian activamente con al menos un nodo conocido.

1.2.6.4. Cosecha

La cosecha en NEM es el proceso de generar bloques y obtener las tarifas de transacción en ese bloque como recompensa por el trabajo contribuido. El algoritmo POI (Prueba de importancia) determina quién puede generar un bloque (o más precisamente: qué bloque generado se considera válido). Para poder cosechar, la cuenta necesita un saldo establecido de al menos 10,000 XEM.

Puede comparar la extracción con la minería en Bitcoin, aunque con la cosecha no crea nuevas monedas XEM, sino que solo gana las tarifas de transacción.

NEM tiene dos métodos diferentes para cosechar: cosecha local y delegada.

- **Cosecha Local**

  En la cosecha local, la clave privada del propietario de la cuenta es traspasada al NIS (Nem Infrastucture Server – Node) que se ejecuta localmente, para firmar los bloques generados. La clave privada nuca es publicada en la red. Cuando un bloque es formado por el cosechador, un nuevo bloque se añade a la cadena y todas las comisiones recolectadas por ese bloque se entregan a la cuenta que hizo la cosecha.

- **Cosecha Delegada**

  Para realizar este método de cosecha es necesario conectarse con un NIS remoto, se transmiten las puntuaciones del PoI al supernodo y la clave privada de forma segura, de esta manera cada vez que el nodo remoto valide un bloque nuevo, un porcentaje de las comisiones son adjudicadas al usuario. Con esta forma de cosechar no es necesario dejar encendido el ordenador ya que otras máquinas estarían realizando el proceso.

1.2.6.5. Prueba de Importancia

Prueba de importancia es un algoritmo de consenso de cadena de bloques que fue introducido por primera vez por NEM . Es el mecanismo que se utiliza para determinar qué participantes de la red (nodos) son elegibles para agregar un bloque a la cadena de bloques, mediante la cosecha. Las cuentas con una puntuación de importancia más alta tendrán una mayor probabilidad de ser elegidas para cosechar un bloque. Para poder ser elegible para el cálculo de importancia, el protocolo NEM requiere que una cuenta tenga al menos 10,000 XEM con derechos de propiedad para ser elegibles para la cosecha.

La prueba de importancia se puede considerar como un novedoso algoritmo de consenso porque, a diferencia de los mecanismos de consenso existentes, como la prueba de participación, busca tener en cuenta el apoyo general de la red. Por ejemplo, con una prueba de participación, se puede argumentar que recompensa a los acaparadores de monedas. Según el modelo de prueba de participación, los nodos se limitan a 'minar' un porcentaje de transacciones que refleja su participación en una criptomoneda. Por ejemplo, una prueba de minero de estaca que posee el 10% de una criptomoneda sería capaz de extraer el 10% de los bloques en la red. La limitación de este modelo de consenso es que incentiva a los nodos de la red a guardar sus monedas, en lugar de gastarlas. También produce un escenario en el que "los ricos se hacen más ricos", ya que los grandes poseedores de monedas pueden explotar un mayor porcentaje de bloques en la red.

Prueba de importancia busca superar los problemas que se pueden encontrar en el modelo de prueba de participación identificando el soporte general de una cuenta de la red. NEM lo hace contabilizando tres factores: derechos de adquisición, socios de transacción y cantidad y tamaño de transacciones en los últimos 30 días.

Adquisición

- Se requiere un mínimo de 10,000 monedas conferidas para la cosecha.
- Cuanto mayor sea el número de monedas con derechos, mayor será la prueba de la puntuación de importancia de la cuenta.
- La prueba de importancia solo cuenta las monedas que han estado en una cuenta durante un número determinado de días.

Socios de transacción

- La prueba de importancia recompensa a los usuarios que realizan transacciones con otras cuentas NEM en la red.
- Los usuarios no pueden jugar en la red intercambiando cuentas entre cuentas, el algoritmo solo tiene en cuenta las transferencias netas a lo largo del tiempo.

Número y tamaño de las transacciones en los últimos 30 días

- Cada transacción (por encima de un tamaño mínimo) contribuye a una prueba de puntaje de importancia de la cuenta.
- Las transacciones más grandes y más frecuentes tienen un mayor impacto en la prueba de puntaje de importancia.

1.2.7. Reputación de Nodos

Las redes P2P tienen la gran ventaja de ser robustas porque no se pueden cerrar eliminando una sola entidad. Sin embargo, los participantes de la red son anónimos y cualquiera puede unirse. Esto hace que sea muy fácil inyectar nodos hostiles en la red que difundan información no válida o traten de perturbar la red de alguna manera. Es necesario identificar los nodos hostiles y reducir la comunicación con ellos. NEM implementa un algoritmo de reputación basado en EigenTrust++.

Este algoritmo funciona como un administrador de reputación. Considera que cada par o nodo dentro de la red punto a punto es veraz, confiable, no malicioso y válido. Como una forma de ilustrar, el nodo A confía en los nodos H e I, y luego también confía en cualquier nodo que sea confiable para el nodo H. En este caso, es el nodo F. Por lo tanto, cada nodo debe ser confiable y las transacciones validadas correctamente, haciendo que la red sea altamente segura.

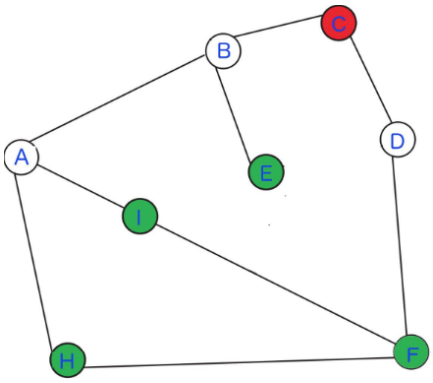


Ilustración 10 Confían en una red de nodos

Nodo A confía en los nodos H, I y F. Nodo B confía en nodo E, pero no confía en nodo C. Nodo D confía en los nodos F, I y H.

Los nodos en la red NEM interactúan entre sí al compartir información sobre entidades como transacciones y bloques de transacciones. Un nodo puede transmitir nuevas entidades a otros nodos o solicitar otros nodos para esas entidades. Después de recibir información de un nodo remoto, un nodo puede verificar la validez de la información y verificar si la información es utilizable. Cada nodo puede decidir si una interacción (o 'llamada') debe verse como un éxito (se recibió nueva información válida), neutral (válida, pero ya conocida, se recibió información) o error (se recibió información inválida). Cada nodo realiza un seguimiento de los resultados de todas sus interacciones con el nodo j en su mapa de experiencia al contar sus interacciones exitosas y fallidas. Las interacciones neutrales son ignoradas. Estas interacciones entre nodos y sus respectivas interpretaciones sirven para definir la confianza local de un nodo, así como la confianza que tiene con otros nodos.

Cada cierto tiempo, los nodos transmiten sus valores de confianza locales a otros nodos. Habiendo recibido los valores de confianza locales de otros nodos, el nodo i puede calcular un valor de confianza agregado para el nodo k pesando el nodo de confianza local j tiene en el nodo k con su propia confianza en el nodo j.

Este proceso de verificación hace que NEM se implemente de manera eficiente y se ajusta a la forma en que el algoritmo Eigentrust administra cada nodo a sus más altas pruebas de seguridad y confiabilidad posibles. La transferencia de datos de un nodo a otro será más rápida.

1.2.8. Sincronización de Tiempo P2P

NEM depende de marcas de tiempo para transacciones y bloques. Idealmente, todos los nodos en la red deberían estar sincronizados con respecto al tiempo. Aunque la mayoría de los sistemas operativos modernos tienen sincronización de tiempo integrada, los nodos aún pueden tener relojes locales que se desvían del tiempo real en más de un minuto. Esto hace que esos nodos rechacen transacciones o bloques válidos, lo que les imposibilita la sincronización con la red.

Por lo tanto, es necesario contar con un mecanismo de sincronización para garantizar que todos los nodos acuerden el tiempo.

Se puede hacer uso de un protocolo NTP (Network Time Protocol) para determinar el tiempo en la red, pero se debe utilizar servidores externos. NEM usa un protocolo personalizado para ser completamente independiente de cualquier entidad externa.

1.2.8.1. Recolectando Muestras

Cada nodo en la red maneja un número entero llamado *offset* que se establece en 0 al inicio. El tiempo del sistema local en milisegundos incrementado por el desplazamiento (que puede ser negativo) es el tiempo de red (otra vez en milisegundos) del nodo.

Una vez que se completa el inicio de un nodo, el nodo (en lo sucesivo denominado *nodo local)* selecciona hasta 20 nodos asociados para realizar una ronda de sincronización de tiempo. Solo los nodos que exponen una importancia mínima se consideran socios.

Para todos los socios seleccionados, el nodo local envía una solicitud pidiéndole al socio su hora de red actual. El nodo local recuerda las marcas de tiempo de la red cuando se envió la solicitud y cuando se recibió la respuesta. Cada nodo asociado responde con una muestra que contiene la marca de tiempo de la llegada de la solicitud y la marca de tiempo de la respuesta. El nodo asociado usa su propio tiempo de red para crear las marcas de tiempo.

Usando las marcas de tiempo, el nodo local puede calcular el tiempo de ida y vuelta

$$rtt = (t_4 - t_1) - (t_3 - t_2)$$

y luego estimar la compensación o entre el tiempo de red utilizado por los dos nodos como

$$o = t_2 - t_1 - \frac{rtt}{2}$$

Esto se repite para cada socio de sincronización de tiempo hasta que el nodo local tenga una lista de estimaciones de compensación.

1.2.8.2. Filtros para Eliminar Datos Incorrectos

Puede haber malas muestras debido a varias razones:

- Un nodo malicioso puede proporcionar marcas de tiempo incorrectas.
- Un nodo honesto puede tener un reloj lejos del tiempo real sin saberlo y sin haberse sincronizado todavía.
- El tiempo de ida y vuelta puede ser muy asimétrico debido a problemas de internet o uno de los nodos está muy ocupado. Esto se conoce como asimetría de canal y no se puede evitar.

Se aplican filtros que intentan eliminar las muestras malas. El filtrado se realiza en 3 pasos:

- Si la respuesta de un compañero no se recibe dentro de un marco de tiempo esperado (es decir, si t 4 - t 1 > 1000ms) la muestra se descarta.
- Si el desplazamiento calculado no está dentro de ciertos límites, la muestra se descarta. Los límites permisibles disminuyen a medida que aumenta el tiempo de actividad de un nodo. Cuando un nodo se une por primera vez a la red, tolera una compensación alta para ajustarse al consenso ya existente del tiempo de red dentro de la red. A medida que pasa el tiempo, el nodo se vuelve menos tolerante con respecto a las compensaciones informadas. Esto garantiza que los nodos maliciosos que informan grandes compensaciones se ignoran después de un tiempo.

- Las muestras restantes se ordenan por su desplazamiento y luego se recortan alfa en ambos extremos. En otras palabras, en ambos lados se descarta una cierta porción de las muestras.

1.3. Características API NEM

La *blockchain* de NEM propone un API RESTful que, en lugar de cargar toda la lógica de la aplicación en la cadena de bloques, puede usar sus funciones probadas a través de llamadas API para:

- Transferencia y almacenamiento de valor.
- Autorización.
- Trazabilidad.
- Autenticación.

La lógica de negocio permanece fuera de cadena. Esto reduce el riesgo inherente de inmutabilidad, ya que podría cambiar el proceso cuando sea necesario. Por ejemplo, después de recibir comentarios de los usuarios, es posible que deba cambiar su software agregando funcionalidad nueva o inesperada. Entre las funcionalidades podemos encontrar:

- **Cuenta**

  Una cuenta es un par de claves (clave pública y privada) asociada con un estado mutable almacenado en la cadena de bloques NEM. En otras palabras, tiene una caja de depósito en la cadena de bloques, que solo usted puede modificar con su par de claves. Como su nombre lo indica, la clave privada debe mantenerse en secreto en todo momento. Cualquier persona con acceso a la clave privada, en última instancia, tiene el control sobre la cuenta.

  Piense en las cuentas NEM como un contenedor de activos en la cadena de bloques. Una cuenta podría representar un depósito de tokens, como la mayoría de las *blockchains*. Sin embargo, también podría representar un solo objeto que debe ser único y actualizable: un paquete que se enviará, una escritura a una casa o un documento para ser notariado.

  Las cuentas tienen las siguientes propiedades:

- **Llave privada:** Una clave privada es una clave para una cuenta. Cualquier persona que tenga la clave privada de una cuenta puede iniciar cualquier acción relacionada con la cuenta.
- **Llave pública:** La clave pública se puede utilizar para verificar las firmas de la cuenta. La clave pública se almacena en la cadena de bloques con la primera transacción emitida. Una cuenta que no ha emitido ninguna transacción tiene su campo de clave pública vacío.
- **Dirección**: Cada cuenta tiene una dirección única . Por lo general, compartirá la dirección derivada, ya que es más corta y recopila más información.
- **Mosaicos**: La cantidad de mosaicos diferentes que posee la cuenta.
- **Importancia**: La relación entre la cantidad de mosaicos de recolección que posee la cuenta y el suministro total de mosaicos.

- **Cuenta Multifirma**

Una cuenta NEM se puede convertir en multifirma. A partir de ese momento, la cuenta no puede anunciar transacciones por sí misma. Se requerirán otras cuentas para anunciar transacciones para ellos. Estas otras cuentas son los cosignatarios de multifirma.

Sin embargo, no siempre es necesario forzar a todos los cosignatarios a firmar la transacción. NEM permite establecer el número mínimo de acuerdos consignatarios. Estas propiedades se pueden editar después para adaptarse a casi todas las necesidades. La implementación actual de multisig de NEM es "M-of-N" . Esto significa que M puede ser cualquier número igual o menor que N, es decir, 1 de 4, 2 de 2, 4 de 9, 9 de 10 y así sucesivamente.

La cantidad de cosignaturas mínimas para aprobar transacciones y eliminar cosignatarios es editable.

Algunas consideraciones importantes a tener en cuenta:

- Las cuentas multisig pueden tener hasta 10 cosignatarios.
- Una cuenta puede ser cosignataria de 5 cuentas multifirma.

- Las cuentas multifirma pueden tener como cosignatario otra multifirma, hasta 3 niveles. Las cuentas de niveles múltiples de múltiples niveles agregan la lógica "AND/OR" a las transacciones de firma múltiple.

- **Namespace**

Los “Namespace” le permiten crear un lugar único en la cadena para su negocio y sus activos en la cadena de bloques NEM.

Un “Namespace” comienza con un nombre que usted elija, similar a un nombre de dominio de Internet. Si una cuenta crea un espacio de nombres, aparecerá como único en el ecosistema NEM.

Una cuenta puede vincular un nombre registrado (espacio de nombres o espacio de subnombre) con una cuenta o un identificador de mosaico.

Subnamespaces

En internet, un dominio puede tener un subdominio. En NEM, los espacios de nombres pueden tener espacios de subnombre.

Puede crear varios subnamespaces con el mismo nombre en diferentes espacios de nombres. Por ejemplo, puede crear los espacios de subnombre foo.bar y foo2.bar.

Los espacios de nombres pueden tener hasta 3niveles, un espacio de nombres y sus dos niveles de dominios de espacio de subnombre.

Alias

Las transacciones de alias hacen que las direcciones largas sean recordables y los mosaicos reconocibles.

El creador del espacio de nombres puede editar el enlace entre un espacio de nombres y un activo. La relación de alias para una transacción dada se puede recuperar más tarde de los recibos del bloque .

Restricciones:

- Una cuenta solo puede asociar un nombre con una cuenta o mosaico, pero esos pueden tener muchos alias vinculados.

- Una cuenta puede asignar un nombre a cualquier cuenta que permita recibir AddressNamespaceTransactions . En contraste, si la cuenta quiere asignar el alias a un mosaicId, debe ser el creador del mosaico.

- **Mosaico**

Los mosaicos son parte de lo que hace que el Sistema de Activos Inteligentes sea único y flexible. Son activos fijos en la cadena de bloques NEM que pueden representar un conjunto de múltiples cosas idénticas que no cambian.

Un mosaico podría ser un token, pero también podría ser una colección de activos más especializados, como puntos de recompensa, acciones, firmas, indicadores de estado, votos o incluso otras monedas.

Cada mosaico tiene un identificador único y un conjunto de propiedades configurables. Durante la creación del mosaico, puede definir:

Propiedad	**Tipo**	**Descripción**
Divisibilidad	Entero	Determina hasta qué punto decimal se puede dividir el mosaico. La divisibilidad de 3 significa que un mosaico se puede dividir en partes más pequeñas de 0.001 mosaicos. La divisibilidad debe estar en el rango de 0 y 6.
Duración	Entero	Especifica el número de bloques confirmados para los que se alquila el mosaico. Para crear mosaicos que no caducan, deje esta propiedad sin definir.
Suministro inicial	Entero	Indica la cantidad de mosaico en circulación. El suministro inicial debe estar en el rango de 0 y 9,000,000,000.
Suministro mutable	Booleano	Si se establece en verdadero, la fuente de mosaico puede cambiar en un momento posterior. De lo contrario, el suministro de mosaico permanece inmutable.

Propiedad	Tipo	Descripción
Transferibilidad	Booleano	Si se establece en verdadero, el mosaico se puede transferir entre cuentas arbitrarias. De lo contrario, el mosaico solo se puede transferir de nuevo al creador del mosaico.

- **Transferencia Transacción**

Las transacciones de transferencia se utilizan para enviar mosaicos entre dos cuentas. Pueden contener un mensaje de 1023caracteres de longitud (Ilustración 11).

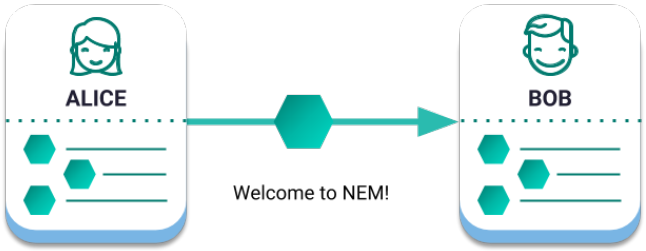


Ilustración 11 Alice envía 10 cat. Moneda a Bob

- **Transacción Agregada**

Las transacciones agregadas combinan varias transacciones en una, permitiendo intercambios sin confianza y otras lógicas avanzadas. NEM hace esto generando un contrato inteligente desechable de una sola vez. Cuando todas las cuentas involucradas han firmado la transacción agregada, todas las transacciones internas se ejecutan al mismo tiempo.

Agregado completo

Una transacción agregada se completa cuando todos los participantes requeridos la han firmado.

Los firmantes pueden firmar la transacción sin utilizar la cadena de bloques. Una vez que tiene todas las firmas requeridas, uno de ellos puede anunciarlo en la red. Si la

configuración de la transacción interna es válida y no hay un error de validación, las transacciones se ejecutarán al mismo tiempo.

Las transacciones completas agregadas permiten agregar más transacciones por bloque al recopilar varias transacciones internas. Por ejemplo (Ilustración 12), Dan anuncia una transacción agregada que combina dos transacciones de transferencia.

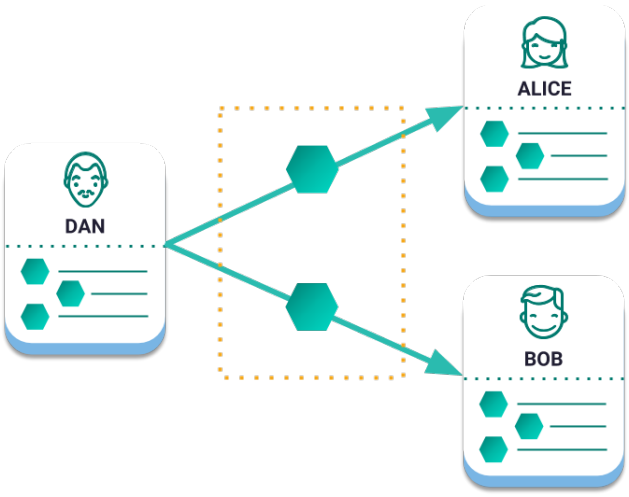


Ilustración 12 Envío de pagos con transacciones completas agregadas

Como él es el único firmante requerido, la transacción se considera completa después de que firmó. Después de anunciarlo a la red, Alice y Bob recibirán los mosaicos al mismo tiempo.

Agregado unido

Una transacción agregada se vincula cuando requiere firmas de otros participantes. Antes de enviar una transacción consolidada agregada , una cuenta primero debe anunciar una transacción de bloqueo de hash y obtener su confirmación con al menos .10 cat.currency.

Una vez que se anuncia un agregado agregado, alcanza un estado parcial y notifica su estado a través de WebSockets o llamadas a la API HTTP.

Cada vez que un cosignatario firma la transacción y anuncia una cosignatura consolidada agregada , la red verifica si todos los cosignatarios requeridos han firmado.

Cuando se adquieren todas las firmas, la transacción cambia a un estado no confirmado hasta que la red lo incluye en un bloque.

- **Intercambios de cadenas cruzadas**

Un intercambio entre cadenas permite el intercambio de tokens a través de diferentes *blockchains*, sin utilizar una parte intermedia (Ilustración 13) en el proceso.

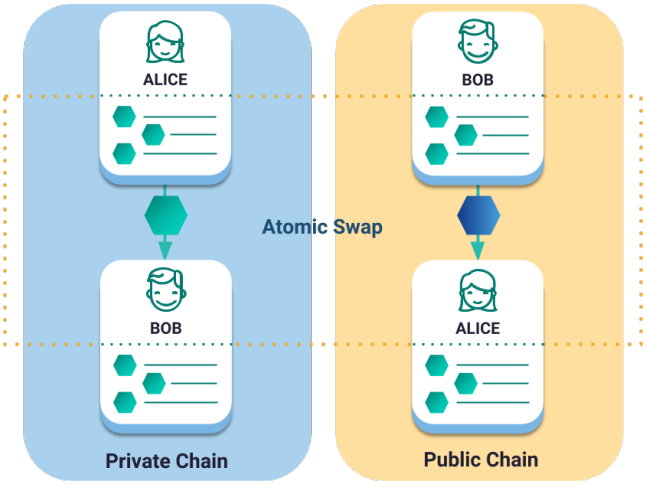


Ilustración 13 Intercambio de cadenas atómicas entre redes públicas y privadas.

Para crear un entorno de confianza para un intercambio, se requiere un tipo de transacción específico que se conoce comúnmente como Contrato de bloqueo de tiempo hashed (HTLC). Dos componentes adicionales caracterizan este tipo de transacción: hashlocks y timelocks . Se puede encontrar una explicación detallada en el Wiki de Bitcoin.

En otras palabras, para reducir el riesgo de contraparte, el receptor de un pago debe presentar una prueba de la transacción que debe ejecutarse. De no hacerlo, los fondos bloqueados se liberan una vez que se alcanza la fecha límite, incluso si solo un actor no está de acuerdo. La siguiente figura ilustra el protocolo de intercambio de cadenas cruzadas.

2. Node.js

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación ECMAScript,

asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como, por ejemplo, servidores web. Fue creado por Ryan Dahl en 2009 y su evolución está apadrinada por la empresa Joyent, que además tiene contratado a Dahl en plantilla.

2.1. Aspectos Técnicos

2.1.1. Concurrencia

Node.js funciona con un modelo de evaluación de un único hilo de ejecución, usando entradas y salidas asíncronas las cuales pueden ejecutarse concurrentemente en un número de hasta cientos de miles sin incurrir en costos asociados al cambio de contexto. Este diseño de compartir un único hilo de ejecución entre todas las solicitudes atiende a necesidades de aplicaciones altamente concurrentes, en el que toda operación que realice entradas y salidas debe tener una función callback. Un inconveniente de este enfoque de único hilo de ejecución es que Node.js requiere de módulos adicionales como *cluster* para escalar la aplicación con el número de núcleos de procesamiento de la máquina en la que se ejecuta.

2.1.2. Motor V8

V8 es el entorno de ejecución para JavaScript creado para Google Chrome. Es software libre desde 2008, está escrito en C++ y compila el código fuente JavaScript en código de máquina en lugar de interpretarlo en tiempo real.

Node.js contiene libuv para manejar eventos asíncronos. Libuv es una capa de abstracción de funcionalidades de redes y sistemas de archivo en sistemas Windows y sistemas basados en POSIX como Linux, Mac OS X y Unix.

El cuerpo de operaciones de base de Node.js está escrito en JavaScript con métodos de soporte escritos en C++.

2.1.3. Módulos

Node.js incorpora varios "módulos básicos" compilados en el propio binario, como por ejemplo el módulo de red, que proporciona una capa para programación de red asíncrona y

otros módulos fundamentales, como por ejemplo Path, FileSystem, Buffer, Timers y el de propósito más general Stream. Es posible utilizar módulos desarrollados por terceros, ya sea como archivos ".node" precompilados, o como archivos en javascript plano. Los módulos Javascript se implementan siguiendo la especificación CommonJS para módulos, utilizando una variable de exportación para dar a estos scripts acceso a funciones y variables implementadas por los módulos.

Los módulos de terceros pueden extender node.js o añadir un nivel de abstracción, implementando varias utilidades middleware para utilizar en aplicaciones web, como por ejemplo los frameworks *connect* y *express*. Pese a que los módulos pueden instalarse como archivos simples, normalmente se instalan utilizando el *Node Package Manager (npm)*que nos facilitará la compilación, instalación y actualización de módulos, así como la gestión de las dependencias. Además, los módulos que no se instalen el directorio por defecto de módulos de Node necesitarán la utilización de una ruta relativa para poder encontrarlos. El wiki Node.js proporciona una lista de varios de los módulos de terceros disponibles.

2.2. Desarrollo homogéneo entre cliente y servidor

Node.js puede ser combinado con una base de datos documental (por ejemplo, MongoDB o CouchDB) y JSON lo que permite desarrollar en un entorno de desarrollo JavaScript unificado. Con la adaptación de los patrones para desarrollo del lado del servidor tales como MVC y sus variantes MVP, MVVM, etc. Node.js facilita la reutilización de código del mismo modelo de interfaz entre el lado del cliente y el lado del servidor.

2.3. Bucle de eventos

Node.js se registra con el sistema operativo y cada vez que un cliente establece una conexión se ejecuta un callback. Dentro del entorno de ejecución de Node.js, cada conexión recibe una pequeña asignación de espacio de memoria dinámico, sin tener que crear un hilo de ejecución. A diferencia de otros servidores dirigidos por eventos, el bucle de gestión de eventos de Node.js no es llamado explícitamente, sino que se activa al final de cada ejecución de una función callback. El bucle de gestión de eventos se termina cuando ya no quedan eventos por atender.

3. ReactJS

ReactJS es una biblioteca Javascript de código abierto para crear interfaces de usuario con el objetivo de animar al desarrollo de aplicaciones en una sola página. Es mantenido por Facebook, Instagram y una comunidad de desarrolladores independientes y compañías.

React intenta ayudar a los desarrolladores a construir aplicaciones que usan datos que cambian todo el tiempo. Su objetivo es ser sencillo, declarativo y fácil de combinar. La biblioteca sólo maneja la interfaz de usuario en una aplicación; está construida únicamente para utilizar el patrón de diseño modelo–vista–controlador (MVC), y puede ser utilizada conjuntamente con otras bibliotecas de Javascript o más grandes #MVC. También puede ser utilizado con las extensiones de React-based que se encargan de las partes no-UI (no gráficas) de una aplicación web. Mantiene un virtual DOM propio, en lugar de confiar solamente en el DOM del navegador. Esto deja a la biblioteca determinar qué partes del DOM han cambiado comparando contenidos entre la versión nueva y la almacenada en el virtual DOM, y utilizando el resultado para determinar cómo actualizar eficientemente el DOM del navegador.

Fue creada por Jordan Walke, un ingeniero de software en Facebook. Influenciado por XHP, un marco de componentes de HTML para PHP.3

4. JSON Web Encryption

La especificación JWE (cifrado web JSON) estandariza la manera de representar un contenido cifrado en una estructura de datos basada en JSON. Define dos formas serializadas para representar la carga útil encriptada: la serialización compacta JWE y la serialización JWE JSON . Estas dos técnicas de serialización se analizan en detalle en las secciones a continuación. Al igual que en JWS, el mensaje que se va a cifrar utilizando el estándar JWE no debe ser una carga útil JSON, puede ser cualquier contenido

JWE Compact Serialización

Con la serialización compacto JWE, un contador JWE está construido con cinco componentes clave, cada uno separado por un punto (.): Cabecera JOSE , JWE clave cifrada , JWE vector de

inicialización , JWE adicional de datos de autenticación (AAD) , JWE texto cifrado y autenticación JWE Tag.

El encabezado JOSE es el primer elemento del token JWE producido bajo la serialización compacta. La estructura del encabezado JOSE es la misma, como vimos en JWS, salvo en algunas excepciones. La especificación JWE introduce dos nuevos elementos (enc y zip), que se incluyen en el encabezado JOSE del token JWE, además de lo que está definido por la especificación de la firma web JSON (JWS).

Para entender la sección de clave cifrada de JWE de JWE, primero debemos entender cómo se encripta una carga útil JSON. El elemento enc del encabezado JOSE define el algoritmo de cifrado de contenido y debe ser un algoritmo de cifrado autenticado simétrico con datos asociados (AEAD) . El elemento alg del encabezado JOSE define el algoritmo de cifrado para cifrar la clave de cifrado de contenido (CEK) . Este algoritmo también se puede definir como el algoritmo de ajuste de clave, ya que envuelve el CEK .

Algunos algoritmos de cifrado, que se utilizan para el cifrado de contenido requieren un vector de inicialización, durante el proceso de cifrado. El vector de inicialización es un número generado aleatoriamente, que se utiliza junto con una clave secreta para cifrar los datos. Esto agregará aleatoriedad a los datos cifrados, lo que evitará la repetición, incluso los mismos datos se cifran con la misma clave secreta una y otra vez. Para descifrar el mensaje en el extremo del destinatario del token, debe conocer el vector de inicialización, por lo tanto, se incluye en el token JWE, bajo el elemento **Vector de inicialización JWE** . Si el algoritmo de cifrado de contenido no requiere un vector de inicialización, entonces el valor de este elemento debe mantenerse vacío.

El cuarto elemento del token JWE es el valor codificado en base64url del texto cifrado JWE. El texto cifrado JWE se calcula cifrando la carga útil JSON de texto sin formato utilizando la ***Clave de cifrado de contenido (CEK)*** , el ***vector de inicialización JWE*** y el valor de ***Datos de autenticación adicional (AAD)*** , con el algoritmo de cifrado definido por el elemento de encabezado ***enc*** . El algoritmo definido por el elemento de encabezado **enc** debe ser un algoritmo de ***cifrado autenticado*** simétrico ***con datos asociados (AEAD)*** . El algoritmo ***AEAD*** , que se utiliza para cifrar la carga útil de texto sin formato, también permite especificar ***datos autenticados adicionales (AAD)*** .

El valor codificado en base64url de la **etiqueta autenticada JWE** es el elemento final del token JWE. Como se discutió antes, el valor de la etiqueta de autenticación se produce durante el proceso de cifrado ***AEAD*** , junto con el texto cifrado. La etiqueta de autenticación garantiza la integridad del texto cifrado y los ***datos autenticados adicionales (AAD)***

JWE JSON Serialización

A diferencia de la serialización compacta JWE, la serialización JWE JSON puede producir segmentación de datos cifrados en múltiples destinatarios sobre la misma carga útil JSON. La forma serializada definitiva bajo la serialización JWE JSON representa una carga útil cifrada en un objeto JSON. Este objeto JSON incluye seis elementos de nivel superior: ***protegido*** , ***no protegido*** , ***destinatarios*** , ***iv*** , ***texto cifrado*** y ***etiqueta*** . A continuación se muestra un ejemplo de un token JWE, que se serializa bajo la serialización JWE JSON.

Capitulo IV - Marco Aplicativo

Para la elaboración de este proyecto se propuso la arquitectura (Ilustración 14) compuesta por:

- Sistema Administrativo: En el cual se configura todo lo concerniente a un evento electoral.
- Aplicación de Voto: Boleta electrónica donde el votante escogerá sus candidatos y votará.
- API Voto: Un API la cual recibe y controla todas las peticiones de configuración de un evento electoral, así como también los votos emitidos, servirá de comunicación con las unidades de persistencia.
- Base de Datos Relacional (MySQL): Una base de datos relacional MySQL donde se registrará información de configuración de un evento electoral.
- Blockchain Publica NEM: Una base de datos distribuida para registrar informacion que no mutara en el tiempo.

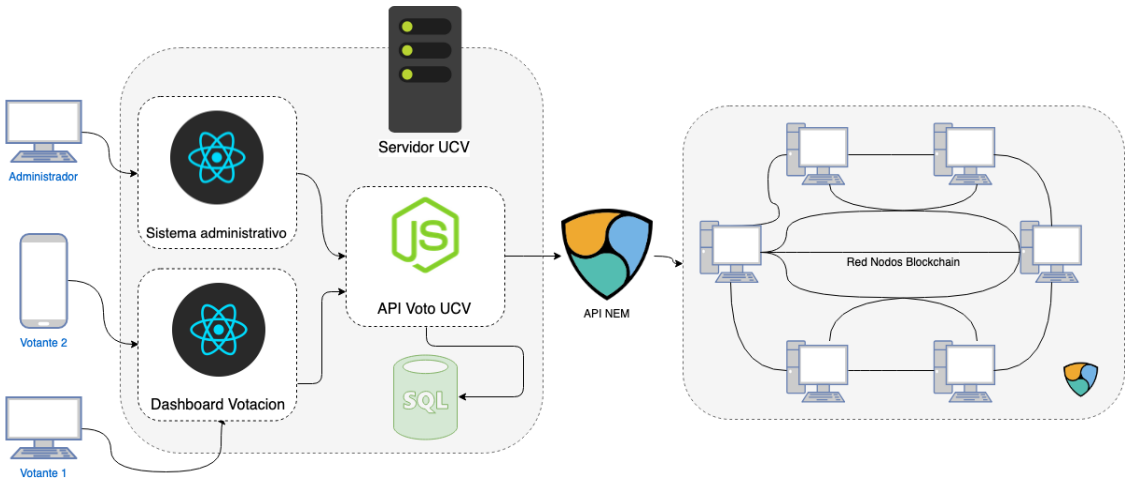


Ilustración 14 Arquitectura Sistema Voto UCV

1. Diseño de Persistencia de Datos

Para la persistencia de datos se utilizó una base de datos relacional y una blockchain publica.

1.1. Diseño Base de Datos Relacional

Se decidió diseñar una solución que se apoyara en un base de datos relacional para registrar datos que sirven para la configuración de la lógica electoral universitaria y cuyo contenido cambia con poca regularidad, como facultades, escuelas, tipo de elecciones, cargos de candidatos, usuarios que manejan el sistema, así como también para manejar la fase de

autenticación y acceso de los electores a la aplicación de voto electrónico. A continuación, se describen las tablas creadas (Ilustración 15).

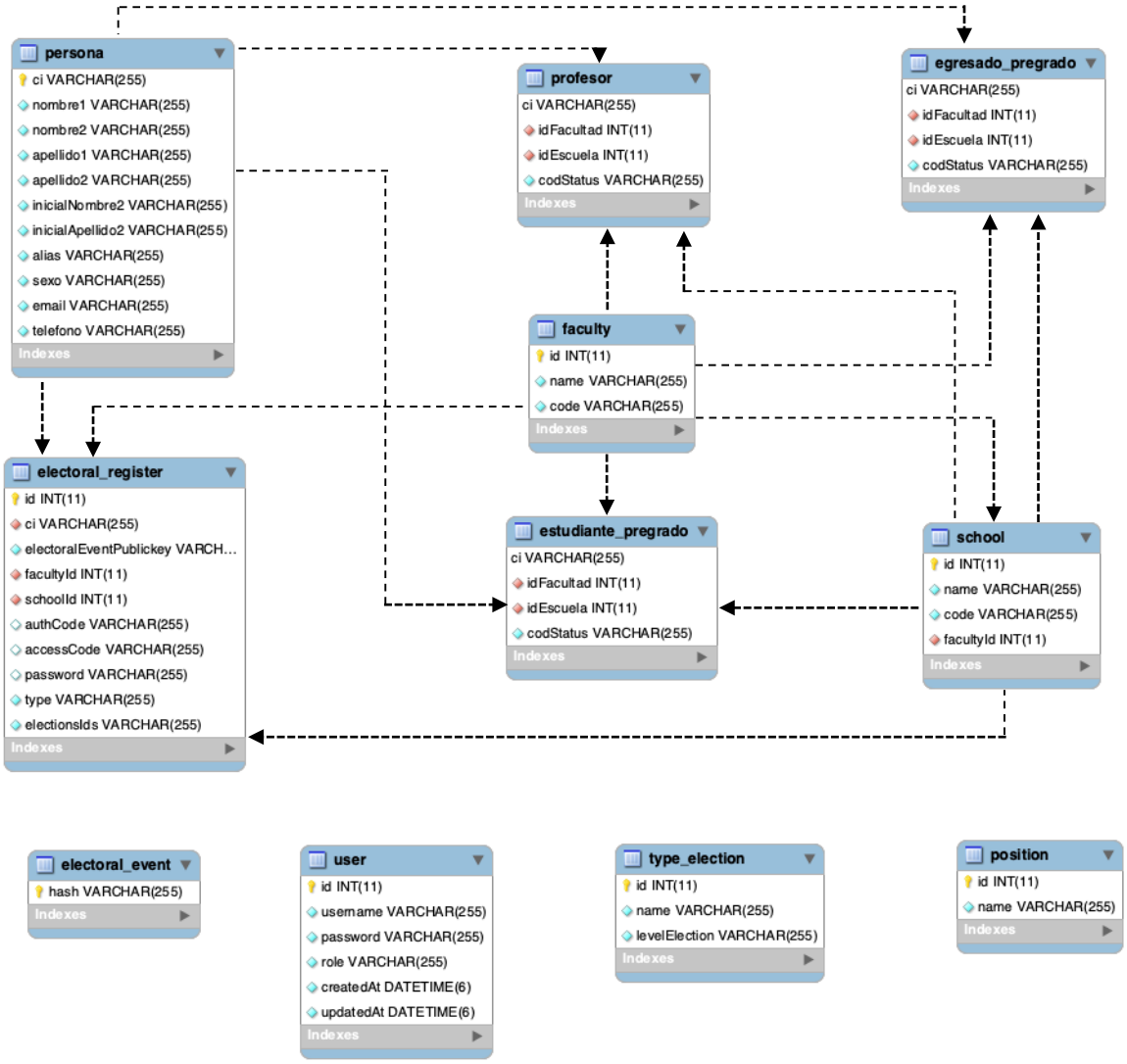


Ilustración 15 Diagrama Base de Datos

Persona

Información personal de cada individuo de la universidad

Campo	Descripción
ci	Número que identifica a una persona.
nombre1	Primer nombre.
nombre2	Segundo nombre.
apellido1	Primer apellido.
apellido2	Segundo apellido.
inicialNombre2	Inicial del segundo nombre.
inicialApellido2	Inicial del Segundo apellido.
alias	Seudónimo de la persona.
sexo	Sexo de la persona.
email	Dirección de correo electrónico.
telefono	Número telefónico.

Estudiante Pregrado

Registros de un estudiante de alguna carrera de pregrado de la universidad

Campo	Descripción
ci	Número que identifica a una persona. Referencia a la tabla “Persona”.
idFacultad	Identificador de la facultad donde estudia. Referencia a la tabla “Facultad”.
idEscuela	Identificador de la escuela donde estudia. Referencia a la tabla “Escuela”.
codStatus	Código que identifica si un estudiante puede votar.

Profesor

Registros de un profesor de alguna carrera de pregrado de la universidad.

Campo	Descripción
ci	Número que identifica a una persona. Referencia a la tabla “Persona”.
idFacultad	Identificador de la facultad donde estudia. Referencia a la tabla “Facultad”.
idEscuela	Identificador de la escuela donde estudia. Referencia a la tabla “Escuela”.
codStatus	Código que identifica si un profesor puede votar.

Egresado Pregrado

Registros de un egresado de alguna carrera de pregrado de la universidad.

Campo	Descripción
ci	Número que identifica a una persona. Referencia a la tabla “Persona”.
idFacultad	Identificador de la facultad donde estudia. Referencia a la tabla “Facultad”.
idEscuela	Identificador de la escuela donde estudia. Referencia a la tabla “Escuela”.
codStatus	Código que identifica si un egresa puede votar.

Evento Electoral

Tabla que almacena los códigos hash que identifican una transacción de creación de evento electoral.

Campo	Descripción
hash	Código que identifica una transacción de creación de evento electoral en la blockchain.

Registro Electoral

Contiene a los electores y sus respectivas elecciones que participaran en un evento electoral. Además, es utilizada para manejar la autenticación, acceso y contraseña de voto de los electores.

Campo	Descripción
id	Identificador incremental.
ci	Número que identifica a una persona. Referencia a la tabla “Persona”.
electoralEventPublickey	Llave publica del evento electoral.
facultyId	Identificador de la facultad donde estudia. Referencia a la tabla “Facultad”.
schoolId	Identificador de la escuela donde estudia. Referencia a la tabla “Escuela”
authCode	Hash código de autenticación.
accessCode	Hash código de acceso.
password	Hash contraseña de votación.
type	Tipo de elector
electionsIds	Identificadores de las elecciones que puede participar.

Facultad

Describe el nombre y código de una facultad.

Campo	Descripción
id	Identificador autoincremental.
name	Nombre de la facultad.
code	Código de la facultad

Escuela

Describe el nombre, código de una escuela además de relacionar la facultad a la que pertenece.

Campo	Descripción
id	Identificador autoincremental.
name	Nombre de la escuela.
code	Código de la escuela
facultyId	Identificador de la facultad donde pertenece. Referencia a la tabla “Facultad”.

Cargo

Contiene todos los cargos elegibles en tipo de elección.

Campo	Descripción
id	Identificador autoincremental.

name	Nombre del cargo.

Tipo Elección

Almacena los tipos de elecciones que pueden ser creadas como, por ejemplo, rector, decano, centros de estudiantes, etcétera.

Campo	Descripción
id	Identificador autoincremental.
name	Nombre del tipo de elección.
levelElection	Nivel de la elección (Universidad, Facultad, Escuela)

Usuario

Registra los accesos de los usuarios que manejaran el sistema de configuración.

Campo	Descripción
id	Identificador autoincremental.
username	Nombre de usuario para acceder al sistema.
password	Hash contraseña para acceder al sistema.
role	Rol del usuario en el sistema

1.2. Implementación de blockchain NEM

Se trabajó con la blockchain publica NEM por las bondades que ofrece descritas en NEM (New Economy Movement). Se utilizó la blockchain para el registro de datos que no son

modificables, deben persistir en el tiempo y ser de dominio público para tener una transparencia de los procesos.

Haciendo uso de las funciones integradas de NEM, el registro de los datos se realiza mediante transacciones, en su campo de mensajes. Se optó por definir un formato de fácil lectura y escritura como lo es JSON para disponer de los datos que se almacenen en la blockchain, este consta de un código que identifica de que tipo es la transacción y los datos que se almacenan.

1.2.1. Modelo de Datos

Comisión Electoral

La comisión electoral es el ente autorizado de gestionar los procesos electorales. Este es representado como una cuenta de donde se crearán todas las transacciones como crear evento, elección, activar evento electoral, entre otros.

Evento Electoral

Las cuentas se pueden utilizar para representar activos, identificando cada evento electoral de forma única y recopilando el historial de transacciones e información almacenada.

Cualquier persona podría rastrear información de un evento electoral al verificar su dirección.

Las cuentas de evento electoral no necesitan enviar transacciones, solo las reciben. Para garantizar que ninguno de ellos firme transacciones, se generó cada clave pública de manera determinista con un hash del nombre del evento electoral que represente la clave pública, sin conocer la clave privada relacionada.

$$publicKey = sha256\ (nombre\_evento\_electoral)$$

Al hacer esto, si las referencias de la base de datos se pierden, la información del evento electoral aún será recuperable de la cadena de bloques. La dirección se puede generar de nuevo proporcionando el nombre del evento electoral.

La información de las elecciones que componen el evento electoral es almacenada en la cuenta del evento electoral para posteriormente poder consultarla.

Registro Electoral

Todos los participantes que puedan votar en el evento electoral son almacenados en la blockchain, es creada una cuenta NEM para su posterior búsqueda, la misma es formada de la siguiente manera:

publicKey = sha256 (publickey_evento_electoral – Registro Electoral)

Votos Encriptados

Cada elección que compone el evento electoral tendrá asociada una cuenta NEM en donde se alojaran de manera encriptada los nombres de los candidatos seleccionados por el votante, de esta manera se garantiza que los votos solo serán consultados una vez se realice la totalización del evento electoral. Estas cuentas son creadas de forma determinista con la siguiente formula:

publicKey = sha256 (id_eleccion - Votos)

Candidato

Estas cuentas, identifican a cada candidato de una elección de forma única y recopila la transaccion de votos adjudicados, luego de finalizar el evento electoral y realizar la totalización.

Al igual que las cuentas de eventos electorales no necesitan enviar transacciones, ser almacén de votos. Se crea la cuenta con una clave pública de manera determinista usando el hash del identificador de la elección más la cedula de identidad

publicKey = sha256 (id_eleccion + cedula_identidad_candidato)

Votante

Son representados como una cuenta en donde se recibirá los mosaicos de votación y serán enviados a cada uno de los candidatos elegidos. Es creada con la llave publica del evento electoral, cedula de identidad, código de acceso y una contraseña introducida por el votante.

Voto

Los votos son representados como mosaicos, que al momento de ser creados su identificador es registrado en la cuenta del evento electoral para ser utilizado nuevamente.

2. Descripción del Proceso Electoral

El proceso electoral desarrollado en este proyecto de investigación se divide en los siguientes subprocesos:

2.1. Crear evento electoral

El administrador del sistema asigna el nombre, y las fechas que habilitan los procesos involucrados en el evento electoral (Ilustración 16).

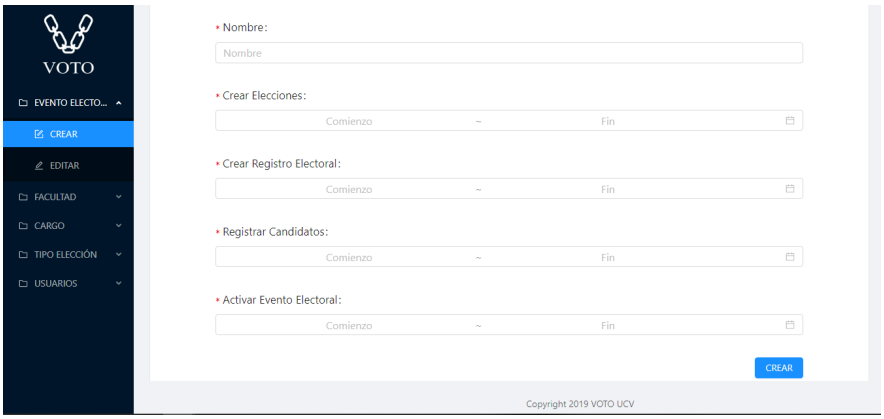


Ilustración 16 Crear Evento Electoral

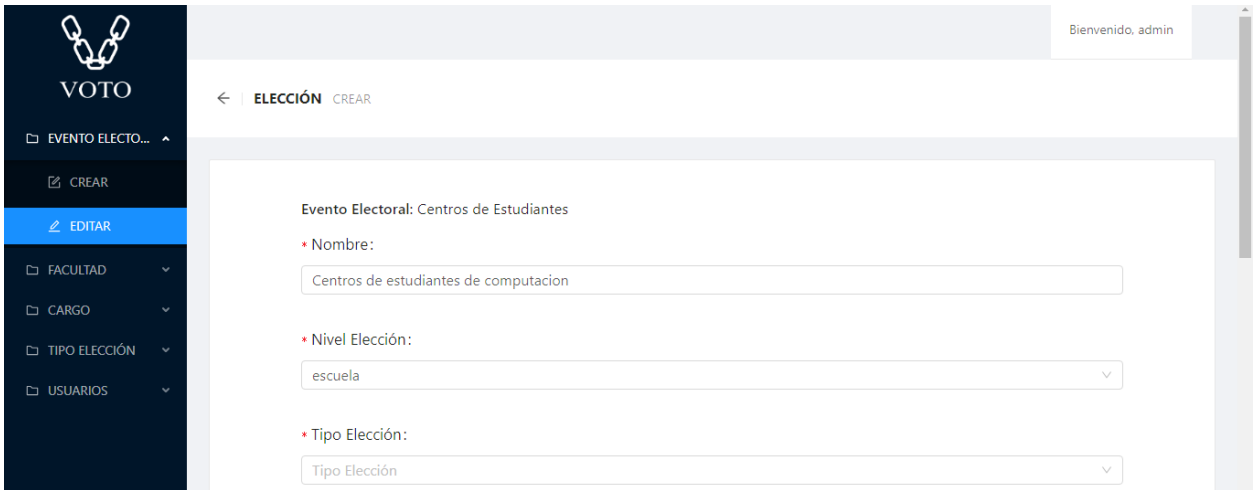
Al crear el evento electoral se genera una transacción *blockchain* desde la cuenta NEM de la comisión electoral hacia la cuenta NEM del evento electoral para registrar su información.

Información del mensaje de la transacción:

Atributo	Descripción
code	Código que identifica la creación de un evento electoral
name	Nombre del evento electoral
startDate	Fecha referencia de comienzo del evento electoral
endDate	Fecha referencia de fin del evento electoral

2.2. Crear elecciones

Una vez creado el evento electoral se espera a la fecha correspondiente para que el usuario administrador ingrese los datos que describen la elección. (Ilustración 17).



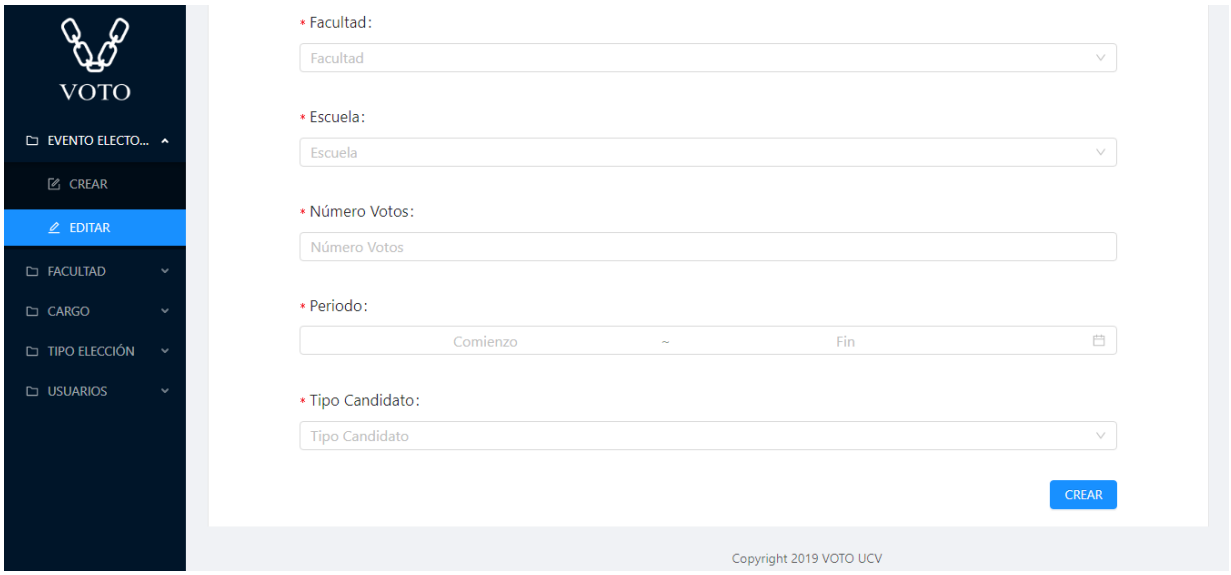


Ilustración 17 Crear Elección

Luego, desde la cuenta NEM de la comisión electoral hacia la cuenta NEM del evento electoral, se genera una transacción con la información de la elección captada.

Información del mensaje de la transacción con los datos de la elección:

Atributo	**Descripción**
code	Código que identifica la creación de una elección
Id	código que identifica una elección para que no se pueda crear más de una elección igual, creado a partir de la llave publica del evento electoral, tipo de elección, identificador de facultad, identificador de escuela.
name	Nombre de la elección.
type	Tipo de elección, por ejemplo, rector, centros de estudiantes, etcétera.
levelElection	A qué nivel de entidad se realiza la elección, por ejemplo, facultad.
typeCandidate	Tipo de candidato que se escogerá en la elección, como lo son, uninominal o lista.

typeElector	Tipo de elector que participara en la elección, como lo son, estudiante, profesor, egresado o consolidado.
facultyId	Identificador de facultad donde se realizará la elección.
schoolId	Identificador de escuela donde se realizará la elección.
allowedVotes	Número de votos para emitir por elector.
period	Periodo ejercicio de funciones.

2.3. Crear registro electoral

Una vez creada todas las elecciones se procede a crear el registro electoral en la fecha estipulada. Se seleccionan las personas que están habilitados a participar en las elecciones, y se procede a almacenarlos en la blockchain(Ilustración 18).

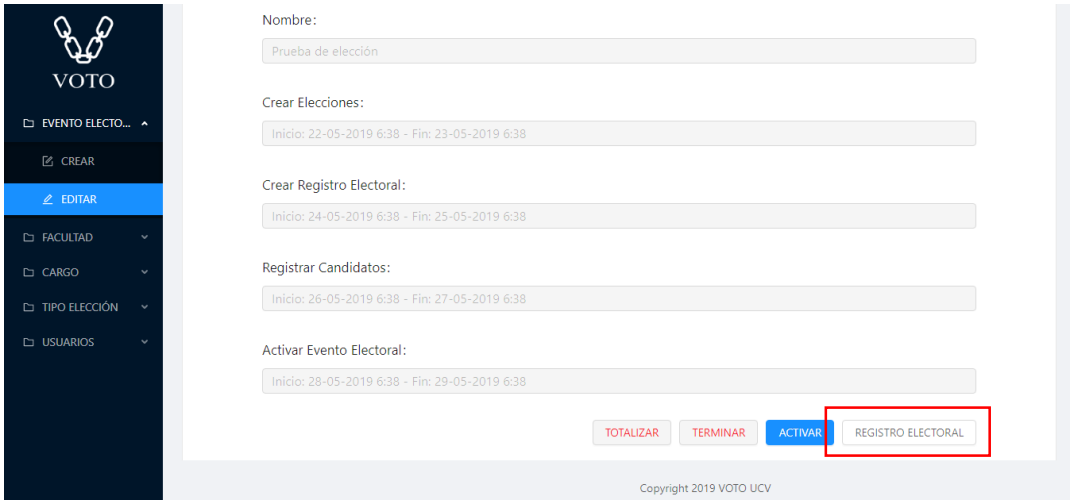


Ilustración 18 Crear Registro Electoral

Con la información de cada uno de los electores se creará una transacción, para posteriormente generar una transacción agregada y emitir todas las transacciones de forma atómica desde la cuenta NEM de la comisión electoral a la cuenta de registros electoral asociada al evento electoral. Una vez se complete la transacción de los electores, se crea una transacción desde la cuenta NEM de la comisión electoral hacia la cuenta NEM del evento electoral, la cual

indica que el registro electoral ha sido creado. Por último, son creado los registros en la base de datos relacional que servirán como método de autenticación.

Información del mensaje de la transacción de creación del registro electoral:

Atributo	Descripción
code	Código que identifica la creación del registro electoral.

Información del mensaje de la transacción del electoral:

Atributo	Descripción
code	Código que identifica la inserción de un elector en la blockchain.
identityDocument	Cedula de identidad del elector
facultyId	Identificador de la facultad a la que pertenece
schoolId	Identificador de la escuela a la que pertenece
type	Tipo de elector (estudiante, profesor, egresado)
electionsIds	Identificadores de las elecciones que puede participar el elector

2.4. Registro de candidatos

En el tiempo establecido, luego de crear el registro electoral, se procede a registrar los candidatos de cada una de las elecciones, estos deben ser parte del registro electoral. Con la información ingresada, se genera una transacción por cada uno de los candidatos y en conjunto con otra que contiene el id de la elección es creada una transacción agregada y almacenada en la cuenta del evento electoral.

Información del mensaje de la transacción con los datos asociado a un candidato:

Atributo	Descripción
code	Código que identifica la creación de un candidato.
name	Nombre del candidato.
identityDocument	Documente de identificación del candidato.
position	Cargo por el cual se está postulando.
list (opcional)	Nombre de la lista a la que pertenece el candidato en caso de ser una elección en donde el tipo de candidato sea lista

Información del mensaje de la transacción con el id de la elección:

Atributo	Descripción
code	Código que identifica el registro de candidatos de una elección
electionId	Identificador de la elección a la que pertenece los candidatos

2.5. Activar evento electoral

Cuando se quiera dar inicio al evento electoral se verificara la fecha pautada para hacerlo creando el mosaico de votación (Ilustración 19).

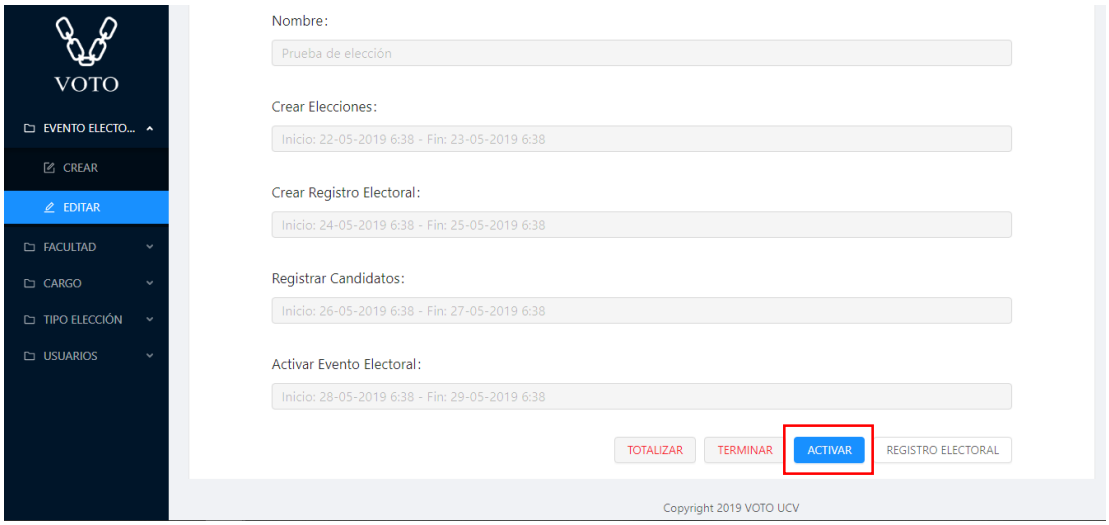


Ilustración 19 Activar Evento Electoral

Con las siguientes transacciones se genera una agregada. En primer lugar, la transacción de definición del mosaico de voto en donde se especifica que es mutable y transferible, en segundo lugar, una transacción para incrementar en uno la cantidad de mosaico, y finalmente se genera una transacción que contiene el identificador del mosaico. En este caso las primeras dos transacciones son de configuración en la *blockchain* y la ultima es enviada desde la cuenta NEM de la comisión electoral a la cuenta NEM del evento electoral.

Información del mensaje de la transacción:

Atributo	**Descripción**
code	Código que identifica la creación del mosaico de votación.
mosaicIdHex	Identificador del mosaico en la blockchain

Después de haber creado el mosaico de votación se obtiene el registro electoral. Por cada elector que conforma el registro se genera un código de autenticación que en conjunto con el documento de identidad y llave publica del evento electoral es creado un JSON Web Encryption

denominado “token de autenticación”, el cual se le envía por correo y finalmente se obtendrá el hash del código de autenticación y se guardará en la base de datos.

2.6. Autenticación Votante

El votante accede a la aplicación de voto electrónico a través del link que contiene el token de autenticación enviado a su correo (Ilustración 20).

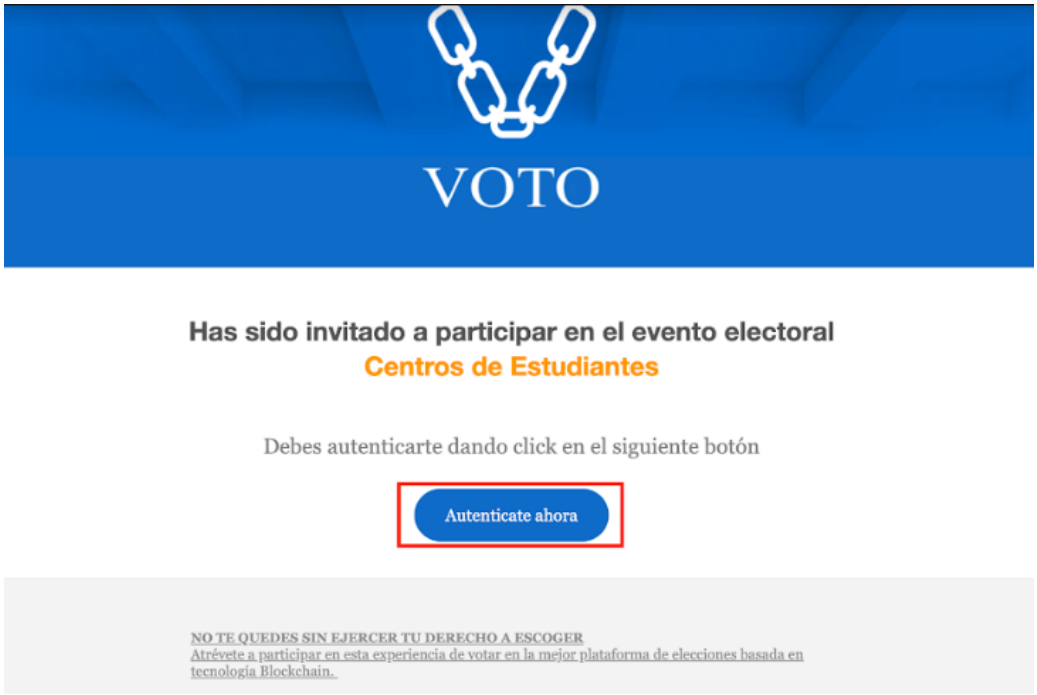


Ilustración 20 Correo de Autenticación

Este token es descifrado y se obtiene el documento de identidad, la llave publica del evento electoral y el código de autenticación, seguidamente se ubica el elector en el registro electoral de la *blockchain* y de la base de datos, se obtiene el código de autenticación que será validado con el recibido previamente. Luego de comprobar la autenticidad del elector se genera un código de acceso que en conjunto con su documento de identidad y la llave publica del evento electoral se forma un JSON Web Encryption denominado “token de acceso”, el cual es enviado al correo del elector y por último se obtendrá el hash del código de acceso y se guardará en la base de datos (Ilustración 23).

2.7. Acceso Votante

El votante accede a la aplicación de voto electrónico a través del link que contiene el token de acceso enviado a su correo (Ilustración 21).

Ilustración 21 Correo de Acceso

Este token es descifrado y se obtiene el documento de identidad, la llave publica del evento electoral y el código de acceso, seguidamente se ubica el elector en el registro electoral de la *blockchain* y la base de datos, se obtiene el código de acceso que será validado con el recibido previamente. Luego de comprobar que el elector está autorizado se le indica que debe ingresar una contraseña que posteriormente será utilizada para finalizar su proceso de votación. Después de almacenar la contraseña se buscan todas las elecciones en la *blockchain* donde puede participar el elector (Ilustración 22) (Ilustración 23).

Ilustración 22 Creación de Contraseña

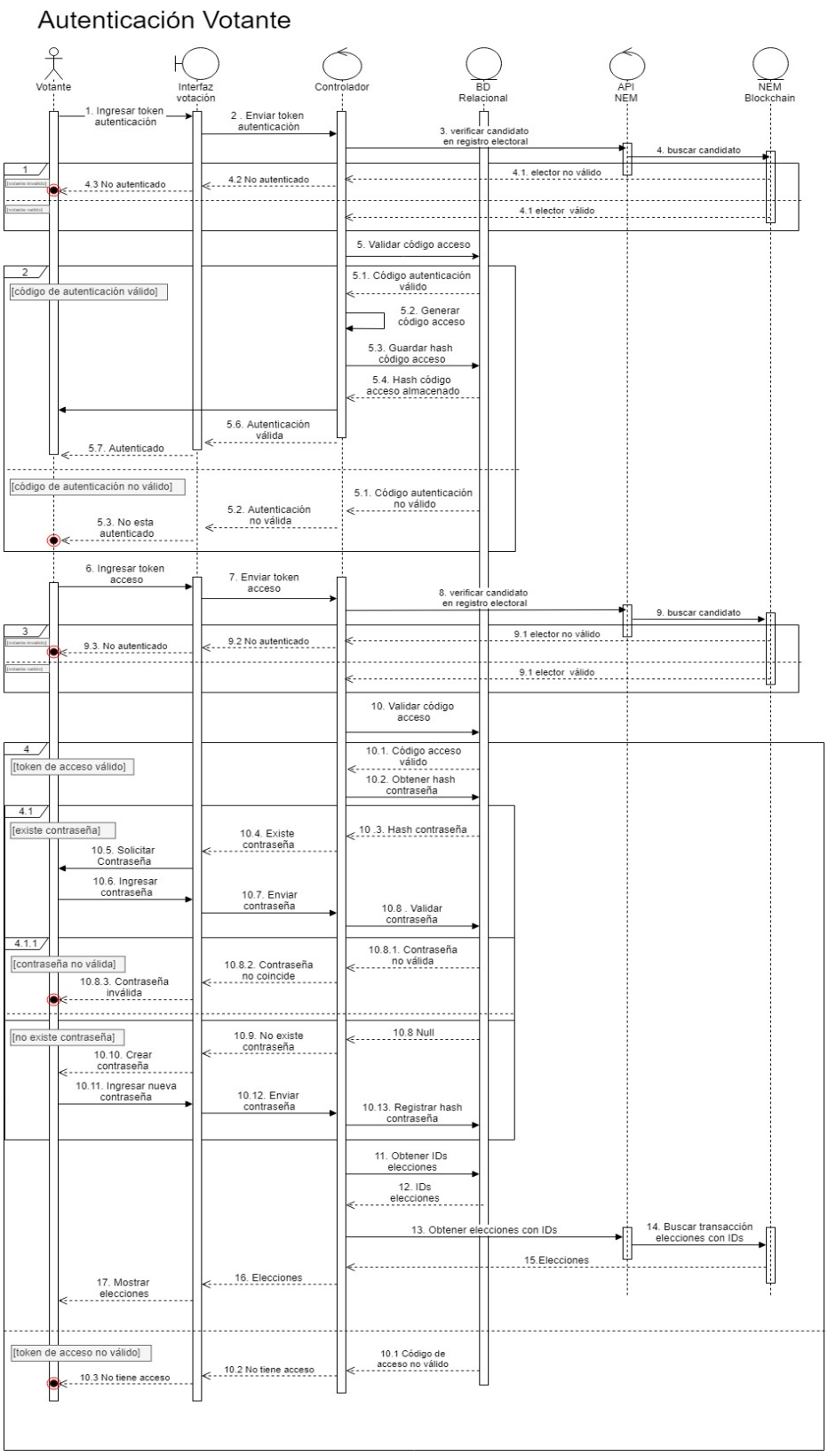


Ilustración 23 Diagrama de Secuencia Autenticación de Votante

2.8. Voto

El elector selecciona todos los candidatos de su preferencia en la papeleta de votación, y al finalizar ingresa su contraseña para poder emitir el voto (Ilustración 24).

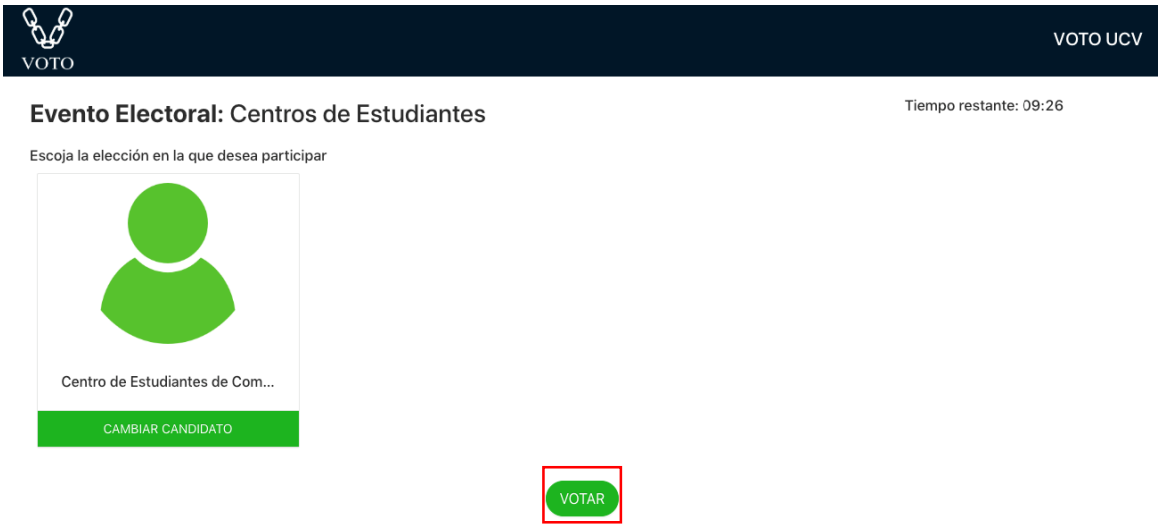


Ilustración 24 Confirmación de Voto

El token de acceso se descifra y se forma una dirección *blockchain* para el votante. Se genera una transacción con la cuenta NEM de la comisión electoral para incrementar en 1 la cantidad de mosaicos de voto de acuerdo a la cantidad de elecciones donde participó el votante. Una vez confirmada la transacción se crea otra que será firmada por la cuenta NEM de la comisión electoral para enviar los mosaicos de voto y comisión en criptomoneda XEM (fee) hacia la cuenta NEM del votante.

Por último, por cada candidato elegido de cada elección se creará una transacción que contendrá el mosaico de votación, además de manera encriptada con la llave privada de la cuenta de la comisión electoral, la cédula del candidato y el id de la elección donde participa, dichas transacciones son enviadas desde la cuenta del votante hacia la cuenta de votos encriptados de la eleccion. También es creada una transacción hacia la cuenta NEM del votante para indicar que ya votó.

Información del mensaje encriptado de la transacción de candidato:

Atributo	Descripción
identityDocument	Cedula de identidad del candidato.
electionId	Identificador de la elección

Información del mensaje de la transacción enviada a la cuenta NEM del votante:

Atributo	Descripción
code	Código que indica que es una transacción de voto válido

Cada transacción de candidato hacia la cuenta NEM de votos encriptados de eleccion y la enviada hacia la cuenta NEM del votante son anunciadas en una transacción agregada para emitir todos los votos al mismo tiempo (Ilustración 25).

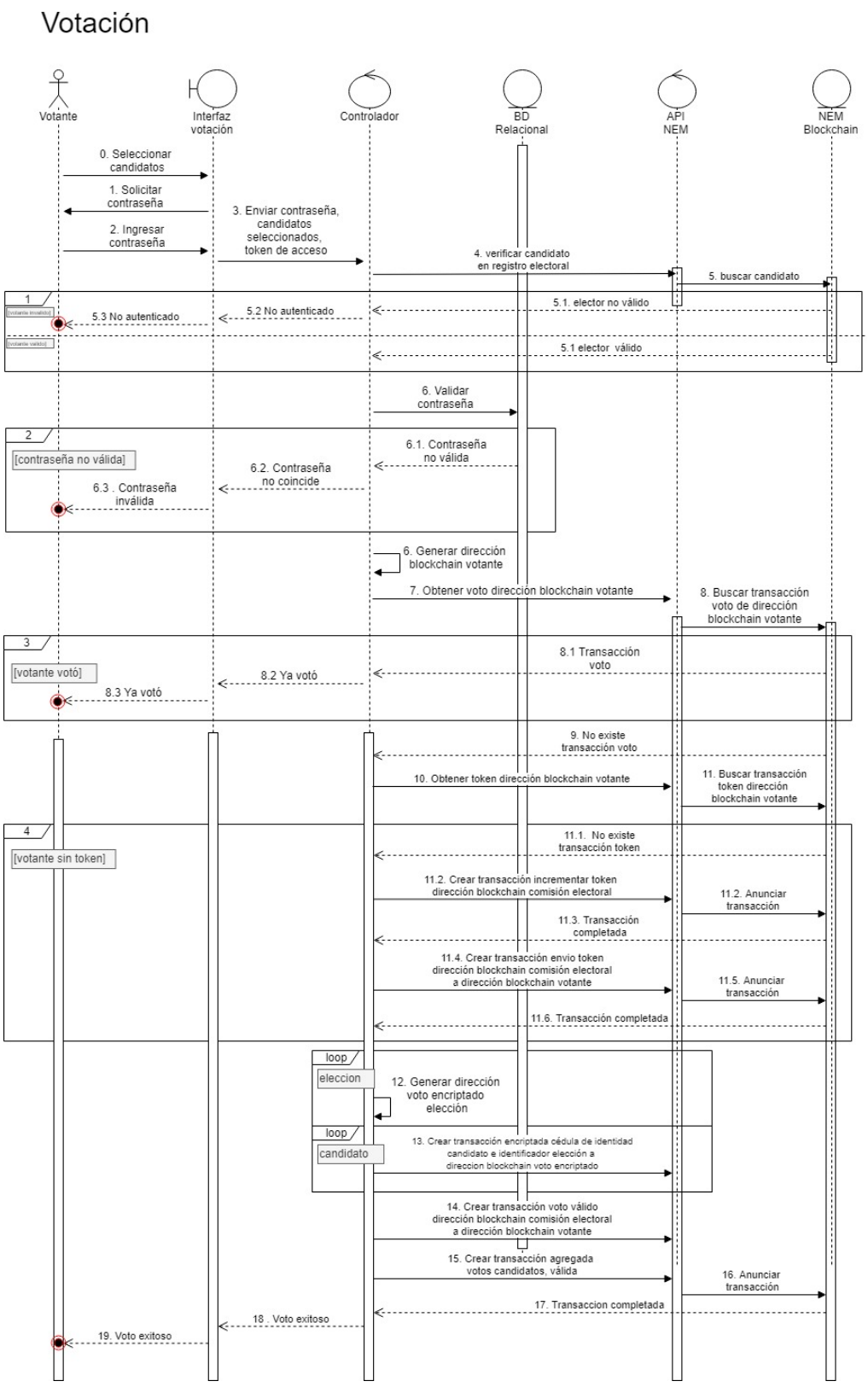


Ilustración 25 Diagrama de Secuencia de Voto

2.9. Finalizar evento electoral

Cuando se requiera finalizar el evento electoral el administrador se dirigirá al sistema administrativo para hacerlo (Ilustración 26).

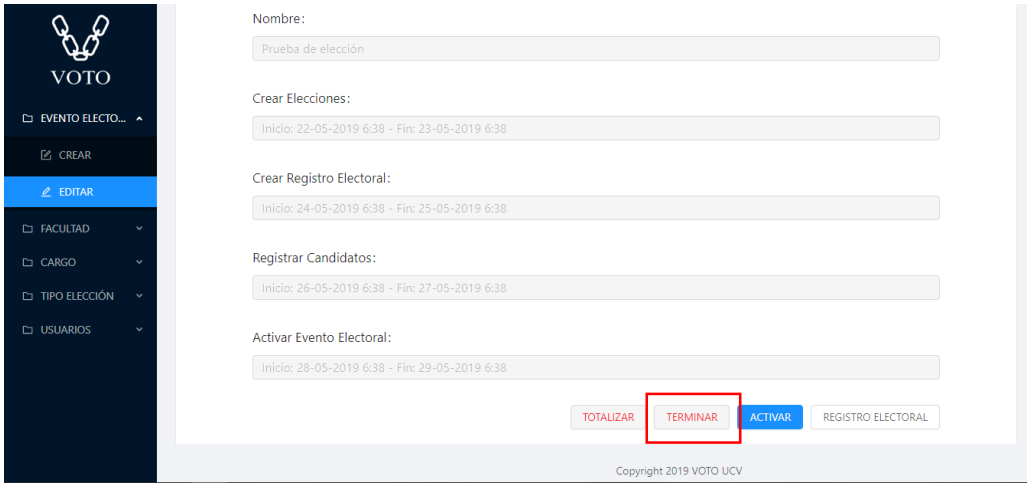


Ilustración 26 Finalizar Evento Electoral

Se creará una transacción desde la cuenta NEM de la comisión electoral hacia la cuenta NEM del evento electoral en donde se especifica en un mensaje que el evento ha finalizado.

Información del mensaje de la transacción:

Atributo	Descripción
code	Código que identifica la finalización del evento electoral

2.10. Totalizar evento electoral

Por cada una de las elecciones se procede a buscar las transacciones en su respectiva cuenta de votos encriptados, para ser desencriptados, totalizados y posteriormente enviar el total de votos obtenidos por candidatos a su respectiva cuenta NEM. (Ilustración 27).

Ilustración 27 Resultados Elección

3. Ventajas Sistema Voto UCV

Al crear un evento electoral con el sistema de Voto UCV se crea un valor agregado para los comicios.

- **Transparencia en los resultados**: Normalmente los votos son registrados, contados y verificados por un organismo centralizado, lo que en ciertos casos puede generar desconfianza. El *blockchain* hace un conteo automático sin intermediarios. Al final, es posible incluso que los votantes accedan a una copia de los resultados finales.
- **Reducción de costos operativos**: Automatizando los procesos permite reducir el uso de personal humano para las tareas de registro, control y distribución de datos. Lo que permitiría ahorrar gastos al reducir la cantidad de personas que deban realizar dichas tareas, además de las distintas herramientas que deben usar.
- **Mantiene la privacidad de identidad:** Los intercambios de datos funcionan a través de un alias, así que, en un proceso de votación, los usuarios obtienen un código público, que lo identifica en la red.
- **Seguridad:** Las operaciones criptográficas que se realiza en la *blockchain* para mantener la cadena integra, garantiza que la información almacenada no serán modificada. Además, de garantizar la seguridad física de los electores al poder ejercer el voto desde cualquier lugar.
- **Facilita el conteo**: En los procesos de votación convencionales el escrutinio suele tardar varias horas e incluso días. Con el sistema automatizado esto no es problema. Al final del día todo queda procesado y almacenado de forma segura, confiable y verificable.

- **Probable aumento de la participación electoral**: Si el sistema hace posible la votación digital desde su teléfono inteligente o computadora, votar es tan fácil como iniciar sesión y emitir su voto en pocos minutos. Esto probablemente aumentaría drásticamente la participación electoral, lo que llevaría a una democracia más directa.
- **Minimizar los errores humanos**: El proceso de recuento requiere de varias personas para hacer evitar cometerlos. Con blockchain el proceso sería mucho más eficiente.

4. Pruebas del Sistema

Para las pruebas del sistema totalmente en línea se utilizó:

- Hosting de firebase para alojar los sistemas visuales realizados en React.js, como lo son el admistrador y la papeleta electrónica.
- Servidor Nodejs de Google Cloud Platform para alojar el API del sistema.
- Base de datos MySQL en Google Cloud Platform
- Testnet NEM versión catapult para la blockchain

Se creó un evento electoral para escoger la representación estudiantil de la escuela de computación de la facultad de ciencias de la Universidad Central de Venezuela. Se crearon dos elecciones, una de tipo lista para seleccionar los candidatos que representarán el centro de estudiantes y otra de tipo uninominal para elegir los estudiantes encargados de ser voceros en el consejo de escuela.

Se contó con un registro electoral de 17 personas, de las cuales participaron 12 habiendo ejercido su derecho al voto de manera electrónica mediante el sistema desarrollado, en donde 5 personas reportaron tener dificultad motivada a la inestable conexión a internet por cual tuvieron que reiniciar el proceso de votación con las credenciales creadas previamente demostrando la flexibilidad del sistema.

A continuación, se mostrarán algunas transacciones almacenadas en la cuenta NEM de votos encriptados de la elección “Consejero de escuela de computación”.

En la Ilustración 28 Transacciones de votos encriptadosse puede apreciar que existen votos registrados, sin embargo, como no han sido totalizados, no se conocen los resultados, como en Ilustración 29 Vista de resultados antes de totalización.

{"meta":{"height":
362085,0],"hash":"03A303EC6851E1F483768EFE55C7EB50914EF9AD5A24451DF90A1A2149655A56","merkleComponentHash":"38244747F550FFB84832398EF35BE6260AEA0D3E0A5FD9D31C4E339E21801E
4","index":0,"id":"5CE539408F1EED00017E903D"},"transaction":
"signature":"1F504132499E8F5EF2776AABE7F0836BD1906AA9F47E8EDFF77FA8164FACDF6D7911700CCA7F3EEE2198BB5EBBDE6EACD9CE7612AF2024A524CAE0D4D55CF00A","signer":"ABDB09266A1A4BCD
CFC2323781D3899B4D523F27E2D45E20C9658D1001B6996","version":36866,"type":16705,"maxFee":[0,0],"deadline":[280289195,23],"cosignatures":
{"signer":"142A40F3955792E4810C087B3EEB497BE3E27F755C29C140D964549478EBCBEC","signature":"9C49A9AF1F96B62884BAE20097A419F9646C51E1714DBB1D8195B36547C585E1BEBA0568B9E3424
4ACBC947234EFD7BA972F09C8410BD7488F8612990332C0B"}],"transactions":[{"meta":{"height":
362085,0],"aggregateHash":"03A303EC6851E1F483768EFE55C7EB50914EF9AD5A24451DF90A1A2149655A56","aggregateId":"5CE539408F1EED00017E903D","index":0,"id":"5CE539408F1EED00017
903E"},"transaction":
"signer":"ABDB09266A1A4BCD5CFC2323781D3899B4D523F27B2D45E20C9658D1001B6996","version":36867,"type":16724,"recipient":"90E0E241DDA21F567FA1838B75EF6D45258B37AB95F5791743"
"message":
"type":1,"payload":"393330433534334130304146373730344646354646364130313738324133463344443134413737323836464538314534303245454341353235394236343637453332313846313132464444
413244374534454133323946453144433338344241354634363932334134423343464232423035373539314236364634364533046354443443237413443453337373234465358414437443130374134454138383
353738354434314438363239364236443934343738454464433363230464537303746333937413731343132394233414544463334383435363830333934463931384146303635444234303635434233423146413
363636423944363638324139453245353646303143443941363233434446324236443835363245331135424543324536374145363239303732304533364135441323837324146413534315"},"mosaics":[]}},
"meta":{"height":
362085,0],"aggregateHash":"03A303EC6851E1F483768EFE55C7EB50914EF9AD5A24451DF90A1A2149655A56","aggregateId":"5CE539408F1EED00017E903D","index":1,"id":"5CE539408F1EED00017
903F"},"transaction":
"signer":"ABDB09266A1A4BCD5CFC2323781D3899B4D523F27B2D45E20C9658D1001B6996","version":36867,"type":16724,"recipient":"90780C5F88B89A4B6D501C5FE43BCCDDAE274DBA110F2CFAFA"
"message":
"type":1,"payload":"393932353139353838463046394331423833333734334639433846303632373437303438414645463446353845463535363543324446384137353838363036383035373439363934433
324245453942423835353744304631393544323345383341324339443641374538384336424530333634413345324137383037323432434131444546443344303344443646434644464433837444643314344363374
354343334130323245414146373533313041464131353930354644423638303446444242466454146424132314434453044363643363734304243304441333346323132354532443245424130303634304238454133
384143304143414238324241314633333832383339374546433631443530364446463045363939363745355353413935423232393646353839454541463431384345334135324141414436"},"mosaics":[]}},
"meta":{"height":
362085,0],"aggregateHash":"03A303EC6851E1F483768EFE55C7EB50914EF9AD5A24451DF90A1A2149655A56","aggregateId":"5CE539408F1EED00017E903D","index":2,"id":"5CE539408F1EED00017
9040"},"transaction":
"signer":"ABDB09266A1A4BCD5CFC2323781D3899B4D523F27B2D45E20C9658D1001B6996","version":36867,"type":16724,"recipient":"90780C5F88B89A4B6D501C5FE43BCCDDAE274DBA110F2CFAFA"
"message":
"type":1,"payload":"354646333441383643464536314239433043363745314446384638423945323446353042313631324145313345444135463934453043364537343743383341303432463130373333314333
313544453337343335434336423432423836373233464243323341374437364346334646333043363041313330463241373033333039384541413235334545363638393543384137354337424113139304645334533
334646383242443945374444323638313646304243354230364241443935433139463046313442303941394436384233443430353742344434334444363043304531344431413144443636336413238383433363
414134363831414242453538313245304536354234423731464138453744354537433337454230363742344344463043443636323459393633324233304442353543396323413345137"},"mosaics":[]}},

Ilustración 28 Transacciones de votos encriptados

Ilustración 29 Vista de resultados antes de totalización

Una vez finalizado el evento electoral se procede a realizar la totalización de votos, en la Ilustración 30 podemos observar la cuenta NEM de uno de los candidatos, la cual refleja el total de votos obtenidos en la presente elección, estos de muestran por defecto en un formato

hexadecimal (no encriptados). Cualquier usuario, una vez finalizado el evento electoral podrá acceder a los resultados obtenidos como en la Ilustración 31.


```
[{"meta":{"height":
[362262,0],"hash":"67C463D41100F6E4F82F8599C45D74866ED273A64DAD070D109B288CDFDC09FE","merkleComponentHash":"67C463D41100F6E4F82F8599C45D74866ED273A64DAD070D109B288CDFDC09FE
","index":0,"id":"5CE543EF8F1EED00017E91CC"},"transaction":
{"signature":"B7529201AED80C4E95148ADDFC7E204676D9ABEB5E80C4B60DC536970CB125DF91546AF5602D2985A1129C653C2313AC5F67B08FE9CD3C0B5F80F275F53FDF08","signer":"142A40F3955792E481
0C087B3EEB497BE3E27F755C29C140D964549478EBCBEC","version":36866,"type":16705,"maxFee":[0,0],"deadline":[283147238,23],"cosignatures":[],"transactions":[{"meta":{"height":
[362262,0],"aggregateHash":"67C463D41100F6E4F82F8599C45D74866ED273A64DAD070D109B288CDFDC09FE","aggregateId":"5CE543EF8F1EED00017E91CC","index":0,"id":"5CE543EF8F1EED00017E9
1CD"},"transaction":
{"signer":"142A40F3955792E4810C087B3EEB497BE3E27F755C29C140D964549478EBCBEC","version":36867,"type":16724,"recipient":"90192528B75ECCEDC14EF5A2061BE75625573A5D40860BD166","
message":{"type":0,"payload":"7B22636F6465223A22303039222C2264617461223A7B22766F746573223A367D7D"},"mosaics":[]}},{"meta":{"height":
[362262,0],"aggregateHash":"67C463D41100F6E4F82F8599C45D74866ED273A64DAD070D109B288CDFDC09FE","aggregateId":"5CE543EF8F1EED00017E91CC","index":1,"id":"5CE543EF8F1EED00017E9
1CE"},"transaction":
{"signer":"142A40F3955792E4810C087B3EEB497BE3E27F755C29C140D964549478EBCBEC","version":36867,"type":16724,"recipient":"901796FE85F80978F7B94EADA432A2C1C8819425C91253BB40","
message":{"type":0,"payload":"7B22636F6465223A22303039222C2264617461223A7B22766F746573223A367D7D"},"mosaics":[]}},{"meta":{"height":
[362262,0],"aggregateHash":"67C463D41100F6E4F82F8599C45D74866ED273A64DAD070D109B288CDFDC09FE","aggregateId":"5CE543EF8F1EED00017E91CC","index":2,"id":"5CE543EF8F1EED00017E9
1CF"},"transaction":
{"signer":"142A40F3955792E4810C087B3EEB497BE3E27F755C29C140D964549478EBCBEC","version":36867,"type":16724,"recipient":"90AEC469D9A6BDCFFA4410FD5CFEA05EFFF555A03AADF957443","
message":{"type":0,"payload":"7B22636F6465223A22303039227D"},"mosaics":[]}}]}}]
```


Ilustración 30 Transacción final de votos

Ilustración 31 Vista de Resultados después de totalización

Conclusión

Por décadas el subproceso de votación en las elecciones de la Universidad Central de Venezuela (UCV) ha sido realizado de forma manual, método implementado sin problemas hasta hace algunos años, pero su realización se ha visto dificultada por la situación económica y social que atraviesa la universidad, impidiendo costear los gastos necesarios para suministros y personal logístico.

Es necesario adoptar nuevas tecnologías que permitan agilizar el sufragio, reducir costos e incrementar la seguridad del proceso, para ello se sugiere la adaptación a un sistema de votación electrónico, pero estos sistemas han sido poco aceptados por la sociedad y entes gubernamentales porque presuntamente infringen los principios del sufragio universal.

En los últimos años la tecnología *blockchain* ha demostrado ser una solución para procesos en donde los datos se encuentran distribuidos en distintos nodos ofreciendo seguridad, transparencia y anonimato para el ecosistema. Esta tecnología es el componente idóneo para un sistema de votación electrónico, ya que con sus características resuelve los problemas que estos sistemas generan.

Por consiguiente, para resolver el problema que presenta el subproceso de votación en la UCV se propuso la utilización de un sistema de votación electrónico utilizando como componente adicional la tecnología *blockchain*.

Referencias

[1] J. Franco, «El Derecho Humano al Voto,» Comisión Nacional de los Derechos Humanos, Ciudad de México, 2016.

[2] M. Solvak y K. Vassil, «E-voting in Estonia:Technological Diffusion and Other Developments Over Ten Years (2005 - 2015),» Johan Skytte Institute of Political Studies, Estonia, 2016.

[3] Y. Wu, «An E-voting System based on Blockchain and Ring Signature,» School of Computer Science, University of Birmingham, Birmingham, 2007.

[4] P. Neumann, «Security Criteria for Electronic Voting,» SRI International, Menlo Park, 1993.

[5] O. Spycher, R. Koenig, R. Haenni y M. Schläpfer, «A New Approach Towards Coercion-Resistant Remote E-Voting in Linear Time,» Springer-Verlag, Heidelberg, 2012.

[6] M. Palugulla, «Electronic-voting approach with an open cloud computing architecture,» *International Journal Of Engineering And Computer Science,* vol. III, nº 11, pp. 9012-9015, 2014.

[7] E. Chaparro, «El Sistema de Voto Electrónico de la Ciudad de Buenos Aires: Una "Solución" en Busca de Problemas,» Enrique A. Chaparro y Fundación Vía Libre, Buenos Aires, 2015.

[8] R. Chica, «Weaknesses in Centralized adn Decentralized Internet Voting Protocols,» Universitá di Pisa, Pisa, 2018.

[9] D. Rubin, «Brave New Ballot: The Battle to Safeguard Democracy in the Age of Electronic Voting,» Morgan Road Book, New York, 2006.

[10] Universidad Central de Venezuela, «Manual de Organización de la Universidad Central de Venezuela,» División de Organización Y Sistemas, Caracas, 2016.

[11] Universidad Central de Venezuela, «Reglamento de Elecciones Universitarias,» Caracas, 2007.