UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS

TESIS DE GRADO

PROTOCOLOS BLOCKCHAIN APLICADOS A UN SISTEMA DE VOTACIÓN ELECTRÓNICA

CASO: CARRERA INGENIERÍA DE SISTEMAS – UPEA

Para Optar al Título de Licenciatura en Ingeniería de Sistemas
Mención: INFORMÁTICA Y COMUNICACIONES

Postulante: Hector Churata Sonco

Tutor Metodológico: M.Sc. Enrique Flores Baltazar

Tutor Especialista: Ing. Ramiro Kantuta Limachi

Tutor Revisor: Ing. Elías Carlos Hidalgo Mamani

EL ALTO – BOLIVIA

2020

DEDICATORIA

A Dios por darme la oportunidad de llegar a esta instancia de mi vida.

A mi familia por el apoyo incondicional y por brindarme su amor.

A mis tutores por la ayuda y colaboración en el desarrollo de la presente tesis.

AGRADECIMIENTOS

Agradecer a Dios por todas las bendiciones y en especial por haberme dado la oportunidad de cumplir esta meta trazada, por darme salud y fuerzas en momentos complicados que me tocó vivir.

A mi familia por la paciencia, amor y apoyo en todos estos años, a pesar de los errores cometidos, siempre estuvieron a mi lado.

A mis tutores, Tutor Especialista Ing. Ramiro Kantuta Limachi, Tutor Metodológico Ing. Enrique Flores Baltazar, Tutor Revisor Ing. Elías Carlos Hidalgo Mamani, que fueron de gran ayuda para concluir con la tesis de grado que me propuse, por darme orientación, compartirme su conocimiento y agradecido por la paciencia que tuvieron en el desarrollo del presente tema.

A los docentes de la Carrera de Ingeniería de Sistemas por trasmitirme sus conocimientos para mi formación académica.

A la Universidad Pública de El Alto por la oportunidad que me dio de ser parte y haberme brindado una formación académica.

Y finalmente a todos mis amigos y compañeros que fueron apoyo en el tiempo de formación.

RESUMEN

Actualmente, la tecnología se ha implementado en diferentes campos de trabajo, construyendo un mundo diferente, uno de los muchos campos donde se aplica es en los procesos electorales, donde se desarrollaron sistemas de votación electrónica, con el fin de agilizar los procesos electorales tradicionales, sin embargo, todavía existen discusiones sobre este tema. Existen un gran grupo de personas quienes piensan que no es factible llevar a cabo procesos electorales a través de un sistema de votación electrónica, ya que son más vulnerables a posibles ataques informáticos y manipulación de datos. Esto fue lo que motivo a realizar esta investigación. Hoy en día se ha llegado a escuchar con mayor fuerza sobre la tecnología blockchain y sus grandes beneficios en la seguridad de la información, causando impacto en el área financiero, donde las monedas digitales funcionan sobre el sistema de blockchain. Para el desarrollo de este tema de investigación se propuso tomar como caso de estudio la Carrera de Ingeniería de Sistemas perteneciente a la Universidad Pública de EL Alto, donde se observó que llevan un proceso electoral tradicional que consiste en emitir votos por medio de papeletas físicas. Para el presente trabajo de investigación se construyó un sistema de votación electrónica, utilizando metodologías, métodos, herramientas de desarrollo para luego aplicar algoritmos blockchain, con el fin de comprobar la seguridad que proporciona a los datos y probar la hipótesis planteada.

Palabras claves: Blockchain, seguridad, algoritmos, voto electrónico.

ABSTRACT

Currently, technology has been implemented in different fields of work, building a different world, one of the many fields where it is applied is in electoral processes, where electronic voting systems were developed, in order to streamline traditional electoral processes, however, there are still discussions on this topic. There is a large group of people who think that it is not feasible to carry out electoral processes through an electronic voting system, since they are more vulnerable to possible computer attacks and data manipulation. This was the reason for this investigation. Today, blockchain technology and its great benefits in information security have been heard with greater force, causing an impact in the financial area, where digital currencies work on the blockchain system. For the development of this research topic, it was proposed to take as a case study the Systems Engineering Career belonging to the Public University of EL Alto, where it was observed that they carry out a traditional electoral process that consists of casting votes by means of physical ballots. For the present research work, an electronic voting system was built, using methodologies, methods, and development tools to then apply blockchain algorithms in order to check the security it provides to the data and test the hypothesis proposed.

Key words: Blockchain, security, algorithms, electronic voting.

INDICE GENERAL

Págs.

INDICE DE FIGURAS

Págs.

INDICE DE TABLAS

Págs.

CAPÍTULO I

MARCO PRELIMINAR

MARCO PRELIMINAR

1.1 Introducción

Hoy en día se buscan métodos con el objetivo de mejorar la seguridad en los sistemas de información, esto nos lleva a conocer la tecnología blockchain (cadena de bloques), que como ventaja principal tiene es brindar la seguridad e integridad de datos. Esta tecnología se aplicó por primera vez en el área financiero (monedas digitales) y en el día de hoy se comienza a aplicar a nuevos campos como por ejemplo en los procesos electorales.

A nivel mundial la tecnología blockchain aplicado al voto electrónico se encuentra en investigaciones y pruebas, en el país de Suiza, ciudad de Zug llevará acabo una votación de prueba basado en blockchain, otro caso reciente fue: “Las elecciones primarias de West Virginia el 8 de mayo, completando el primer voto gubernamental respaldado por blockchain en la historia de EE.UU., reportó ETHNews el 9 de mayo. Mientras la mayoría de los votantes emiten boletas regulares, especiales los votantes en ciertos distritos votaron en una plataforma móvil basada en blockchain” (Alexandre, 2018).

En Bolivia recientemente Jóvenes cochabambinos, usando la tecnología blockchain (cadena de bloques), desarrollaron sistemas informáticos para realizar una votación electrónica y acceder a fichas médicas de manera segura, confidencial e íntegra de los datos. De esta manera, con la aplicación de esta tecnología estos procesos tienen menos riesgo de un ataque (Camacho, 2017).

También una de las últimas publicaciones fue que el Tribunal Supremo Electoral de Chuquisaca, realizó pruebas de un sistema de voto electrónico en establecimientos educativos para elección de sus gobiernos estudiantiles. Olga Mary Martínez presidenta del TDE Chuquisaca, destacó que este sistema es “ágil, bastante seguro y amigable para los estudiantes” y está elaborado con la aplicación del procedimiento electoral, que permite conformar el padrón electoral de la unidad educativa, el registro de los frentes, la búsqueda de votantes, la

emisión del voto y el cómputo simultáneo al proceso de votación en una pantalla táctil (Correo del sur, 2018).

Con el presente trabajo de investigación se pretende probar que la tecnología Blockchain garantiza la transparencia y seguridad de los votos de un sistema de votación electrónica. Esta tecnología es una alternativa para dar solución a posibles fraudes electorales, manipulación de datos a favor de algún candidato y muchos otros problemas que conlleva un proceso electoral.

1.2 Antecedentes

1.2.1 Antecedentes académicos

- **Internacional**

  - (Valeria M. Baldeon y Joel F. Zambrana; 2018) “Implementación de un prototipo de una red descentralizada Blockchain para el voto electrónico en la Universidad de Guayaquil”. El objetivo general del trabajo es: “Implementar un prototipo de una red descentralizada Blockchain para realizar el voto de manera electrónica obteniendo un mayor grado de confianza y seguridad al momento de realizar los procesos electorales en la Universidad de Guayaquil”, utiliza la metodología de desarrollo ADDIE. Universidad de Guayaquil, Guayaquil – Ecuador.

  - (Alex Calvopiña y Stephanie Garcia; 2016) “Diseño e implementación del voto electrónico mediante el uso de una aplicación web, para las elecciones de la Federación de Estudiantes de la Pontificia Universidad Católica del Ecuador”. El objetivo general del proyecto es: “Diseñar e implementar el voto electrónico mediante el uso de una aplicación web, para las elecciones de la Federación de Estudiantes de la Pontificia Universidad Católica del Ecuador”, la metodología que se utilizo es Extreme Programming (XP). Pontificia Universidad Católica del Ecuador, Quito – Ecuador.

- (Jesús E. Plasencia; 2018) “Sistema de votación electrónica basado en blockchain”. Como objetivos se ha planteado el desarrollo de un sistema de voto electrónico que cumpla las siguientes características: Multiautoridad, Auditoría abierta, Descentralizado en la red. Universidad de la Laguna, Santa Cruz de Tenerife – España.

➢ **Nacional**

- (Braian S. Villarpando; 2017) “Diseño de un modelo de sistema de voto Electrónico en proceso de elecciones”. El objetivo general de la tesis es: “Diseñar un modelo de voto electrónico con características de anonimato para un proceso eleccionario”. UMSA, La Paz – Bolivia.

1.3 Planteamiento del Problema

1.3.1 Problema principal

Los procesos electorales en la Carrera de Ingeniería de Sistemas no cuentan con un sistema de votación electrónica basada en la tecnología blockchain, por el cual existe la inseguridad de los votos emitidos, como consecuencia de aquello genera desconfianza, reclamos por parte de los votantes y por los propios frentes políticos.

1.3.2 Problemas secundarios

- Métodos de verificación de votantes deficiente y simple.
- Procesos demorosos para los reportes finales debido a la centralización de papeletas y su posterior conteo.
- Susceptibilidad en los procesos de conteo y post-votación.
- Falta de aplicación de nuevas tecnologías para el proceso de sufragio.
- Incertidumbre en los datos con respecto a la seguridad.

Una vez identificado los problemas, se plantea la siguiente interrogante:

¿Cómo el desarrollo de un sistema de votación electrónica basada en la tecnología blockchain contribuirá a mejorar la seguridad y transparencia de los procesos tradicionales de sufragio?

1.4 Objetivos

1.4.1 Objetivo general

Desarrollar un sistema de votación electrónica basada en la tecnología blockchain, para mejorar la seguridad y transparencia de los procesos electorales.

1.4.2 Objetivos específicos

- Analizar la tecnología blockchain y sus ventajas.
- Crear una herramienta tecnológica que permita agilizar el proceso electoral.
- Diseñar una interfaz fácil y amigable para la interacción con el usuario.
- Aplicar la tecnología blockchain a un sistema de votación electrónica.
- Mostrar el nivel de seguridad que ofrece la tecnología blockchain.

1.5 Hipótesis

La aplicación de protocolos blockchain a un sistema de votación electrónica contribuirá a mejorar la seguridad y transparencia de los procesos electorales.

1.5.1 Identificación de variables

Variable dependiente:

- Seguridad y transparencia.

Variable independiente:

- Protocolos blockchain.

Variable interviniente:

- Sistema de votación electrónica.

1.5.2 Operacionalización de variables

Variables		Dimensiones	Indicadores	Herramientas
Variable Independiente	Protocolos blockchain	Herramienta tecnológica que permita mejorar la seguridad.	Incorruptibilidad Transparente	Bloques Algoritmos
Variable Dependiente	Seguridad y transparencia	Nivel de seguridad alto.	TCP (Protocolo de Control de Transmisión)	Algoritmo de encriptación RSA
Variable Interviniente	Sistema de votación electrónica	Nuevas tecnologías. Administración electoral.	Emisión de voto Nuevas elecciones Escrutinio	Gestor de Base de datos. Framework. Lenguajes de programación. Servidor Web.

1.5.3 Conceptualización de variables

Protocolos blockchain

Un blockchain es un controlador digital compartido, que muestra las manipulaciones y que registra las transacciones en redes “par a par” públicas o privadas. El distribuidor, que está distribuido en todos los nodos que pertenecen a la red, registra constantemente en una cadena secuencial de bloques de cifrado hash enlazado, que es el historial de intercambios de activos que ha tenido lugar entre los pares de la red (IBM, 2018).

Seguridad y transparencia

Seguridad es un conjunto de sistemas, medios organizativos, medios humanos y acciones dispuestas para eliminar, reducir o controlar los riesgos y amenazas que puedan afectar a una persona a una entidad a una instalación o a un objeto. La seguridad proporciona las condiciones para afrontar el peligro, en síntesis, seguridad es la minimización del riesgo. (Escuela Penitenciaria Nacional, 2019)

La transparencia en las contiendas electorales tiene como propósito, dar a conocer a la sociedad en general las actividades que desarrollan los candidatos, los partidos políticos, los órganos electorales ya sean administrativos o jurisdiccionales y los propios ciudadanos. (López, 2020)

Sistema de votación electrónica

El sistema de voto electrónico, tiene como objetivo facilitar al elector el ejercicio del voto, eliminando las barreras iniciales que puedan tener algunos votantes ante las nuevas tecnologías. Se trata de una urna electrónica dotada de pantalla táctil con la que el votante visualiza y elige sus opciones de voto. (Gobierno de España, 2020)

1.6 Justificación

1.6.1 Justificación técnica

Los procesos electorales en la Carrera de Ingeniería de Sistemas se realizan de manera tradicional, es decir mediante papeletas y urnas físicas, una ventaja que tiene la carrera es que cuenta con equipos de cómputo para realizar una elección y la implementación de un sistema de votación electrónica basada en protocolos blockchain.

1.6.2 Justificación económica

El sistema de voto electrónico hace el uso de herramientas de desarrollo de software libre, llegando a ser un beneficio para la institución en la reducción de gastos que conlleva un proceso electoral.

En agosto de 2011, se aprobó la nueva Ley de Telecomunicaciones y Tecnologías de Información y Comunicación, que en su artículo 77 asume como una de las tareas del Estado la promoción y priorización del uso de Software Libre en todos sus niveles.

1.6.3 Justificación social

El presente trabajo de investigación se justifica socialmente porque proporciona la seguridad de los votos emitidos por los votantes, evitando que sean modificados, además que el uso de nuevas tecnologías para un proceso electoral contribuye en la optimización del tiempo que se demora en realizar las tareas necesarias para llevar a cabo una votación.

1.6.4 Justificación científica

La tecnología blockchain hace el uso de algoritmos matemáticos para el cifrado de datos, contribuye en la seguridad de la información, en síntesis, se considera que el uso de la tecnología blockchain contribuirá en el avance tecnológico y científico.

1.7 Metodología

1.7.1 Método científico

El método científico es un conjunto de pasos ordenados que se emplean para adquirir nuevos conocimientos. Para poder ser calificado como científico debe basarse en el empirismo, en la medición y, además, debe estar sujeto a la razón. (Gargantilla, 2020)

Estos son los cinco pasos del método científico:

- Observación: hace referencia a lo que queremos estudiar o comprender.
- Hipótesis: se formula una idea que pueda explicar lo observado.
- Experimentación: se llevan a cabo diferentes experimentos para comprobar o refutar una hipótesis.
- Teoría: permite explicar la hipótesis más probable.
- Conclusiones: se extraen de la teoría formulada.

1.7.2 Metodología de desarrollo UWE

La propuesta de Ingeniería Web basada en UML (UWE (Koch, 2000)) es una metodología detallada para el proceso de autoría de aplicaciones con una definición exhaustiva del proceso de diseño que debe ser utilizado. Este proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con las propuestas en el Proceso Unificado de Modelado. (Mínguez, 2010).

Los modelos que abarcan la metodología UWE son los siguientes:

- Modelo de requerimientos
- Modelo conceptual
- Modelo de navegación

- Modelo de presentación
- Modelo de procesos

1.7.3 Métrica de calidad

ISO 9126

La ISO 9126 es un estándar internacional para evaluar la calidad del software en base a un conjunto de características y sub-características de la calidad. Cada sub-característica consta de un conjunto de atributos que son medidos por una serie de métricas. (Moreno, Toledo, Lopez, & Cruz, 2020)

Las características de norma ISO 9126 son:

- **Funcionabilidad:** conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen lo indicado o implica necesidades.
- **Confiabilidad:** conjunto de atributos relacionados con la capacidad de mantener un nivel de presentación bajo condiciones establecidas durante un periodo de tiempo establecido.
- **Usabilidad:** Conjunto de atributos relacionados con el esfuerzo necesitado para el uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.
- **Eficiencia:** Conjunto de atributos relacionados con la relación entre el nivel de desempeño del SW y la cantidad de recursos necesitados bajo condiciones establecidas.
- **Mantenibilidad:** Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema SW.
- **Portabilidad:** Conjunto de atributos relacionados con la capacidad de un sistema SW para ser transferido desde una plataforma a otra.

1.7.4 Modelo de costos

COCOMO

Este modelo permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de costo y de escala. Los factores de costo describen aspectos relacionados con la naturaleza del producto, hardware utilizado, personal involucrado, y características propias del proyecto. El conjunto de factores de escala explica las economías y des economías de escala producidas a medida que un proyecto de software incrementa su tamaño. (Gomez, Lopez, Migani, Otazu, 2020)

1.8 Herramientas

1.8.1 Servidor web

Apache

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual según la normativa RFC 2616.

Apache es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web. (Wikipedia, 2020)

1.8.2 Gestor de base de datos

MariaDB

MariaDB es una bifurcación popular de MySQL creada por los desarrolladores originales de MySQL. Creció de preocupaciones relacionadas con la adquisición de MySQL por Oracle. Ofrece soporte tanto para datos pequeños tareas de procesamiento y necesidades empresariales. Su objetivo es ser un reemplazo

directo para MySQL requiere solo una simple desinstalación de MySQL y una instalación de MariaDB. MariaDB ofrece las mismas características de MySQL y mucho más. (Point, 2020)

1.8.3 Lenguajes de programación

PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga. (php, 2020)

JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. (Eguiluz, 2019)

1.8.4 Framework

CodeIgniter

Es un programa o aplicación web desarrollada en PHP para la creación de cualquier tipo de aplicación web bajo PHP. Es un producto de código libre, libre de uso para cualquier aplicación. Como cualquier otro framework, Codeigniter contiene una serie de librerías que sirven para el desarrollo de aplicaciones web y además propone una manera de desarrollarlas que debemos seguir para obtener provecho de la aplicación.

Marca una manera específica de codificar las páginas web y clasificar sus diferentes scripts, que sirve para que el código esté organizado y sea más fácil de crear y mantener. (Alvarez, 2017)

Flask

Flask es un framework minimalista de desarrollo de aplicaciones web. Es utilizado para crear una aplicación que permita monitorizar la red e interaccionar con los nodos y la blockchain de manera interactiva. Se ha preferido usar Flask sobre Django, una alternativa popular, ya que necesita menos configuración y que para el alcance de la aplicación web no era necesario un framework tan completo. (Plasencia, 2018)

1.9 Límites y Alcances

1.9.1 Límites

- El presente trabajo tiene como límite la aplicación de la tecnología blockchain en el proceso de emisión de votos.
- La presente tesis se limita como caso de estudio la carrera de Ingeniería de Sistemas – UPEA.
- EL sistema desarrollado no realiza la emisión de certificados de sufragio.

1.9.2 Alcances

- La Aplicación de protocolos blockchain a un sistema de votación electrónica.
- Demostrar el nivel de seguridad que proporciona los algoritmos de cifrado.
- Facilitar la visualización de votos que serán extraídos del sistema de votación electrónica y la blockchain.

1.10 Aporte

La presente tesis tiene como aporte principal la aplicación de la tecnología blockchain a un sistema votación electrónica, a través de esta tecnología se logrará mejorar los procesos electorales obteniendo mayor eficiencia, transparencia e inmutabilidad de los datos.

CAPÍTULO II

MARCO TEÓRICO CONCEPTUAL

MARCO TEÓRICO

2.1 Introducción

En el presente capítulo se desarrollan las bases teóricas para la construcción del tema de investigación planteado, se recolecto información de distintos libros y autores, de esta manera profundizar los conceptos y características importantes de herramientas, métodos, metodologías, principalmente para sustentar el tema de investigación y objetivos trazados en el anterior capitulo.

2.2 Sistema

Según Sommerville (2011) “Un sistema es una colección intencionada de componentes interrelacionados, de diferentes tipos, que trabajan en conjunto para lograr algún objetivo” (p.266).

Según kendall y Kedall define sistema como: “Una colección de subsistemas que están interrelacionados y son interdependientes; trabajan en conjunto para lograr metas y objetivos predeterminados. Todos los sistemas tienen entrada, procesos, salida y retroalimentación. Algunos ejemplos son un sistema de información computarizado y una organización” (p.563).

En ambas definiciones se puede notar similitud por lo tanto se puede definir la palabra sistema como un conjunto de componentes relacionados para lograr uno o más objetivos.

2.3 Sistema Web

Una aplicación web (web-based application) es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (HTTP) están estandarizados y no han de ser creados por el programador de aplicaciones. (Lujan, 2002, p.48)

Según Molina Rios & Zea Ordoñez, (2017) menciona:

> Por último, se define a una aplicación Web como un programa informático o sitio Web que ejecuta en el internet sin necesidad de una instalación en el ordenador, tan solo con el uso de un navegador esto debido a que se programa en lenguaje HTML y ofrece múltiples ventajas para los usuarios como: acceder a la información de manera ágil y sencilla, recolectar y guardar información, etc. (p.247)

2.4 Sistema de Voto Electrónico

El voto electrónico en sentido amplio, es todo mecanismo de elección en el que se utilicen los medios electrónicos, o cualquier tecnología, en las distintas etapas del proceso electoral, teniendo como presupuesto básico que el acto efectivo de votar se realice mediante cualquier instrumento electrónico de captación del sufragio.

En sentido estricto, el voto electrónico es el acto preciso en el cual el emitente del voto deposita o expresa su voluntad a través de medios electrónicos (urnas electrónicas) o cualquier otra tecnología de recepción del sufragio. (Téllez, 2010, p.16)

Aplicación de dispositivos y sistemas de tecnología de la información y telecomunicación al acto de sufragio. Total, o parcialmente, a todo proceso electoral, o a algunas de las distintas actividades del sufragio, el registro y verificación de la identidad del elector, Incluso emisión misma del voto en una urna electrónica (con o sin impresión inmediata de boleta de papel para control de ciudadano o de la autoridad); el recuento en la mesa o el global consolidado, la transmisión de resultados, u otras actividades. (Price, 2006, p.16)

Por lo que deducimos que un sistema de voto electrónico es un programa creado mediante tecnologías con el fin de emitir votos mediante un dispositivo o medio electrónico.

2.5 Tipos de Sistema de Voto Electrónico

Existen diferentes tipos de sistema de votación electrónica en la actualidad, según el Instituto Internacional para la Democracia y la Asistencia Electoral, (2011) menciona los siguientes tipos de sistema de voto electrónico:

- **Registro Electrónico Directo (RED).** Las RED pueden implementarse con o sin un comprobante impreso verificado por el votante (VVPAT, por sus siglas en inglés). Este último tiene el propósito de arrojar una prueba física de los votos emitidos.
- **Reconocimiento Óptico de Marcas (OMR, por sus siglas en inglés),** que funcionan a partir de lectores ópticos que reconocen la opción marcada por el votante en una papeleta especial. Los sistemas OMR pueden funcionar ya sea mediante un conteo centralizado (de forma que las papeletas pasan por un lector óptico en centros especiales de escrutinio) o mediante sistemas de votación y conteo de lector óptico (PCOS, por sus siglas en inglés), en los que los votos son registrados por el lector óptico y contabilizados en las mesas directamente, en el momento en que el elector introduce la papeleta en la máquina de votación.
- **Impresoras de papeletas electrónicas (EBP, por sus siglas en inglés).** Estas máquinas similares a las RED producen un papel para ser leído por la máquina o un comprobante electrónico que contiene la opción escogida por el elector. Este comprobante se introduce en otro lector óptico de papeletas, el cual hace el conteo de forma automática.
- **Sistemas de votación en línea.** Los votos son transmitidos por internet a un servidor central para su conteo. Pueden ser emitidos ya sea desde computadoras públicas, desde kioscos ubicados en las mesas de votación, o bien –y esto es lo más común– desde cualquiera computadora con conexión a internet accesible para los votantes. (p.11-12)

Por otra parte, Téllez (2010) identifica cuatro tipos de sistema que existen y se aplican, en algunas democracias occidentales:

a. **Sistema de votación mediante tarjeta perforada**: afecta al elector en la fase de establecimiento de sus preferencias, ya que debe perforar su opción en una tarjeta a través de un aparato (no electrónico, sino más bien mecánico). En un segundo momento, la tarjeta es introducida en una urna tabulador capaz de realizar el recuento de las perforaciones asignadas a cada opción. Este sistema es todavía muy utilizado en varios estados de EUA, a pesar de haber quedado bastante obsoleto.

b. **Sistemas de voto mediante un aparato lector:** es la evolución del sistema anterior. Se trata de aparatos capaces de "leer" marcas realizadas por el votante en una papeleta con un bolígrafo. Es el mismo sistema utilizado para el tratamiento de algunas loterías o tests. En esta ocasión, de nuevo podemos decir que el votante no entra en contacto directamente con la tecnología. Pero sí su papeleta —que sigue siendo de formato papel— cuando se introduce en el aparato lector y de recuento. En la actualidad, el aparato lector ha sido desarrollado de manera que ya no sólo reconoce cruces o marcas, sino también caracteres como números (que permitirían ordenar opciones) o incluso palabras.

c. **Sistemas de voto mediante aparatos de grabación directa:** con este tipo de sistema, el votante entra totalmente en contacto con la tecnología en todas las fases de la emisión de su voto. Se trata de aparatos similares a los cajeros automáticos, en los que el elector establece sus preferencias gracias a una pantalla táctil o a una pantalla y un teclado. En algunos casos, el propio aparato registra el voto. En otras, el voto se graba en un soporte externo que el votante ha introducido previamente en el aparato (por ejemplo, una tarjeta magnética). Tras emitir su voto, el votante utiliza su tarjeta a modo de una papeleta tradicional, introduciéndola en una urna, que a su vez será un aparato lector de tarjetas magnéticas y que realizará el recuento.

d. **Sistema de voto electrónico remoto:** este sistema de votación prevé que el votante no deba desplazarse hasta el colegio electoral y pueda emitir su voto a través de la red. Puede tratarse de una red interna y controlada por la propia institución que organiza la convocatoria, o puede realizarse la votación desde cualquier plataforma conectada a internet (principalmente un ordenador, pero también una agenda electrónica o un teléfono móvil).

Según una publicación de un artículo del periódico El Diario (2018) menciona cinco tipos de sistema para emitir votos:

- En papel

El de voto electrónico en papel es mediante máquina de votar o tarjetas perforadas, sistemas de votación de escaneo óptico, sistemas de marcado y más tarde sistemas de votación con lápiz óptico incluyeron un Marcador Electrónico de Papeletas (EBM)".

- Red pública

Asimismo, el de votación de red pública significa que los datos de la votación pueden ser transmitidos como papeletas individuales tal como han sido emitidos, en paquetes de datos a lo largo del día de la elección, o como un paquete al final de la elección. Esto incluye el voto por la Internet como por vía telefónica.

- Registro Directo

En el voto electrónico de registro directo se graban los votos por medio de una papeleta de votación en forma de pantalla provista de componentes mecánicos, es decir botones o pantallas de digitación, que pueden ser activados por el votante.

- Boleto único

  En cambio, el de boleto único electrónico, se trata de una máquina con una pantalla táctil, una impresora, una lectograbadora de chips y boletas de votación que contiene un chip en su interior.

- Votación por Internet

  El tipo de votación por internet significa usar lugares remotos, es decir desde cualquier computadora habilitada para este hecho y conectada a la red para marcar en las casillas computarizadas.

El tipo de sistema de votación electrónica que tomaremos para este trabajo será la votación en línea, donde el usuario o la persona que emita el voto lo realizará mediante un ordenador cuyo dato viajara hacia un servidor central.

2.6 Blockchain

Una blockchain no es otra cosa que una base de datos que se halla distribuida entre diferentes participantes, protegida criptográficamente y organizada en bloques de transacciones relacionados entre sí matemáticamente. Expresado de forma más breve, es una base de datos descentralizada que no puede ser alterada. Otro elemento muy importante a tener en cuenta en ella es que, por definición, se trata de un sistema que permite que partes que no confían plenamente unas en otras puedan mantener un consenso sobre la existencia, el estado y la evolución de una serie de factores compartidos. (Preukschat, Kuchkovsky, & Gómez, 2017, p.23)

Según auren; ACCID; ALHOS; UPF; Barcelona School; Economistas contables, (2019) la tecnología blockchain “Es una base de datos distribuida donde cada nodo o usuario en la red ejecuta y registra transacciones agrupándolas en forma de bloques”.

Una cadena de bloques, conocida en inglés como blockchain, es una estructura de datos en la que la información contenida se agrupa en conjuntos (bloques) a los que se les añade metainformaciones relativas a otro bloque de la cadena anterior en una línea temporal, de manera que, gracias a técnicas criptográficas, la información contenida en un bloque solo puede ser repudiada o editada modificando todos los bloques posteriores. Esta propiedad permite su aplicación en un entorno distribuido de manera que la estructura de datos blockchain puede ejercer de base de datos pública no relacional que contenga un histórico irrefutable de información. (Wikipedia, 2020)

En resumen, se puede decir que una blockchain es una estructura de datos donde los datos se agrupan en bloques y se hallan protegidas criptográficamente.

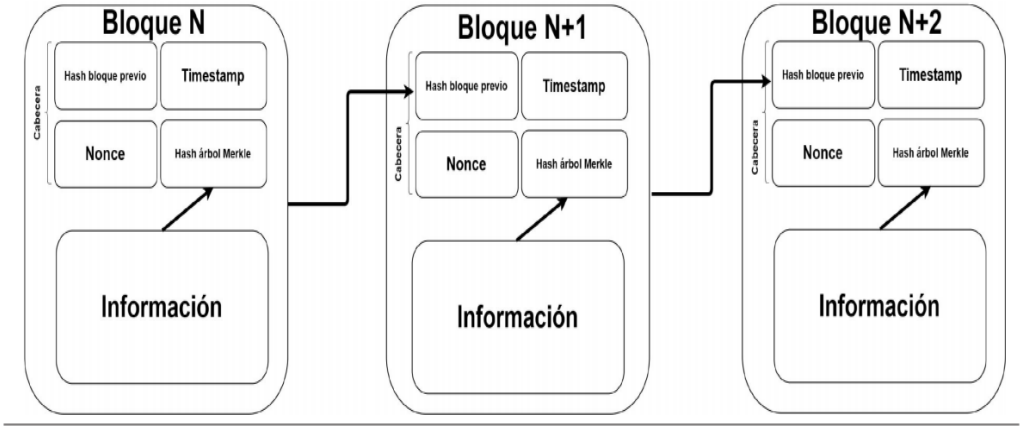


Figura 2.1 Estructura de los bloques de la blockchain de bitcoin.
Fuente: Dolader, Bel, & Muñoz, 2017

Existe dos tipos de blockchain: públicas y privadas.

- Las públicas son, por ejemplo, sobre las que trabajan bitcoin o ethereum, en donde el público en general tiene acceso. Es aquella en la que no hay restricciones ni para leer los datos de la cadena de bloques (los cuales pueden haber sido cifrados) ni para enviar transacciones para que sean incluidas en la cadena de bloques. En ellas es fácil entrar y salir, son transparentes, están construidas con precaución para la operación en un entorno no confiable.

- En las privadas solo pueden entrar quienes digan los propietarios. Es aquella en la que tanto los accesos a los datos de la cadena de bloque como el envío de transacciones para ser incluidas, están limitadas a una lista predefinida de entidades. (Navarro, 2017, p.3)

2.6.1 Bloque

Un bloque es un conjunto de transacciones confirmadas e información adicional que se ha incluido en la cadena de bloques. Cada bloque que forma parte de la cadena (menos el primer bloque que inicia la cadena) está formado por:

1. Un código alfanumérico que enlaza con el bloque anterior

2. El “paquete” de transacciones que incluye

3. Otro código alfanumérico que enlazará con el siguiente bloque.

(Navarro, 2017)

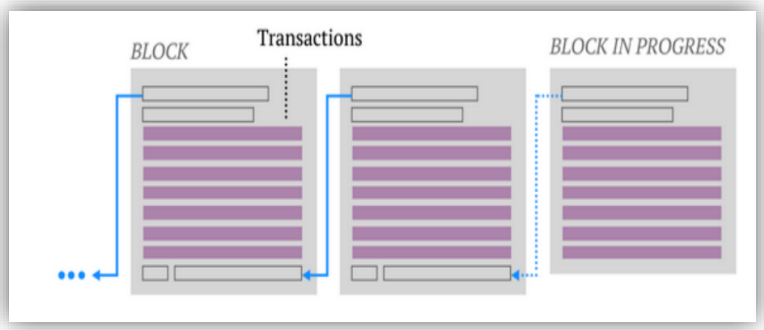


Figura 2.2 Cadena de bloques..
Fuente: Benjamin Y. Navarro; 2017.

2.6.2 Nodos

Son computadoras conectadas a la red utilizando un software que almacena y distribuye una copia actualizada en tiempo real del blockchain. 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 minería, se encargan además de escuchar

nuevas transacciones y agruparlas en bloques para proponerlos como trabajo a los mineros, que luego de ser confirmados son propagados a la red y añadidos a la cadena (Navarro, 2017).

Según Preukschat, Kuchkovsky, & Gómez (2017) “puede ser un ordenador personal o, según la complejidad de la red, una mega computadora. Con independencia de la capacidad de cómputo, todos los nodos han de poseer el mismo software/protocolo para comunicarse entre sí”.

2.6.3 Red P2P

Arquitectura de red distribuida donde todos los elementos son vistos como nodos, generando que la percepción de cliente/servidor desaparezca, ya que cada uno de ellos realiza las dos funciones. Estas redes fueron diseñadas para compartir tanto información como recursos de cualquier tipo (audio, vídeo, texto) sin la necesidad de disponer de un sistema central o servidor, ya que cada nodo comparte sus recursos con el resto de la red al mismo tiempo que solicita al resto de nodos los recursos que poseen. (Navarro Sánchez, 2019, p.5)

2.6.4 Red descentralizada

Un sistema descentralizado: a diferencia de un sistema centralizado, donde toda la información está controlada por una única entidad, aquí son todos los ordenadores conectados los que controlan la red porque todos son iguales entre sí; es decir, no hay una jerarquía entre los nodos, al menos en una blockchain pública. En una privada sí puede haber jerarquía. (Preukschat, Kuchkovsky, & Gómez, 2017)

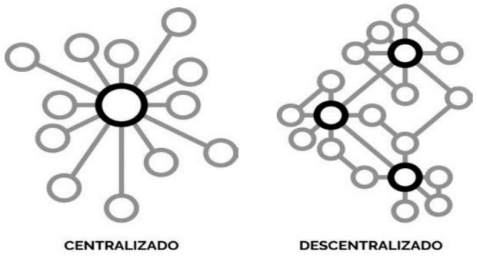


Figura 2.3 Modelo de red centralizada y de red descentralizada.
Fuente: Preukschat, Kuchkovsky, & Gómez, 2017

La blockchain se compone de tres partes importantes que conjuntamente realizan el trabajo propuesto, según (Preukschat, Kuchkovsky, & Gómez, 2017) son:

- **La criptografía:** por tal entendemos un procedimiento que, utilizando un algoritmo con clave (clave de cifrado), transforma un mensaje sin atender a su estructura lingüística o significado, de tal forma que sea incomprensible o, al menos, difícil de comprender, a toda persona que no tenga la clave secreta (clave de descifrado) del algoritmo empleado. En la blockchain, la criptografía tiene la responsabilidad de proveer un mecanismo infalible para la codificación segura de las reglas del protocolo que rigen el sistema. Es también fundamental para evitar la manipulación, hurto o introducción errónea de información en la cadena de bloques, así como la responsable de generar firmas e identidades digitales encriptadas.

- **La cadena de bloques o blockchain:** es la base de datos diseñada para el almacenamiento de los registros realizados por los usuarios. Todas las blockchains han de actuar bajo las mismas reglas o protocolo para dar validez al bloque — y a la información recogida— e incorporarlo a la cadena de bloques. Una vez realizada esta tarea, la cadena continuará con la emisión del siguiente bloque, permaneciendo inalterable la información registrada a través de la criptografía. Esta forma de obrar elimina la necesidad de un tercer ente de confianza.

- **Un consenso:** se trata de una parte imprescindible entre los usuarios de la blockchain. Este consenso se sustenta en un protocolo común que verifica y confirma las transacciones realizadas, y asegura la irreversibilidad de las mismas. De igual modo, este consenso debe proporcionar a todos los usuarios una copia inalterable y actualizada de las operaciones realizadas en la blockchain. (p.26-27)

2.7 Protocolo

Rodriguez Aragón, (2020) define protocolo como “Conjunto de reglas y especificaciones técnicas que permiten la comunicación entre extremos de manera fiable”.

En informática y telecomunicación, un protocolo de comunicaciones es un sistema de reglas que permiten que dos o más entidades de un sistema de comunicación se comuniquen entre ellas para transmitir información por medio de cualquier tipo de variación de una magnitud física. Se trata de las reglas o el estándar que define la sintaxis, semántica y sincronización de la comunicación, así como también los posibles métodos de recuperación de errores. (WIKIPEDIA, 2020)

Se puede decir que un protocolo es un conjunto de reglas, conjunto de instrucciones, en el área de la blockchain son algoritmos.

2.8 Protocolo de Internet TCP/IP

El Protocolo de Internet es un protocolo de comunicación de datos digitales clasificado funcionalmente en la capa de red según el modelo internacional OSI. Su función principal es el uso bidireccional en origen o destino de comunicación para transmitir datos mediante un protocolo no orientado a conexión que transfiere paquetes conmutados a través de distintas redes físicas previamente enlazadas según la norma OSI de enlace de datos. (Wikipedia, 2020)

IP y TCP son un par de protocolos bien compenetrados. El IP es un protocolo de interconexión de red orientado a datagrama. Por tanto, no dispone del concepto de circuito virtual, de manera que no es capaz de recuperar tramas perdidas, ni de garantizar que las tramas se entregarán en el orden correcto puesto que los paquetes pueden seguir caminos diferentes y, por tanto, sufrir retardos diferentes, ni que el ritmo de recepción sea el adecuado para que el receptor procese convenientemente los datos (Barceló Ordinas, Íñigo Griera, Martí Escalé, & Peig Olivé, 2004).

2.8.1 Estructura de protocolos de internet

De hecho, podríamos considerar que el modelo de la red Internet consta sólo de cuatro partes o niveles; es decir, todo lo que hay por debajo del IP, el IP, el TCP y todo lo que hay por encima del TCP (Barceló Ordinas, Íñigo Griera, Martí Escalé, & Peig Olivé, 2004)

a) **Por debajo de IP. A este nivel, en el entorno Internet**, se le llama nivel de red local o, simplemente, nivel de red. Por norma general, está formado por una red LAN, o WAN (de conexión punto a punto) homogénea. Todos los equipos conectados a Internet implementan dicho nivel.

b) **Nivel IP o nivel Internet (nivel de Internetworking).** Este nivel confiere unidad a todos los miembros de la red y, por consiguiente, es el que permite que todos se puedan interconectar, con independencia de si se conectan a la misma por medio de línea telefónica, ISDN o una LAN Ethernet. El direccionamiento y la asignación de direcciones constituyen sus principales funciones. Todos los equipos conectados a Internet implementan este nivel.

c) **Nivel TCP o nivel de transporte.** Este nivel confiere fiabilidad a la red. El control de flujo y de errores se lleva a cabo principalmente dentro de este nivel, que sólo es implementado por los equipos usuarios de la red Internet o por los terminales de Internet. Los equipos de conmutación (direccionadores o routers) no lo necesitan.

d) **Por encima de TCP.** Nivel de aplicación: Este nivel corresponde a las aplicaciones que utilizan Internet: clientes y servidores de WWW, correo electrónico, FTP, etc. Por ello se le denomina nivel de aplicación. Sólo es implementado por los equipos usuarios de la red Internet o los terminales de Internet. Los equipos de conmutación no lo utilizan.

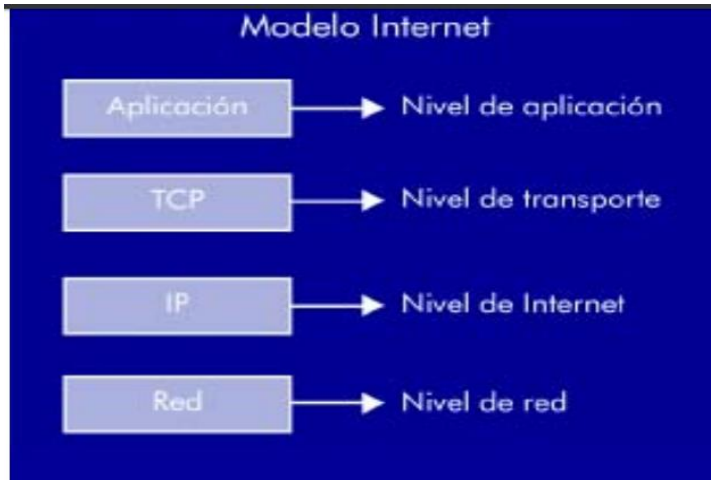


Figura 2.4 Modelo Internet.
Fuente: Barceló Ordinas, Íñigo Griera, Martí Escalé, & Peig Olivé, 2004

Es importante destacar que sólo los equipos terminales implementan todos los niveles; los equipos intermedios únicamente implementan el nivel de red y el nivel IP.

En los niveles intermedios existen otros protocolos complementarios, además de TCP e IP. La figura siguiente sería un mapa bastante completo de los protocolos que se usa en Internet:

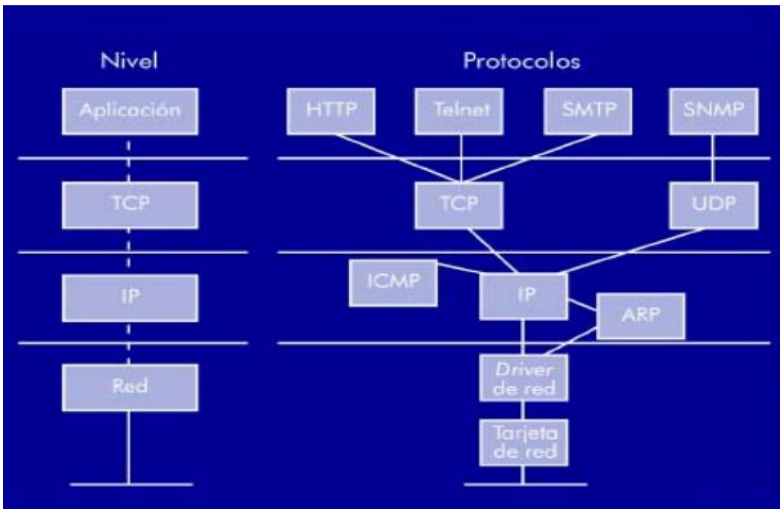


Figura 2.5 Protocolos de Internet.
Fuente: Barceló Ordinas, Íñigo Griera, Martí Escalé, & Peig Olivé, 2004.

2.9 Protocolos Blockchain

Un protocolo estándar: en 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. (Preukschat, Kuchkovsky, & Gómez, 2017)

El protocolo de consenso busca asegurarse de que el próximo bloque de transacciones que sea agregado a la cadena represente “la única versión de la verdad”. Debe ser diseñado de tal modo que evite que actores maliciosos introduzcan cambios ilegítimos en el registro. (Federico, 2019)

Los protocolos blockchain son algoritmos que hacen posible que las transacciones se realicen de forma segura y eficiente. Para el presente trabajo de investigación, estos algoritmos hacen posible que las emisiones de votos sean de forma segura.

2.10 Ingeniería de Software

Sommerville, (2011) define:” La ingeniería de software es una disciplina de ingeniería que se interesa por todos los aspectos de la producción de software, desde las primeras etapas de la especificación del sistema hasta el mantenimiento del sistema después de que se pone en operación”. (p.7)

2.10.1 Ingeniería Web

Pressman, (2010) define:” La ingeniería web es una versión adaptada del enfoque de ingeniería de software que se presenta en todo este libro. Propone una estructura ágil, pero disciplinada, para construir sistemas y aplicaciones basados en web con calidad industrial”. (p.317)

2.10.2 Metodología de Desarrollo de Software (UWE)

Ingeniería Web basada en UML (UWE) es un enfoque que proporciona un conjunto de elementos web para el modelar la estructura del sistema web.

UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto una de sus características principales es la personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario. (Quishpe, 2013)

UWE se especializa en la especificación de aplicaciones que se adaptan, y por eso hace énfasis especial en las características de personalización, y la definición de los modelos de usuario o en un patrón de características de navegación basado en preferencias, tareas o conocimiento. Otros aspectos de interés de la metodología UWE es la orientación a objetos, usuarios y la definición de un modelo de referencia que da soporte a la metodología y formaliza los modelos por el grado de restricciones y definiciones que proporciona. (Hernandez, 2010, p.6-7)

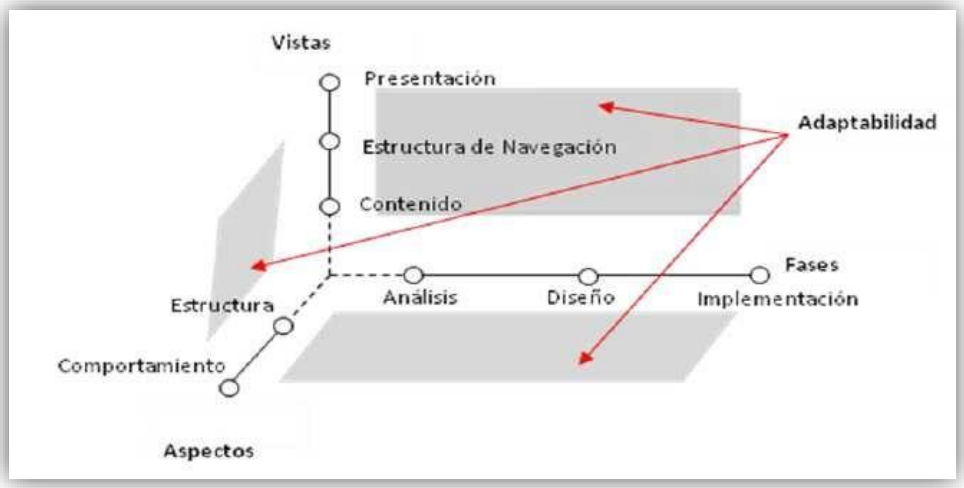


Figura 2.6 Dimensiones del Modelado (UWE).
Fuente: Nolivos, 2013.

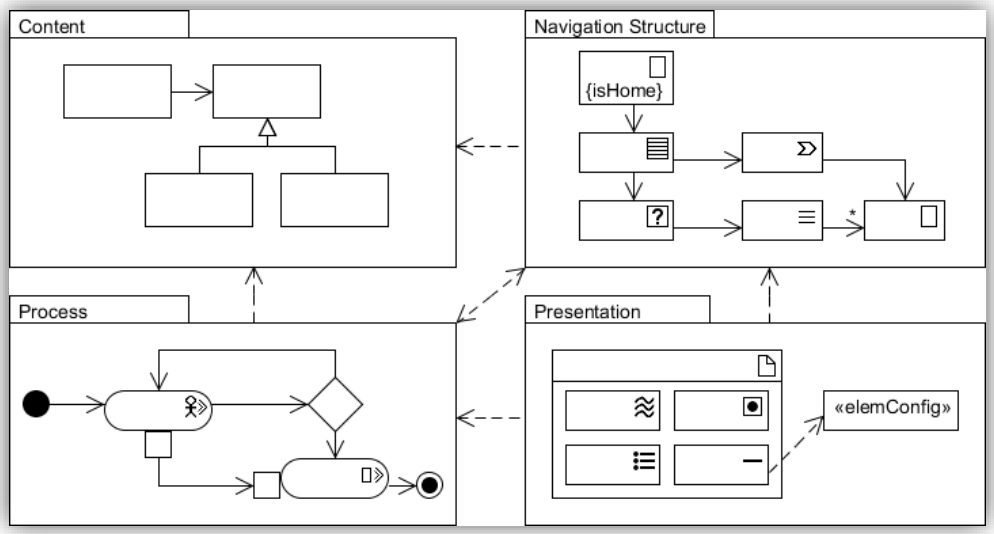


Figura 2.7 Vista general de modelos UWE
Fuente: Nolivos, 2013.

Según Nolivos (2013) los modelos de la metodologia UWE son:

a) Modelo de Requerimientos

El modelo de requerimientos permite diferenciar los procesos de navegación de los procesos del negocio mismo mediante el uso de diagramas de casos de uso para la captura de requisitos, que da como resultado un modelo de casos de uso acompañado de documentación que describe las reglas de adaptación, los usuarios y las interfaces.

b) Modelo Conceptual

Un diagrama de clases se utiliza para representar gráficamente un modelo conceptual como visión estática que demuestre una colección de los elementos estáticos del dominio. La construcción de este modelo conceptual se debe llevar a cabo de acuerdo con los casos de uso que se definen en la especificación de requerimientos.

c) Modelo Navegacional

El modelo de navegación de una aplicación Web comprende la especificación de qué objetos pueden ser visitados mediante la navegación a través de la aplicación Web y las asociaciones entre ellos.

Su objetivo principal es representar el diseño y estructura de las rutas de navegación al usuario para evitar la desorientación en el proceso de navegación. Este modelo se destaca en el marco de UWE como el más importante, pues con él se pueden representar elementos estáticos, a la vez que se pueden incorporar lineamientos semánticos de referencia para las funcionalidades dinámicas de una aplicación Web.

d) Modelo de Presentación

El Modelo de Navegación no indica cuáles son las clases de navegación y de proceso que pertenecen a una página web. Se puede usar un Diagrama de Presentación con el fin de proveer esta información.

Permite la especificación lógica de la aplicación Web, una representación física puede ser construida sobre este método lógico. Representa las interfaces del usuario por medio de vistas estándares de interacción UML. Las clases del modelo de presentación representan páginas Web o parte de ellas, organizando la composición de los elementos de la interfaz de usuario y las jerarquías del modelo de presentación

e) Modelo de Procesos

Este modelo representa la parte dinámica de la aplicación Web, especificando la funcionalidad de las transacciones y de los flujos de trabajo complejos de las actividades; contrario al modelo navegaciones, que representa la parte estática de la información. (p.39-45)

Tabla 2.1
Etapas de Desarrollo Web basado en UWE.

ACTIVIDAD	*TÉCNICA*	*ENTREGABLE*
Análisis de Requerimientos	*Casos de Uso*	*Diagramas de casos de uso*
Modelo Conceptual	*Diagrama de Clases*	*Diagramas de Clases*
	Diagrama de Secuencia	*Diagramas de Secuencia*
	Diagrama de Estado	*Diagramas de Estados*
	Diagrama de Despliegue	*Diagramas de Despliegue*
	Diagrama de Implementación	*Diagramas de implementación*
Modelo Navegacional	*Diagrama de Navegación*	*Diagramas de Navegación*
Modelo de Presentación	*Diagrama de Presentación*	*Diagramas de Presentación*
Modelo de Tareas	*Diagrama de Actividad*	*Diagramas de Actividades*

Fuente: Nolivos, 2013.

2.10.3 Arquitectura de software

Bass, Clements y Kazman (2003) definen este término de la manera siguiente:

“La arquitectura del software de un programa o sistema de cómputo es la estructura o estructuras del sistema, lo que comprende a los componentes del software, sus propiedades externas visibles y las relaciones entre ellos”.

La arquitectura de software es importante porque afecta el desempeño y la potencia, así como la capacidad de distribución y mantenimiento de un sistema

Bass y sus colaboradores (2003) analizan tres ventajas de diseñar y documentar de manera explícita la arquitectura de software:

a) **Comunicación con los participantes** La arquitectura es una presentación de alto nivel del sistema, que puede usarse como un enfoque para la discusión de un amplio número de participantes.

b) **Análisis del sistema** En una etapa temprana en el desarrollo del sistema, aclarar la arquitectura del sistema requiere cierto análisis. Las decisiones de diseño arquitectónico tienen un efecto profundo sobre si el sistema puede o no cubrir requerimientos críticos como rendimiento, fiabilidad y mantenibilidad.

c) **Reutilización a gran escala** Un modelo de una arquitectura de sistema es una descripción corta y manejable de cómo se organiza un sistema y cómo interoperan sus componentes. Por lo general, la arquitectura del sistema es la misma para sistemas con requerimientos similares y, por lo tanto, puede soportar reutilización de software a gran escala.

2.10.4 Arquitectura cliente – servidor

Modelo arquitectónico para sistemas distribuidos donde la funcionalidad del sistema se ofrece como un conjunto de servicios proporcionados por un servidor. A ellos acceden computadoras cliente que usan los servicios. Variantes de este enfoque, como las arquitecturas cliente-servidor de tres capas, usan múltiples servidores. (Sommerville, 2011, p.733)

El modelo cliente-servidor, la computación cliente-servidor, la tecnología cliente-servidor y la arquitectura cliente-servidor se refieren a un modelo de diseño que podemos considerar como aplicaciones que se ejecutan en una red. En términos muy básicos, podemos imaginar que el cliente hace (y el servidor ejecuta, o cumple de cierta forma con) la solicitud. Eso se consideraría una arquitectura cliente-servidor de dos niveles. (Kendall & Kendall, 2011, p.529-530)

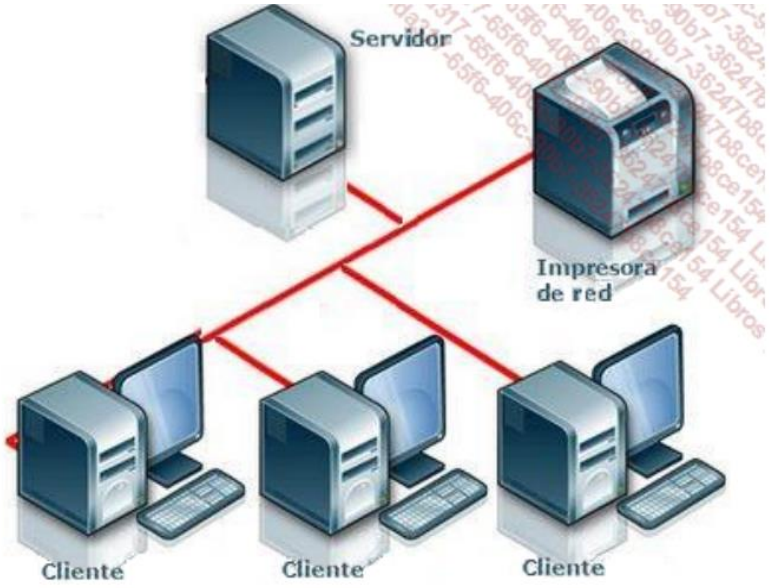


Figura 2.8 Arquitectura Cliente/servidor
Fuente: Barceló Ordinas, Íñigo Griera, Martí Escalé, & Peig Olivé, 2004.

a) Arquitectura cliente servidor dos niveles

Una arquitectura cliente-servidor de dos niveles es la forma más simple de arquitectura cliente-servidor. El sistema se implementa como un solo servidor lógico más un número indefinido de clientes que usan dicho servidor. (Sommerville, 2011)

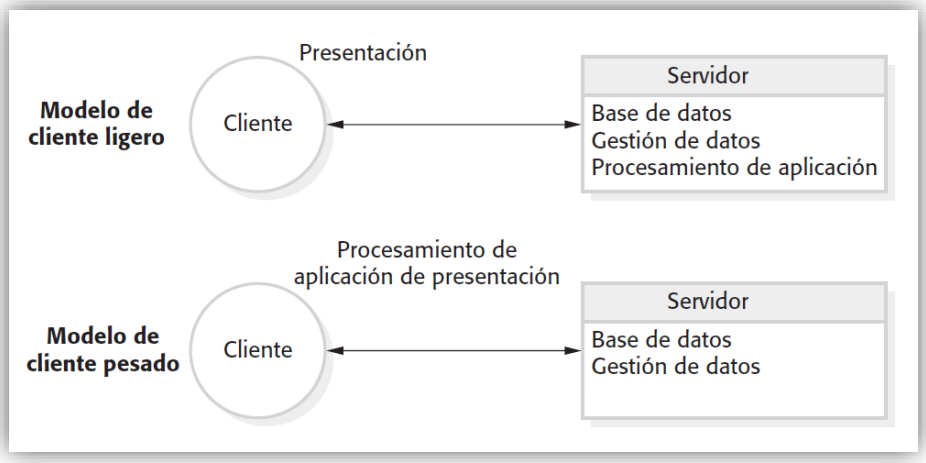


Figura 2.9 Cliente servidor de dos niveles
Fuente. Sommerville, 2011.

b) Arquitectura cliente servidor multinivel

El modelo cliente-servidor de tres niveles puede extenderse a una variante multinivel, donde se agregan servidores adicionales al sistema. Esto podría implicar el uso de un servidor Web para gestión de datos y servidores separados para procesamiento de aplicación y servicios de base de datos. Los sistemas multinivel también pueden usarse cuando las aplicaciones necesitan tener acceso y utilizar datos de diferentes bases de datos. En este caso, tal vez se requiera agregar un servidor de integración al sistema. El servidor de integración recolecta los datos distribuidos y los presenta al servidor de aplicación como si fuera de una sola base de datos. Como se estudiará en la siguiente sección, las arquitecturas de componentes distribuidos pueden usarse para implementar sistemas cliente servidor multinivel.

Los sistemas cliente-servidor multinivel que distribuyen el procesamiento de aplicación a través de varios servidores son inherentemente más escalables que las arquitecturas de dos niveles. El procesamiento de aplicación con frecuencia es la parte más volátil del sistema y puede actualizarse fácilmente, ya que se ubica centralmente. El procesamiento, en algunos casos, puede distribuirse entre la lógica de la aplicación y los servidores de gestión de datos; por lo tanto, conduce a una respuesta más rápida a las solicitudes del cliente. (Sommerville, 2011)

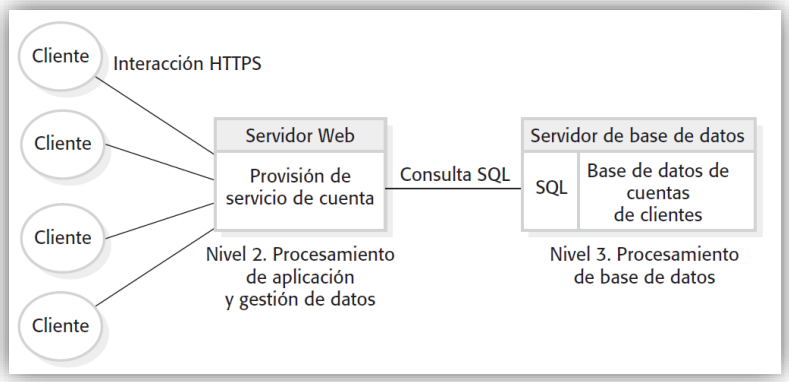


Figura 2.10 Cliente servidor de tres niveles.
Fuente: Sommerville, 2011.

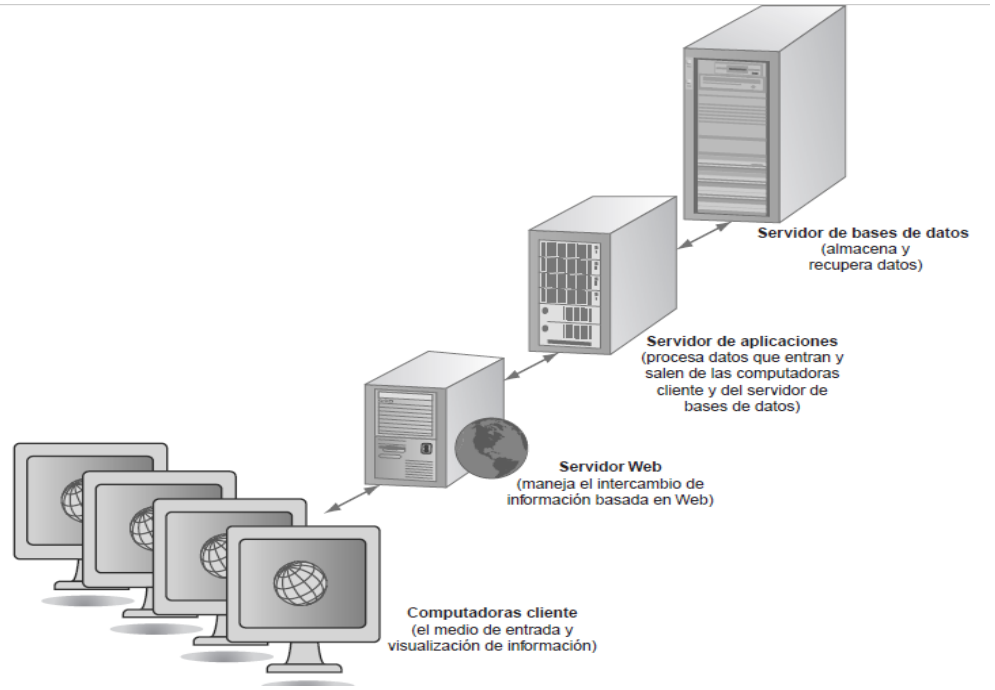


Figura 2.11 Cliente servidor de multinivel
Fuente: Kendall & Kendall, 2011.

2.10.5 Seguridad

En una entrevista para ComputerWorld, el autor y experto en seguridad Gary McGraw comenta lo siguiente:

La seguridad del software se relaciona por completo con la calidad. Debe pensarse en seguridad, confiabilidad, disponibilidad y dependencia, en la fase inicial, en la de diseño, en la de arquitectura, pruebas y codificación, durante todo el ciclo de vida del software [proceso]. Incluso las personas conscientes del problema de la seguridad del software se centran en las etapas finales del ciclo de vida. Entre más pronto se detecte un problema en el software, mejor. Y hay dos clases de problemas. Uno son los errores, que son problemas de implementación. El otro son las fallas del software: problemas de arquitectura en el diseño. La gente presta demasiada atención a los errores, pero no la suficiente a las fallas.

La seguridad del software examina las formas en las que las fallas generan condiciones que llevan a un peligro. Es decir, las fallas no se consideran en el

vacío, sino que se evalúan en el contexto de la totalidad del sistema basado en computadora y de su ambiente. (Pressman, 2010)

Criptografía

La palabra criptografía proviene en un sentido etimológico del griego Kriptos = ocultar, Graphos = escritura, lo que significaría ocultar la escritura, o en un sentido más amplio sería aplicar alguna técnica para hacer ininteligible un mensaje.

En un sentido más amplio, la Criptografía es la ciencia encargada de diseñar funciones o dispositivos, capaces de transformar mensajes legibles o en claro a mensajes cifrados de tal manera que esta transformación (cifrar) y su transformación inversa (descifrar) sólo pueden ser factibles con el conocimiento de una o más llaves (Granados, 2006).

- **Criptografía simétrica**

La clave usada para cifrar y descifrar es idéntica y deberá ser compartida entre el emisor y el receptor. Debido a esta circunstancia, el empleo de este tipo de criptosistemas precisa que emisor y receptor dispongan de un canal seguro para el intercambio de la clave, algo que no ocurre hasta la invención en 1976 de la criptografía de clave pública. (Ramió, 2020)

- **Criptografía asimétrica**

La criptografía asimétrica es por definición aquella que utiliza dos claves diferentes para cada usuario, una para cifrar que se le llama clave pública y otra para descifrar que es la clave privada. El nacimiento de la criptografía asimétrica se dio al estar buscando un modo más práctico de intercambiar las llaves simétricas Diffie y Hellman, proponen una forma para hacer esto, sin embargo no fue hasta que el popular método de Rivest Shamir y Adleman RSA publicado en 1978, cuándo toma forma la criptografía asimétrica, su funcionamiento está basado en la imposibilidad computacional de factorizar números enteros grandes. (Angel, 2020)

En este caso cada entidad (usuario, máquina, etc.) dispone de una clave pública y de una clave privada, inversas entre sí, que pueden ser usadas para cifrar un mensaje la primera o descifrar un criptograma la segunda. Cuando la clave pública es usada para cifrar, el descifrado del criptograma resultante debe ser realizado con la clave privada (y viceversa). El sistema de cifra será seguro si es computacionalmente difícil (en cómputo y en tiempo) averiguar la clave privada conociendo solamente la clave pública. (Ramió, 2020)

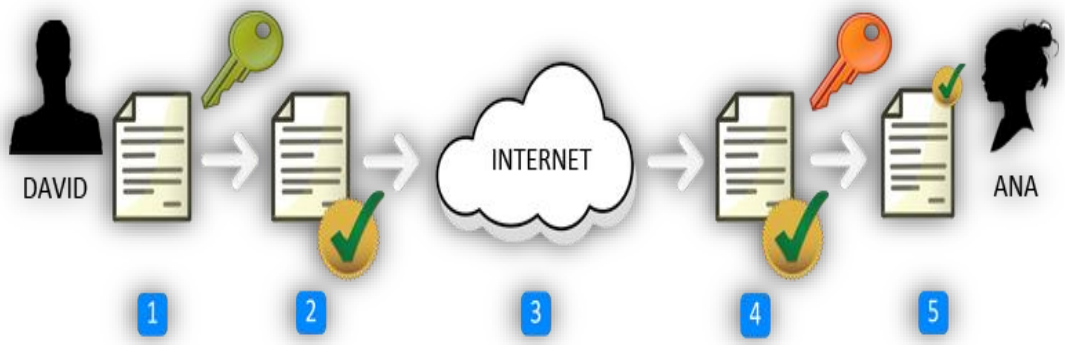


Figura 2.12 Criptografía Asimétrica.
Fuente: Wikipedia, 2020.

- **Función Hash**

Un hash, o función hash, consiste en el mapeo de mensajes de longitudes arbitrarias a valores de n bits. Para que un hash sea considerado criptográfico, debe cumplir con la condición de que el mapeo sea unidireccional, de esta forma se requerirán 2^n operaciones para encontrar un mensaje que pueda ser encriptado al hash considerado. Por otro lado, se dice que un hash es resistente a colisiones si la obtención de un mismo resultado a partir de dos mensajes diferentes es computacionalmente inviable. Estas propiedades inherentes al algoritmo SHA-256 explican su amplia utilización en el firmado digital y la protección de contraseñas. (Universidad Tecnológica Nacional Facultad Regional Buenos Aires, 2015)

El proceso de las funciones hash se le conoce como criptografía, es decir que es capaz de transformar cualquier entrada, ya se texto, una imagen jpg, png o un archivo transformarlo a un único código. Pero existen diferentes modelos de algoritmos que nos permiten realizar estos procesos. (Sánchez, Domínguez, & Velásquez, 2019)

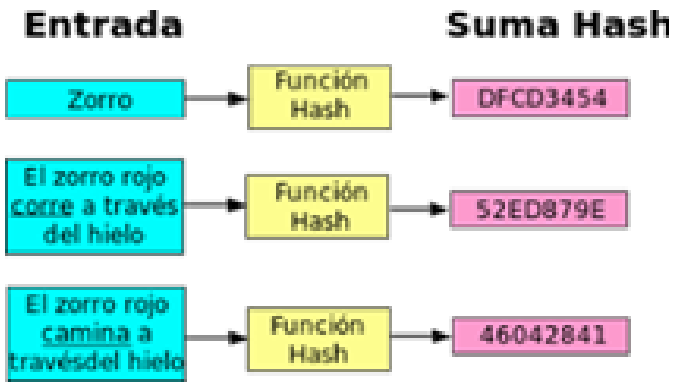


Figura 2.13 Proceso criptográfico de las funciones hash.
Fuente: Sánchez, Domínguez, & Velásquez, 2019.

Tabla 2.2
Fortalezas de las funciones hash más conocidas.

Función hash	Fortaleza teórica	Fortaleza real (2016)
MD5	2 64	2 24
SHA-1	2 80	2 57
SHA-256	2 128	2 128
SHA-512	2 256	2 256
SHA-3 256	2 128	2 128
SHA-3 512	2 256	2 256

Fuente: Diversas en internet.

- **Funciones SHA**

SHA-2 es un conjunto de funciones hash criptográficas (SHA-224, SHA-256, SHA-384, SHA-512) diseñadas por la Agencia de Seguridad Nacional (NSA) y publicada en 2001 por el Instituto Nacional de Estándares y Tecnología (NIST) como un Estándar Federal de Procesamiento de la Información (FIPS).

La SHA-256 y SHA-512 son nuevas funciones hash con palabras de tamaño 32 y 64 bits, respectivamente. Usan diferentes desplazamientos y constantes, pero sus estructuras son por otra parte virtualmente idéntica, diferenciándose únicamente por el número de iteraciones (WIKIPEDIA, 2020).

SHA-256 es una función criptográfica de 256 bits de longitud sin clave, también llamada Manipulation Detection Code (MDC). (Universidad Tecnológica Nacional Facultad Regional Buenos Aires, 2015)

Conocedores que SHA-1 no era del todo seguro, la NSA desarrolla en 2001 una familia de algoritmos hash conocida como SHA-2, que consiste en un conjunto de cuatro funciones hash de 224, 256, 384 y 512 bits. En estos dos últimos se trabaja con palabras de 64 bits (Ramió Aguirre, 2020).

Algoritmo de encriptación RSA

RSA (Rivest, Shamir y Adleman) es un algoritmo de cifrado asimétrico desarrollado en el año 1977 por los anteriormente citados. Este algoritmo se basa en escoger 2 números primos grandes elegidos de forma aleatoria y mantenidos en secreto. La principal ventaja de este algoritmo desde el punto de vista de seguridad radica en la dificultad a la hora de factorizar números grandes. (Sánchez, Rodríguez, & Notario, 2020)

De entre todos los algoritmos asimétricos, quizá RSA sea el más sencillo de como prender e implementar. Como ya se ha dicho, sus claves sirven indistintamente tanto para codificar como para autentificar.

RSA se basa en la dificultad para factorizar grandes números. Las claves pública y privada se calculan a partir de un número que se obtiene como producto de dos primos grandes. El atacante se enfrentará, si quiere recuperar un texto claro a partir del criptograma y la clave pública, a un problema de factorización o tendrá que resolver un logaritmo discreto. (Lucena Lopez, 2011)

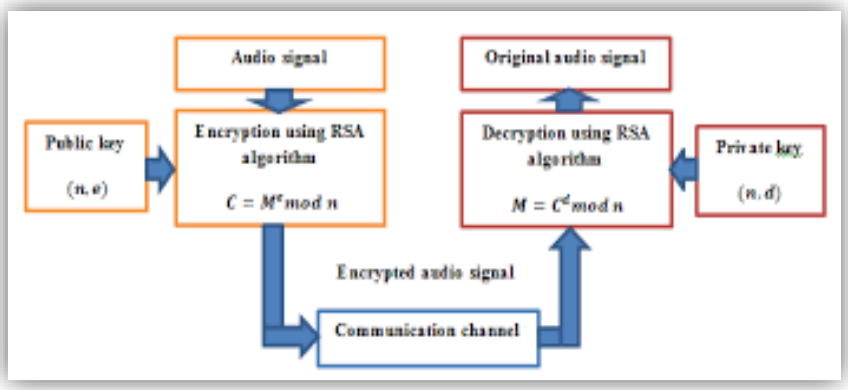


Figura 2.14 Diagrama de bloques de la metodología presentada.
Fuente: Yousif, 2018.

Funciones para cifrado y descifrado

• Para cifrar se aplica la función de cifrado:

$$c = m^e \, mod \, n$$

• Para descifrar se aplica la función de descifrado:

$$m = c^d \, mod \, n$$

Dónde:

c es el mensaje cifrado.

m es el mensaje a cifrar.

n es una clave pública obtenida por el producto de dos primos p y q.

e es una constante y pública.

d es el inverso multiplicativo de e.

Algoritmo

- Genere aleatoriamente dos primos grandes denotados como p y q los cuales corresponden a la clave privada.
- Se calcula la clave pública n como el producto de p y q.
- Calcular fi(n)=(p-1) * (q-1).
- Calcular e, considerando MCD (e, fi(n)) = 1, es decir e de debe ser un primo relativo a fi(n) para que la igualdad se cumpla.
- Mediante el algoritmo extendido de Euclides se busca el valor d que cumpla la relación: (e*d) MOD fi(n) = 1, es decir d = (fi(n)*y+1) /e, donde y es el cociente el cual debe probarse para valores enteros y = 1, 2, …
- El par de números (e, n) corresponden a la clave pública.
- El par de números (d, n) corresponden a la clave privada.
- Para cifrar se aplica la función de cifrado: $c = m^e \bmod n$
- Para descifrar se aplica la función de descifrado: $m = c^d \bmod n$

Firma Digital

Una firma digital es un mecanismo criptográfico que permite al receptor de un mensaje firmado digitalmente identificar a la entidad originadora de dicho mensaje (autenticación de origen y no repudio), y confirmar que el mensaje no ha sido alterado desde que fue firmado por el originador (integridad). (Wikipedia, 2020)

2.10.6 Procesos

Un proceso es un conjunto de actividades, acciones y tareas que se ejecutan cuando va a crearse algún producto del trabajo. Una actividad busca lograr un objetivo amplio (por ejemplo, comunicación con los participantes) y se desarrolla sin importar el dominio de la aplicación, tamaño del proyecto, complejidad del esfuerzo o grado de rigor con el que se usará la ingeniería de software. Una acción (diseño de la arquitectura) es un conjunto de tareas que producen un producto importante del trabajo (por ejemplo, un modelo del diseño de la arquitectura). Una tarea se centra en un objetivo pequeño, pero bien definido (por ejemplo, realizar una prueba unitaria) que produce un resultado tangible.

En el contexto de la ingeniería de software, un proceso no es una prescripción rígida de cómo elaborar software de cómputo. Por el contrario, es un enfoque adaptable que permite que las personas que hacen el trabajo (el equipo de software) busquen y elijan el conjunto apropiado de acciones y tareas para el trabajo. Se busca siempre entregar el software en forma oportuna y con calidad suficiente para satisfacer a quienes patrocinaron su creación y a aquellos que lo usarán. (Pressman, 2010)

Estructura de proceso

La estructura del proceso establece el fundamento para el proceso completo de la ingeniería de software por medio de la identificación de un número pequeño de actividades estructurales que sean aplicables a todos los proyectos de software, sin importar su tamaño o complejidad. Además, la estructura del proceso incluye un conjunto de actividades sombrilla que son aplicables a través de todo el proceso del software. Una estructura de proceso general para la ingeniería de software consta de cinco actividades:

- **Comunicación.**

  Antes de que comience cualquier trabajo técnico, tiene importancia crítica comunicarse y colaborar con el cliente (y con otros participantes).11 Se busca

entender los objetivos de los participantes respecto del proyecto, y reunir los requerimientos que ayuden a definir las características y funciones del software.

- **Planeación.**

  Cualquier viaje complicado se simplifica si existe un mapa. Un proyecto de software es un viaje difícil, y la actividad de planeación crea un "mapa" que guía al equipo mientras viaja. El mapa —llamado *plan del proyecto de software*— define el trabajo de ingeniería de software al describir las tareas técnicas por realizar, los riesgos probables, los recursos que se requieren, los productos del trabajo que se obtendrán y una programación de las actividades.

- **Modelado.**

  Ya sea usted diseñador de paisaje, constructor de puentes, ingeniero aeronáutico, carpintero o arquitecto, a diario trabaja con modelos. Crea un "bosquejo" del objeto por hacer a fin de entender el panorama general —cómo se verá arquitectónicamente, cómo ajustan entre sí las partes constituyentes y muchas características más—. Si se requiere, refina el bosquejo con más y más detalles en un esfuerzo por comprender mejor el problema y cómo resolverlo. Un ingeniero de software hace lo mismo al crear modelos a fin de entender mejor los requerimientos del software y el diseño que los satisfará.

- **Construcción.**

  Esta actividad combina la generación de código (ya sea manual o automatizada) y las pruebas que se requieren para descubrir errores en éste.

- **Despliegue.**

  El software (como entidad completa o como un incremento parcialmente terminado) se entrega al consumidor que lo evalúa y que le da retroalimentación, misma que se basa en dicha evaluación.

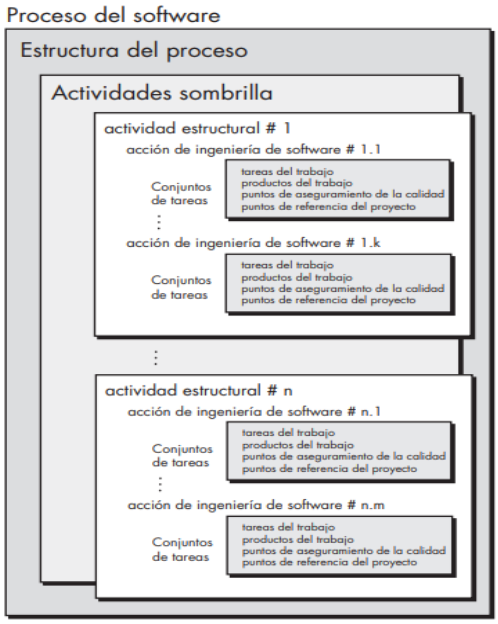


Figura 2.15 Estructura de un proceso de software.
Fuente: Pressman, 2010.

2.10.7 Calidad

a) Caja Negra

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra no son una alternativa para las técnicas de caja blanca. En vez de ello, es un enfoque complementario que es probable que descubra una clase de errores diferente que los métodos de caja blanca.

Las pruebas de caja negra intentan encontrar errores en las categorías siguientes: 1) funciones incorrectas o faltantes, 2) errores de interfaz, 3) errores en las estructuras de datos o en el acceso a bases de datos externas, 4) errores de comportamiento o rendimiento y 5) errores de inicialización y terminación. (Pressman, 2010, p.423)

b) Caja Blanca

La prueba de caja blanca, en ocasiones llamada prueba de caja de vidrio, es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba. Al usar los métodos de prueba de caja blanca, puede derivar casos de prueba que: 1) garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez, 2) revisen todas las decisiones lógicas en sus lados verdadero y falso, 3) ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas y 4) revisen estructuras de datos internas para garantizar su validez. (Pressman, 2010, p.414)

2.11 Método Científico

Proviene del griego: -meta = hacia, a lo largo- -odos = camino-; y del latín scientia = conocimiento; camino hacia el conocimiento, y presenta diversas definiciones debido a la complejidad de una exactitud en su conceptualización: "Conjunto de pasos fijados de antemano por una disciplina con el fin de alcanzar conocimientos válidos mediante instrumentos confiables", "secuencia estándar para formular y responder a una pregunta", "pauta que permite a los investigadores ir desde el punto A hasta el punto Z con la confianza de obtener un conocimiento válido". (Mamani Ortiz, 2014).

En términos generales, el método científico es inherente a la ciencia, tanto a la pura como a la aplicada. Sin método científico no puede haber ciencia. El método científico, como ya lo mencionamos, no es infalible, tampoco es autosuficiente, es decir, debe partir de algún conocimiento previo que se requiera concretar o bien ampliar, para posteriormente adaptarse a las especificaciones de cada tema, materia y/o especialidad. Integra una serie de procedimientos lógicos sistemáticos, racionales e intelectuales que permite resolver interrogantes (Maya, 2014, p.12).

El método científico es el conjunto de pasos, técnicas y procedimientos que se emplean para formular y resolver problemas de investigación mediante la prueba o verificación de hipótesis.

Previo a la aplicación del método científico debe ocurrir un hecho o fenómeno, es decir, cualquier suceso o cambio ocurrido en la naturaleza o en la sociedad, que pueda ser percibido y que sea de interés para el investigador. Una vez sucedido el hecho, se procede con el primer paso. (Arias, 2012, p.18-19)

1. Observación: consiste en la percepción del hecho o fenómeno.

2. Formulación del problema: se basa en la elaboración de una pregunta o interrogación acerca del hecho observado.

3. Formulación de hipótesis: radica en la producción de una suposición o posible respuesta al problema.

4. Verificación: consiste en someter a prueba la hipótesis mediante la recolección de datos.

5. Análisis: los datos obtenidos son procesados para así determinar cuáles confirman o niegan la hipótesis.

6. Conclusión: es la respuesta al problema, producto de la verificación y del análisis efectuado. Es importante señalar que en otros libros de texto pueden aparecer más o menos pasos, pero son los antes indicados los que constituyen la esencia del método científico.

2.12 Hipótesis

La hipótesis es la relación que se establece entre las variables de la investigación. Generalmente, la hipótesis se entiende como la posible solución al problema de la investigación, el cual se puede justificar de manera lógica relacionando las variables. Otros argumentan que la hipótesis es una herramienta de comprobación de las variables con la realidad.

En este sentido se puede afirmar que la hipótesis es importante porque ayuda a orientar los objetivos de la investigación. Así mismo, el investigador tiene la libertad de crear la hipótesis partiendo de la observación detallada de los hechos. Esto implica que la hipótesis no puede ser hecha al azar, ya que tiene que ir con el mismo orden en que se han planteado tanto el problema, como las variables del proyecto. (Arango Quintero, 2012)

2.12.1 Variables

Una variable es algo cuya propiedad puede estar sujeta al cambio, de manera que son parte esencial de la investigación (Sabino, 1980). Las variables son como las siguientes; sexo, religión, violencia y sociedad, escuelas de formación docente, entre otras. Para Sabino la variable es "cualquier característica o cualidad de la realidad que es susceptible de asumir diferentes valores, es decir, que puede variar, aunque para un objeto determinado que se considere puede tener un valor fijo" (1980). (Chávez Abad, 2015, p.47)

2.12.2 Tipos de variables

Según Sabino (1980) son:

- Variable independiente: corresponde a la propiedad que es la causa del fenómeno o el objeto a estudiar.
- Variable dependiente: es la propiedad que se observa y se mide para cambiarlo a través de la manipulación de la anterior variable.
- Variable interviniente: es aquella propiedad que de alguna forma afecta el resultado esperado.
- Variable cualitativa: es aquella que alude a los atributos del objeto o fenómeno.
- Variable cuantitativa: este tipo de variable admite medición, pues las características de este pueden darse en diversas formas de intensidad.

- Variable discreta: es aquella que no permite un punto medio entre los números.
- Variables de control: estas variables son controladas por el investigador, para manejar y dominar cualquier efecto colateral que pueda tener el fenómeno u objeto estudiado.

2.12.3 Tipos de Hipótesis

Según (Arias, 2012) menciona los siguientes tipos de Hipótesis:

- **Hipótesis de investigación**

Es la suposición que se aspira verificar o comprobar. También se le denomina hipótesis de trabajo. Éstas se clasifican en:

a) Explicativas: expresan la posible causa de un hecho.

b) Predictivas: son aquellas que plantean el posible efecto o consecuencia de un hecho.

c) Comparativas: contrastan resultados o características de grupos en condiciones diferentes.

d) Correlacionales: suponen una posible relación estadística entre variables cuantitativas.

e) Descriptivas: indican una probable relación no causal entre variables cualitativas.

- **Hipótesis alternativas**

Son aquellas que plantean opciones distintas a la hipótesis de trabajo o de investigación.

- **Hipótesis nula**

Es la que niega lo supuesto en la hipótesis de investigación. En el caso de comparación de grupos, expresa que no existen diferencias significativas entre los resultados obtenidos por éstos. Así mismo es contraria a la hipótesis no direccional.

Tabla 2.3
Tipos de Hipótesis

HIPÓTESIS	De investigación o de trabajo	Explicativa	
		Predictivo	Experimental
			No experimental
		Comparativa	Experimental
			No experimental
			Direccional
			No direccional
		Correlacional	
		Descriptiva	
	Alternativa		
	Nula		

Fuente: Arias, 2012.

2.13 Métrica de Calidad

2.13.1 ISO 9126

El estándar ISO 9126 se desarrolló con la intención de identificar los atributos clave del software de cómputo. Este sistema identifica seis atributos clave de la calidad:

- **Funcionalidad.** Grado en el que el software satisface las necesidades planteadas según las establecen los atributos siguientes: adaptabilidad, exactitud, interoperabilidad, cumplimiento y seguridad.
- **Confiabilidad.** Cantidad de tiempo que el software se encuentra disponible para su uso, según lo indican los siguientes atributos: madurez, tolerancia a fallas y recuperación.
- **Usabilidad.** Grado en el que el software es fácil de usar, según lo indican los siguientes subatributos: entendible, aprendible y operable.
- **Eficiencia.** Grado en el que el software emplea óptimamente los recursos del sistema, según lo indican los subatributos siguientes: comportamiento del tiempo y de los recursos.
- **Facilidad de recibir mantenimiento.** Facilidad con la que pueden efectuarse reparaciones al software, según lo indican los atributos que siguen: analizable, cambiable, estable, susceptible de someterse a pruebas.
- **Portabilidad.** Facilidad con la que el software puede llevarse de un ambiente a otro según lo indican los siguientes atributos: adaptable, instalable, conformidad y sustituible.

Igual que otros factores de la calidad del software estudiados en las subsecciones anteriores, los factores ISO 9126 no necesariamente conducen a una medición directa. Sin embargo, proporcionan una base útil para hacer mediciones indirectas y una lista de comprobación excelente para evaluar la calidad del sistema. (Pressman, 2010)

Tabla 2.4
Características de la norma ISO-9126 y aspectos que atiende cada una

Características	Preguntas
Funcionalidad	¿Las funciones y propiedades satisfacen las necesidades explicitas e implícitas?
Confiabilidad	¿Puede mantener el nivel de rendimiento, bajo ciertas condiciones y por cierto tiempo?
Usabilidad	¿El software es fácil de usar y aprender?
Eficiencia	¿Es rápido y minimalista en cuanto al uso de recursos?
Mantenibilidad	¿Es fácil de modificar y verificar?
Portabilidad	¿Es fácil de transferir de un ambiente a otro?

Fuente: Moreno, Toledo, Lopez, & Cruz, 2020.

2.14 Costos

2.14.1 COCOMO

COCOMO' 81 está compuesto por tres modelos que corresponden a distintos niveles de detalle y precisión. Mencionados en orden creciente son: Modelo Básico, Intermedio y Detallado. La estimación es más precisa a medida que se toman en cuenta mayor cantidad de factores que influyen en el desarrollo de un producto de software. (Gómez, López, Migani, & Otazú, 2020)

COCOMO II es un modelo que permite estimar el coste, esfuerzo y tiempo cuando se planifica una nueva actividad de desarrollo software. Está asociado a los ciclos de vida modernos. El modelo original COCOMO ha tenido mucho éxito, pero no puede emplearse con las prácticas de desarrollo software más recientes tan bien como con las prácticas tradicionales. (Moreno A. M., 2012).

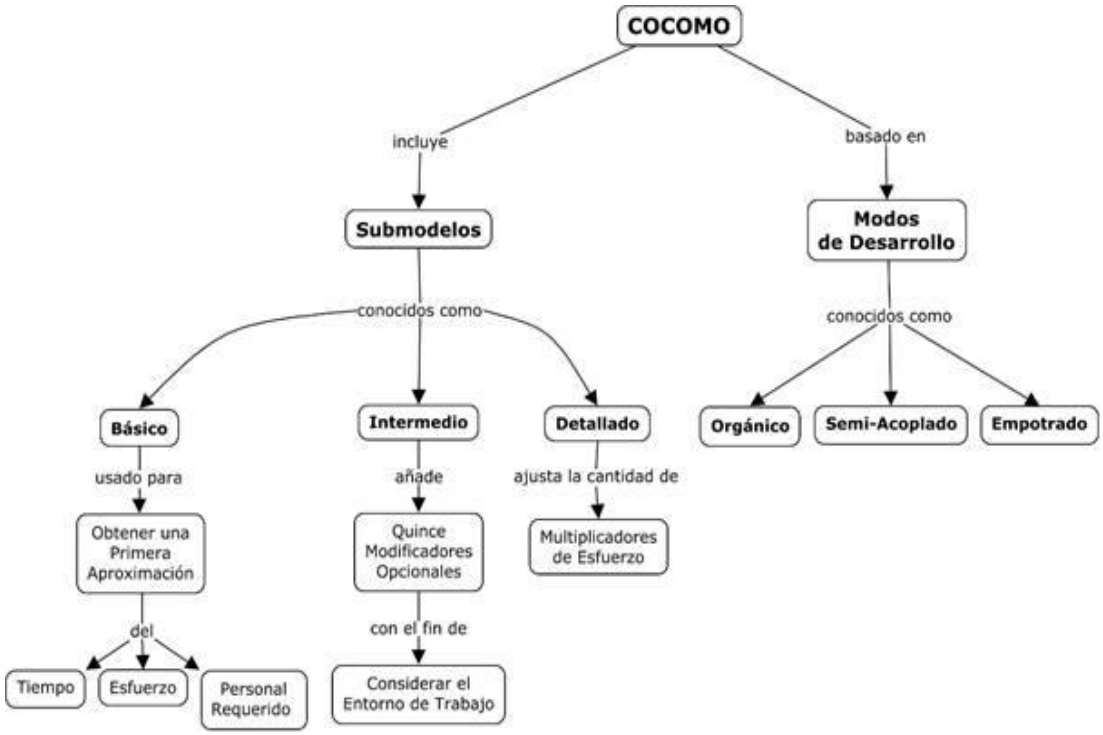


Figura 2.16 Conceptualización básica de COCOMO
Fuente: Moreno, Toledo, Lopez, & Cruz, 2020.

Según la fuente de Aparicio, (2020) describe los modelos de la siguiente manera:

- **El modelo básico** estima el coste del proyecto –pequeño o mediano- en función de número de líneas de código estimadas. En este modelo, el algoritmo COCOMO establece varios criterios de desarrollo, dependiendo el nivel de dificultar no del nivel de experiencia de los desarrolladores –que se supone- sino de posibles dificultades que se pueden encontrar en el desarrollo o limitaciones del hardware usado en el desarrollo del software.

- **El modelo intermedio** se utiliza para estimaciones más complejas. Éste incluye 15 atributos dentro de 4 categorías del software para determinar el coste del proyecto.

- Atributos del producto: garantía de funcionamiento requerida para creación del software, tamaño de la BBDD, etc.
- Atributos del ordenador usado: capacidad de almacenamiento, rapidez del ordenador, etc.
- Atributos del personal: experiencia en el tipo de software a desarrollar, en el lenguaje usado, etc.
- Atributos del proyecto: software usado para el desarrollo, lenguaje necesario para crear el software, etc.

- **El modelo detallado**, incorpora las características del modelo intermedio y lleva a cabo una evaluación del impacto de los motivantes del coste en cada caso -análisis, diseño, etc.- del proceso de ingeniería del software.

Cada submodelo también se divide en modos que representan el tipo de proyecto, y puede ser:

- **modo orgánico:** un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas decenas de miles (medio).
- **modo semilibre o semiencajado:** corresponde a un esquema intermedio entre el orgánico y el rígido; el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.
- **modo rígido o empotrado:** el proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla (Wikipedia, 2020).

Tabla 2.5
Valores constantes por modo de desarrollo.

Modo de desarrollo	COCOMO Básica	COCOMO Intermedia	b	c	d
Orgánico	2.4	3.2	1.05	2.50	0.38
Semiacoplado	3.0		1.12		0.35
Empotrado	3.6	2.8	1.20		0.32

Fuente: Boehm, 1981.

- **Fórmula para calcular esfuerzo.**

Las ecuaciones que se utilizan en los tres modelos son:

$$E = a \times (Kl)^b \times m(X)$$

$$T = c \times (E)^d$$

$$P = E/T$$

Donde:

- E es el esfuerzo requerido por el proyecto, en persona-mes
- T es el tiempo requerido por el proyecto, en meses
- P es el número de personas requerido por el proyecto
- KLDC es el número de líneas de código en miles.
- a, b, c y d son constantes con valores definidos en una tabla. Según cada modelo.
- Kl es la cantidad de líneas de código, en miles
- m(X) es un multiplicador que depende de 15 atributos.

Tabla 2.6
Valores de los factores de escala.

Multiplicadores de esfuerzo (ME)			Valoración					
			Muy bajo	Bajo	Nominal	Alto	Muy alto	Extremadamente alto
Atributos del producto								
1	RELY	Fiabilidad requerida del software	0.75	0.88	1.00	1.15	1.40	
2	DATA	Tamaño de la base de datos		0.94	1.00	1.08	1.16	
3	CPLX	Complejidad del producto	0.70	0.85	1.00	1.15	1.30	1.65
Atributos de la computadora								
4	TIME	Restricciones del tiempo de ejecución			1.00	1.11	1.30	1.66
5	STOR	Restricciones del almacenamiento principal			1.00	1.06	1.21	1.56
6	VIRT	Inestabilidad de la máquina virtual		0.87	1.00	1.15	1.30	
7	TURN	Tiempo de respuesta del computador		0.87	1.00	1.07	1.15	
Atributos del personal								
8	ACAP	Capacidad del analista	1.46	1.19	1.00	0.86	0.71	
9	AEXP	Experiencia en la aplicación	1.29	1.13	1.00	0.91	0.82	
10	PCAP	Capacidad de los programadores	1.42	1.17	1.00	0.86	0.70	
11	VEXP	Experiencia en S.O. utilizado	1.21	1.10	1.00	0.90		
12	LEXP	Experiencia en el lenguaje de programación	1.14	1.07	1.00	0.95		

Atributos del proyecto							
13	MODP	Uso de prácticas de programación modernas	1.24	1.10	1.00	0.91	0.82
14	TOOL	Uso de herramientas software	1.24	1.10	1.00	0.91	0.83
15	SCED	Restricciones en la duración del proyecto	1.23	1.08	1.00	1.04	1.10

Fuente: Boehm, 1981.

Según la fuente de Wikipedia, (2020) el significado de los atributos es la siguiente:

- De software
    - RELY: garantía de funcionamiento requerida al software. Indica las posibles consecuencias para el usuario en el caso que existan defectos en el producto. Va desde la sola inconveniencia de corregir un fallo (*muy bajo*) hasta la posible pérdida de vidas humanas (*extremadamente alto*, software de alta criticidad).
    - DATA: tamaño de la base de datos en relación con el tamaño del programa. El valor del modificador se define por la relación: D/K, donde D corresponde al tamaño de la base de datos en bytes y K es el tamaño del programa en cantidad de líneas de código
    - CPLX: representa la complejidad del producto.
- De hardware
    - TIME: limitaciones en el porcentaje del uso de la CPU.
    - STOR: limitaciones en el porcentaje del uso de la memoria.
    - VIRT: volatilidad de la máquina virtual.
    - TURN: tiempo de respuesta requerido.

- De personal
    - ACAP: calificación de los analistas.
    - AEXP: experiencia del personal en aplicaciones similares.
    - PCAP: calificación de los programadores.
    - VEXP: experiencia del personal en la máquina virtual.
    - LEXP: experiencia en el lenguaje de programación a usar.
- De proyecto
    - MODP: uso de prácticas modernas de programación.
    - TOOL: uso de herramientas de desarrollo de software.
    - SCED: limitaciones en el cumplimiento de la planificación.

2.15 Herramientas

2.15.1 Lenguajes de programación

PHP

PHP (acrónimo recursivo de *PHP: Hypertext Preprocessor*) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga. (php, 2020)

PHP corresponde a las iniciales de personal home page tools (herramientas para páginas iniciales personales). Es un lenguaje de programación tipo script para

entornos web con unas funciones muy semejantes a las de ASP y JSP, utilizado, sobre todo, en servidores Linux para personalizar la información enviada a los usuarios que acceden a un sitio web. Desde un punto de vista técnico, es un lenguaje interpretado de alto nivel, similar en construcciones léxicas y sintácticas a C, C++, Java y Perl, por lo que a quienes ya conozcan estos lenguajes les resultará muy fácil comenzar a escribir código PHP. (Berni & Gil de la Iglesia, 2010)

Python

Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas. (Python, 2020)

Python es un interpretador de instrucciones que permite usar el lenguaje en forma interactiva. Los lenguajes interpretados, a diferencia de los lenguajes compilados, permiten experimentar interactivamente en una ventana y también mediante programas que pueden desarrollarse y probarse a medida que son construidos. Esta interacción facilita el aprendizaje del lenguaje y mejora la productividad. (Rodriguez Ojeda, 2015, p.45)

Características de Python

Alguna de las características mencionadas por Rodriguez Ojeda, (2015) son:

a) Python es un lenguaje interpretado. Se considera sucesor del lenguaje ABC y usa conceptos de otros lenguajes como Modula-3, Lisp, entre otros.

b) Python no obliga a los programadores a adoptar un estilo particular de programación.

c) Se puede instalar en varias plataformas: Windows, Linux, etc. Con menores cambios puede trasladarse entre ellas.

d) Es software libre y de código abierto con licencia GPL (General Public License). Se puede instalar, modificar y distribuir proporcionando el código fuente. Una licencia GPL no ofrece garantía, pero la gran comunidad de usuarios que disponen del código abierto, rápidamente detectan errores.

e) El código escrito en Python es legible, sin marcas para definir bloques como en otros lenguajes. No requiere símbolos de fin de línea. Escribir en este lenguaje es casi como escribir en seudo código en inglés.

Java Script

Según . (Collell, 2013) menciona: “Javascript es el lenguaje de programación utilizado en el desarrollo de aplicaciones web por parte del cliente”. (p.26)

2.15.2 Framework

Flask

Flask es un Microframework de Python que está basado en Werkzeug, Jinja 2 y buenas intenciones. Mediante Flask podemos construir aplicaciones web y servicios Restful con Python de una forma extraordinariamente sencilla. Con pocas líneas podemos llegar a tener un servicio Restful funcionando.

Micro, es un framework pequeño, pero extensible. La idea de Flask es que en una sola página pueda caberte una aplicación web. Por ejemplo, por defecto Flask no trae una capa de abstracción de base de datos. Si bien ya existen múltiples extensiones que agregan dicha capacidad. (Manual Web, 2020)

CodeIgniter

CodeIgniter es un entorno de desarrollo abierto que permite crear webs dinámicas con PHP. Su principal objetivo es ayudar a que los desarrolladores, puedan realizar proyectos mucho más rápido que creando toda la estructura desde cero, proveyendo un rico juego de librerías para tareas comúnmente necesarias, así como una interface simple y estructura lógica para acceder a esas librerías.

CodeIgniter permite enfocarse creativamente en su proyecto minimizando la cantidad de código necesaria para una tarea dada. Este Framework se encuentra desarrollado bajo una licencia open source Apache/BSD-style, así que lo puede usar donde más guste.

CodeIgniter usa el acercamiento Modelo Vista Controlador, que permite una buena separación entre lógica y presentación. Esto es particularmente bueno para proyectos en los cuales los diseñadores están trabajando con sus archivos de plantilla, ya que el código en esos archivos será mínimo. (ECURED, 2020)

Características de CodeIgniter

- Sistema Basado en Modelo-Vista-Controlador
- Compatible con PHP 4
- Extremadamente Liviano
- Clases de base de datos llenas de características con soporte para varias plataformas.
- Soporte de Active Record para Base de Datos
- Formulario y Validación de Datos
- Seguridad y Filtro XSS
- Manejo de Sesión
- Clase de Envío de Email. Soporta Archivos Adjuntos, email de texto/HTML, múltiples protocolos (sendmail, SMTP, and Mail) y más.
- Librería de Manipulación de Imagen (cortar, redimensionar, rotar, etc.). Soporta GD, ImageMagick, y NetPBM
- Clase de Carga (upload) de Archivo
- Clase de FTP
- Localización
- Paginación
- Encriptación de Datos
- Puntos de referencia
- Cacheo de páginas enteras
- Historial de Errores

- Perfilando la Aplicación
- Clase de Calendario
- Clase de Agente del Usuario
- Clase de Codificación Zip
- Clase de Motor de Plantillas

2.15.3 Gestor de base de datos

MariaDB

MariaDB es un Sistema de gestión de bases de datos derivado de MySQL con licencia GPL (General Public License). Es desarrollado por Michael (Monty) Widenius (fundador de MySQL), la fundación MariaDB y la comunidad de desarrolladores de software libre. Introduce dos motores de almacenamiento nuevos, uno llamado Aria -que reemplaza con ventajas a MyISAM- y otro llamado XtraDB -en sustitución de InnoDB. Tiene una alta compatibilidad con MySQL ya que posee las mismas órdenes, interfaces, APIs y bibliotecas, siendo su objetivo poder cambiar un servidor por otro directamente.

Este SGBD surge a raíz de la compra de Sun Microsystems -compañía que había comprado previamente MySQL por parte de Oracle. MariaDB es un fork directo de MySQL que asegura la existencia de una versión de este producto con licencia GPL. Monty decidió crear esta variante porque estaba convencido de que el único interés de Oracle en MySQL era reducir la competencia que MySQL suponía para el mayor vendedor de bases de datos relacionales del mundo, que es Oracle (EcuRed, 2020).

2.15.4 Servidor Web

Apache

Apache es un servidor web multiplataforma, que permite indexación de directorios, uso de sobrenombres con las carpetas, informes configurables sobre errores http, ejecución de programas CGI y que además admite la última versión del protocolo http/1.1.

Una característica importante a señalar es que Apache permite trabajar con servidores virtuales tanto con direcciones IP así como con nombres virtuales. También se podría convertir nuestro servidor en un servidor Proxy. En todo momento, a través de un explorador web, se podría conocer el estado de nuestro servidor, pues tiene registros configurables para guardar dicho estado, así como poder registrar las acciones de los usuarios. (Egea, 2000, p.9)

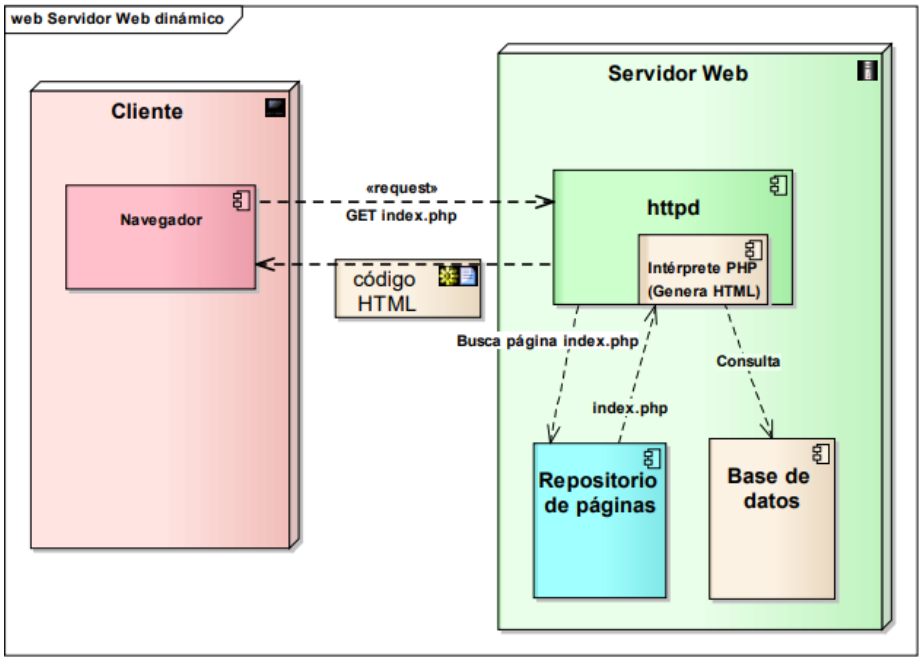


Figura 2.17 Esquema de funcionamiento de un Servidor Web.
Fuente: Pavón, 2013.

CAPÍTULO III

MARCO APLICATIVO

MARCO APLICATIVO

3.1 Introducción

En este capítulo se describe el desarrollo de un sistema de votación electrónica y la aplicación de protocolos blockchain, utilizando aquellos conceptos recolectados en el anterior capítulo, el presente tema de investigación requirió de una metodología, métodos, herramientas las cuales fueron de ayuda para hacer realidad el objetivo planteado, cumpliendo con los requerimientos para su desarrollo.

Lo primero que se realizó es el desarrollo de un sistema de votación electrónica que contenga las funcionalidades básicas y necesarias para la administración de una elección y también para la emisión de votos, luego se procedió al desarrollo y aplicación de los protocolos blockchain.

3.2 Análisis de la Situación Actual

Los procesos electorales dentro de la Carrera Ingeniería de Sistemas se realizan de manera presencial mediante papeletas y urnas de votación físicas, sin embargo, en algunas ocasiones hubo inquietud en el escrutinio de votos al finalizar el día, como también en los resultados finales. Estudiantes como docentes también en ocasiones quedaron inconformes con aquellos resultados denunciando fraude, manipulación de los votos, vulnerabilidad de los resultados.

Si bien un sistema de votación electrónica puede ser de igual manera vulnerable y tener posibilidades en la modificación de los votos logrando corruptibilidad en ellos, se planteó aplicar protocolos blockchain a este sistema para darle mayor seguridad a los votos que se emitan, de esta manera damos inmutabilidad a los datos, así garantizando el proceso de elecciones, dando conformidad a los frentes o partidos como también a los votantes.

3.3 Estructura del Sistema

En la siguiente figura se puede observar la estructura del sistema.

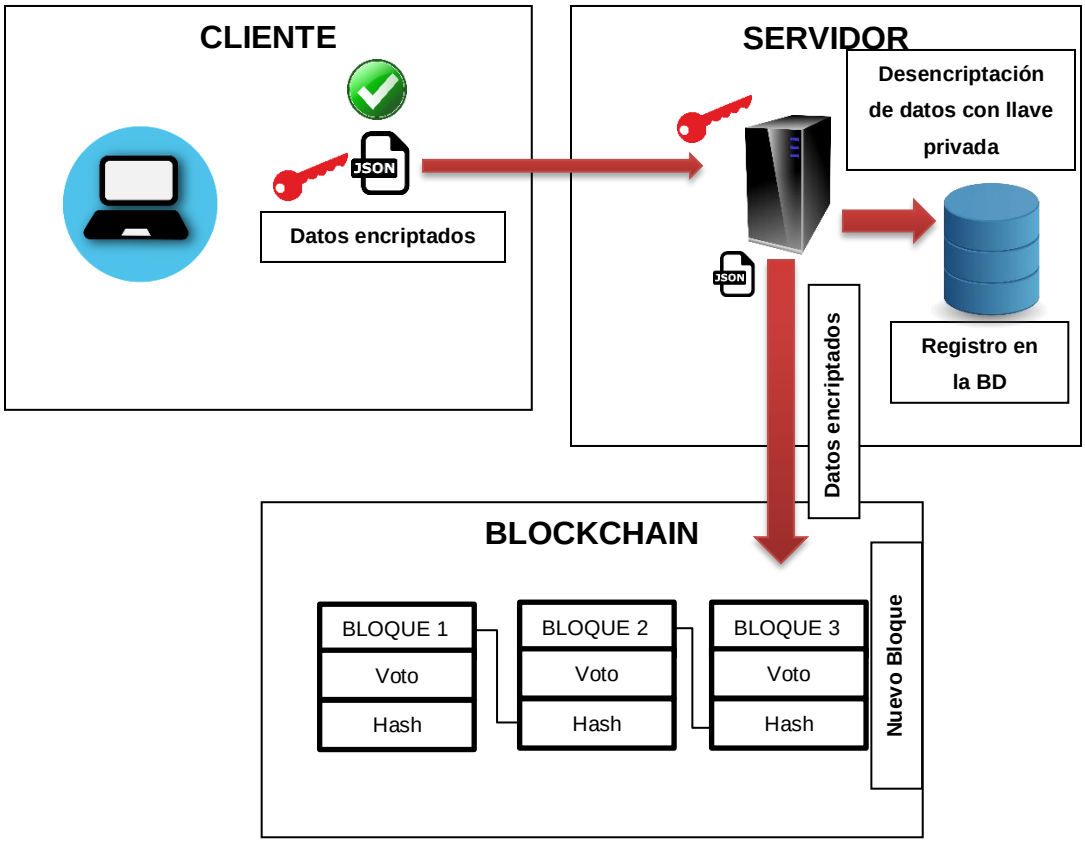


Figura 3.1 Estructura del sistema.
Fuente: Elaboración propia.

La estructura del sistema muestra los componentes para el funcionamiento o comportamiento del sistema de votación electrónica y la aplicación de los protocolos blockchain.

En la figura 3.1 se puede observar de forma clara el trabajo conjunto que se realiza entre el sistema y la cadena de bloques con el fin de brindar seguridad a los datos.

3.4 Aplicación de la Metodología UWE

3.4.1 Análisis de requerimientos

Se obtuvo datos sobre los requerimientos esenciales que debe cumplir este sistema para luego describir los requerimientos funcionales y no funcionales.

- **Requerimientos funcionales**

En la Tabla 3.1 se muestra la descripción de los requerimientos funcionales más relevantes para el desarrollo del prototipo.

Tabla 3.1
Requerimientos funcionales.

Referencia	Función	Categoría
R1	Registro de datos del estudiante y docente	Oculto
R2	Autenticación y validación de usuarios	Oculto
R3	Creación de nuevas elecciones	Evidente
R4	Selección aleatoria de jurados	Oculto
R5	Habilitación de candidatos	Evidente
R6	Registro de terminales	Evidente
R7	Emisión de voto	Evidente
R8	Crear y añadir de bloques a la blockchain	Oculto
R9	Visualización de resultados finales	Evidente
R10	Emitir reportes	Evidente

Fuente: Elaboración propia.

- **Requerimientos no funcionales**

Los requerimientos no funcionales describen las limitaciones del sistema. En la Tabla 3.2 se observa dichos requerimientos.

Tabla 3.2
Requerimientos no funcionales.

Referencia	Función
RNF1	El Sistema realizará una elección a la vez
RNF2	Los datos solo serán modificados por el administrador del sistema
RNF3	Los jurados deben aproximarse al Administrador del sistema para obtener los credenciales de acceso al sistema
RNF4	Las elección se iniciará una vez que el administrador del sistema registre los requisitos para realizar una elección

Fuente: Elaboración propia.

- **Descripción de actores**

Se consideró identificar a los actores quienes serán los que tendrán funciones de gran importancia en el sistema.

Los actores que se tomaron para la interacción con el sistema se muestran en la Tabla 3.3. los cuales juegan un rol diferente.

Tabla 3.3
Descripción de actores

Actor	Descripción
Administrador (Comité electoral)	Usuario con mayor privilegio quien tiene como tarea la creación de elecciones habilitar frentes y candidatos.
Jurado	Usuario que tiene un rol fundamental el cual es iniciar y cerrar la votación en la mesa, además de habilitar a los votantes para que accedan al sistema.
Votante	Usuario que emite su voto.

Fuente: Elaboración propia.

- **Diagramas de caso de uso**

Para describir mejor el funcionamiento del sistema de votación electrónica basada en la tecnología blockchain se realizó diagramas de casos de uso. Se tomó tres situaciones: Pre – votación, Votación, Post – votación.

Caso de uso Pre - votación

Aquí se describe las acciones que se realizan antes del día de votación, cuyas acciones son esenciales para realizar una elección como se observa en la siguiente figura.

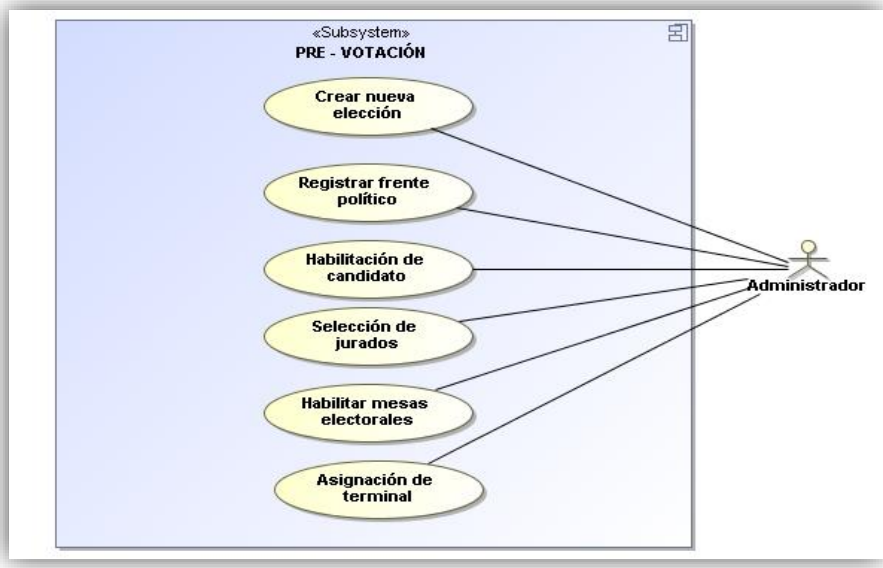


Figura 3.2 Caso de uso pre – votación.
Fuente: Elaboración propia.

En la siguiente tabla se puede observar la especificación del diagrama de caso de uso Pre – votación.

Tabla 3.4
Descripción de caso de uso pre – votación.

Caso de uso	Pre – votación	
Actor	Administrador	
Descripción	El sistema permite al Administrador realizar las acciones necesarias para la realización de un proceso electoral.	
Pre condición	Acceso de internet	
Secuencia	Paso	Acción
	1	Ingresar al sistema con usuario y contraseña
	2	El administrador crea una nueva elección
	3	Registra nuevo frente o partido.
	4	Habilitación de 1 candidato por frente
	5	Selección de jurados y habilitación de mesas
	6	Asignación de equipos de cómputo a las mesas electorales
Post condición	El sistema habilita la elección para ser iniciada	

Fuente: Elaboración propia.

Caso de uso Votación

En este caso de uso de describen las acciones que se realizan en el día de la votación, desde el inicio de la votación hasta el cierre de votación.

Esta fase es una de las más importantes, porque se realiza la emisión de votos los cuales viajan del cliente al servidor, además de que se aplican los protocolos blockchain.

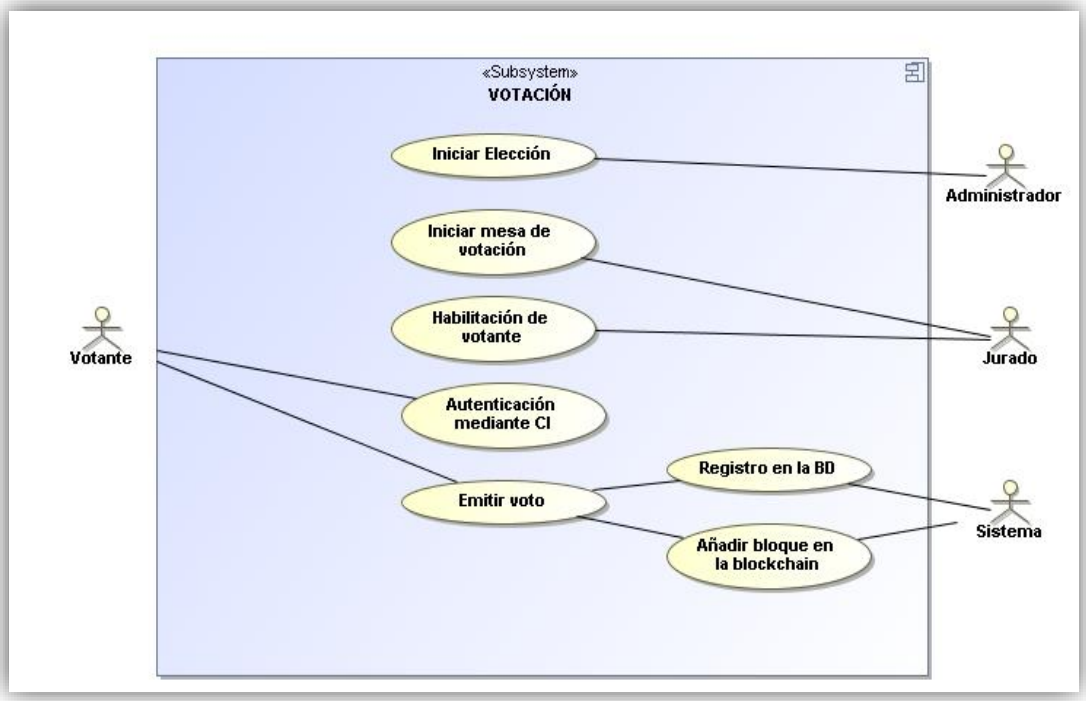


Figura 3.3 Caso de uso Votación.
Fuente: Elaboración propia.

En la siguiente tabla se especifica el diagrama de caso de uso Votación, donde existen cuatro actores que intervienen en el sistema, cada uno con diferentes tareas.

Tabla 3.5
Descripción de caso de uso votación.

Caso de uso	Votación			
Actor	Administrador / Sistema / Jurado / Votante			
Descripción	Se describen las acciones que se realizan en las 8 horas de votación			
Pre condición	El Administrador del sistema debe iniciar la elección			
Secuencia	Paso	Jurado	Votante	Sistema
	1	Inicia sesión y habilita mesa de votación	Ingresa a la sala de votación y digita el número de CI	El sistema abre sesión y visualiza a los candidatos
	2		Presionar sobre la opción preferida para emitir voto	Registra el voto dentro la BD y crea nuevo bloque
	3	Una vez cumplida las 8 horas debe cerrar la votación en la mesa		
Post condición	Conclusión de la elección			

Fuente: Elaboración propia.

Caso de uso Post – votación

En este caso de uso se describen aquellas acciones que se realizan una vez concluida las ocho horas de votación o haber finalizado con todos los votantes habilitados.

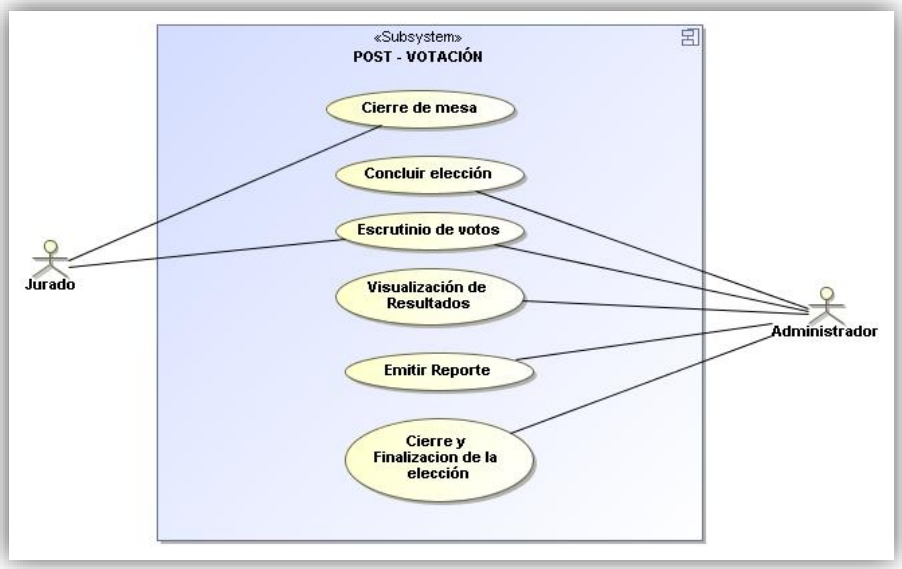


Figura 3.4 Caso de uso post – votación.
Fuente: Elaboración propia.

Tabla 3.6
Descripción de caso de uso post – votación.

Caso de uso	Post – Votación		
Actor	Administrador / Jurado		
Descripción	En el siguiente caso de uso describe las acciones que se realizan después de las 8 horas de votación		
Pre condición	Debe haber cumplido con las 8 horas de votación		
Secuencia	Paso	Jurado	Administrador
	1	Cierre de mesa	Concluye la elección
	2	Observa el escrutinio de votos	Realiza el escrutinio de votos.
	3		Realiza la visualización de resultados en gráficos.
	4		Emite reporte final.
	5		Cierre y finalización de elección.
Post condición	El sistema restringe el acceso a jurados y votantes.		

Fuente: Elaboración propia.

3.4.2 Diagrama de clase

El diagrama de clases nos da una idea clara de la relación de los componentes del sistema, además hace referencia a la estructura de la base de datos.

Cada clase contiene atributos y métodos que se vio conveniente para el modelado del sistema.

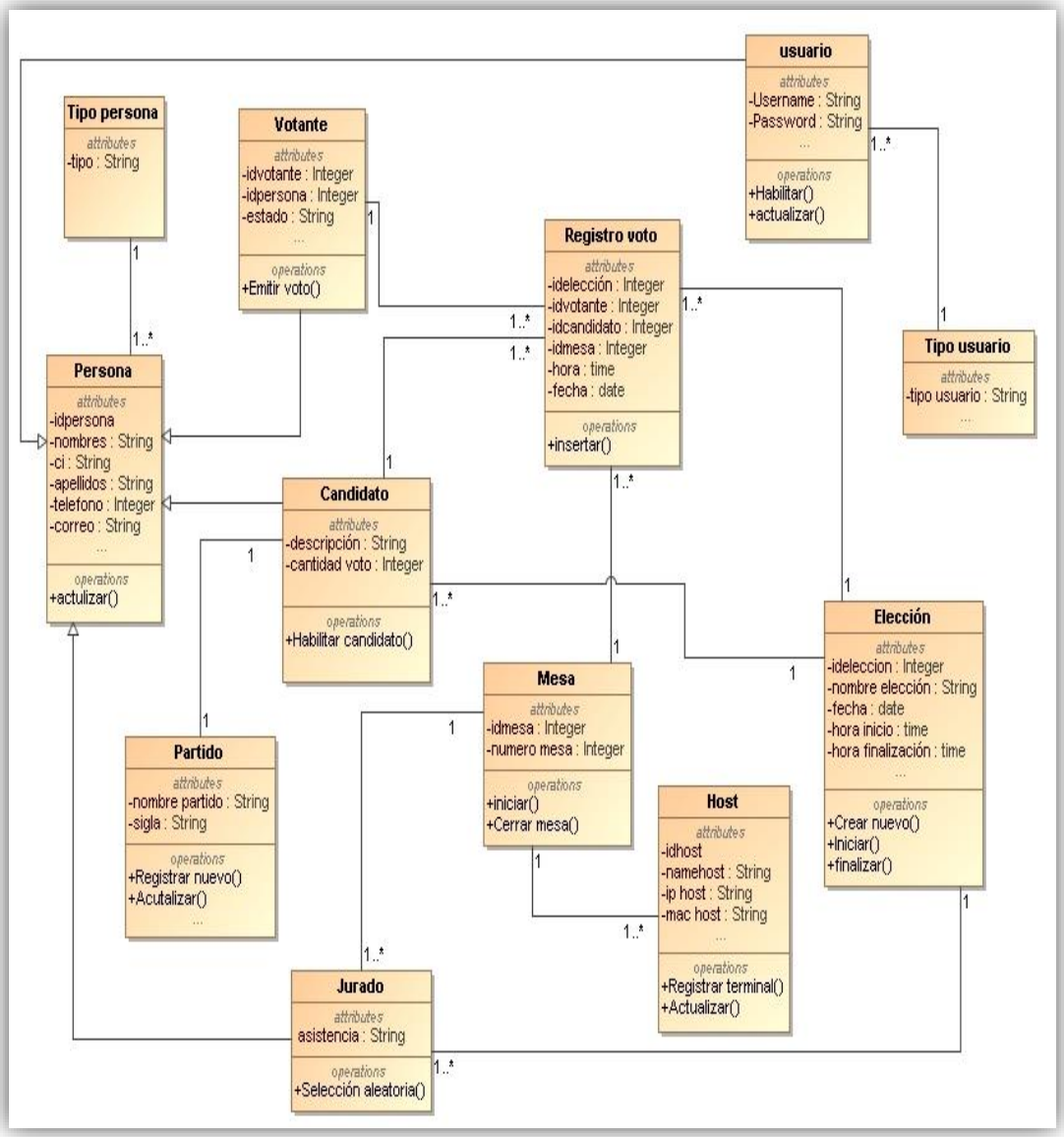


Figura 3.5 Diagrama de clases.
Fuente: Elaboración propia.

3.4.3 Diagramas de navegación

A continuación, se muestran los diagramas de navegación del sistema donde se puede ver la forma de navegar por el sistema.

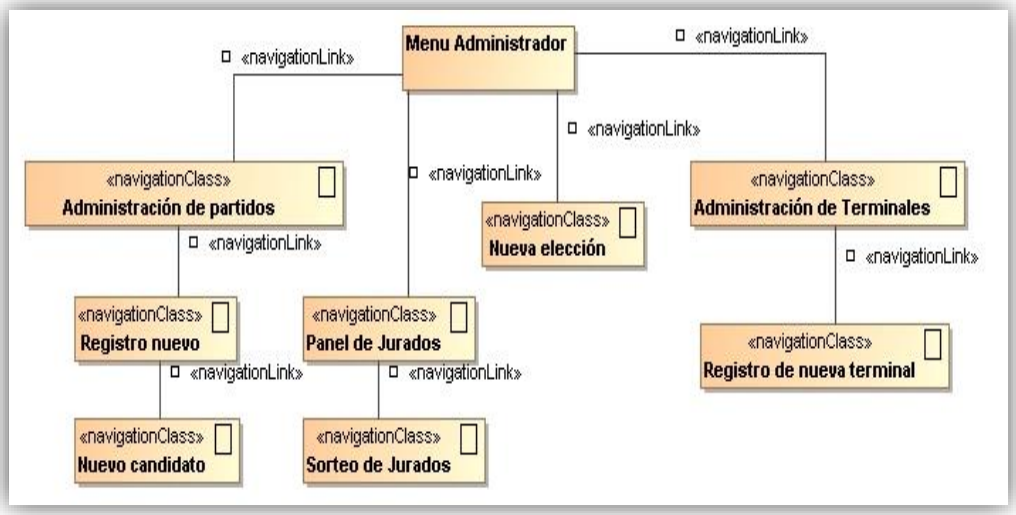


Figura 3.6 Diagrama de navegación pre – votación.
Fuente: Elaboración propia.

En la Figura 3.7. se puede observar la navegación en el caso de la votación.

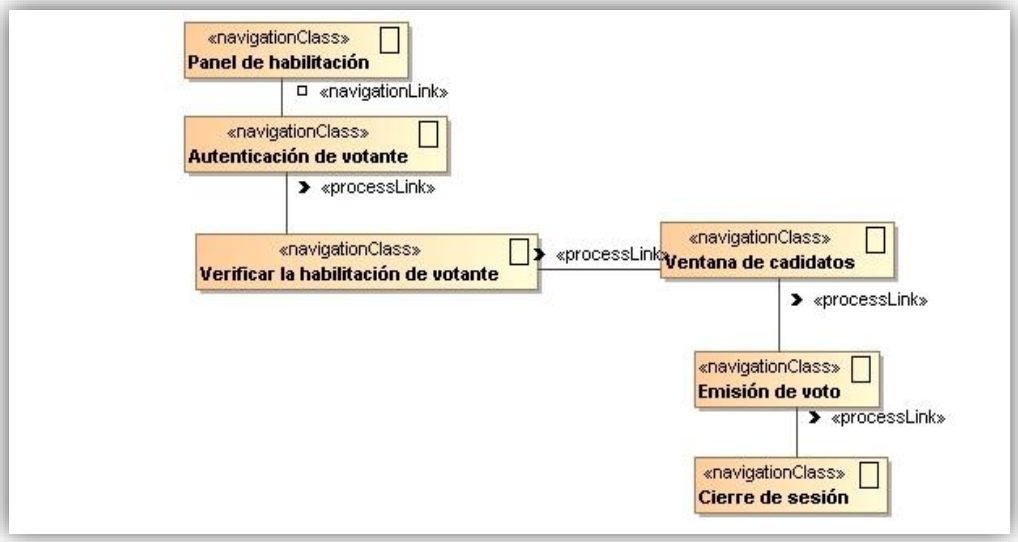


Figura 3.7 Diagrama de navegación votación.
Fuente: Elaboración propia.

En la siguiente figura se muestra la manera que se conectan las paginas para el caso Post – votación.

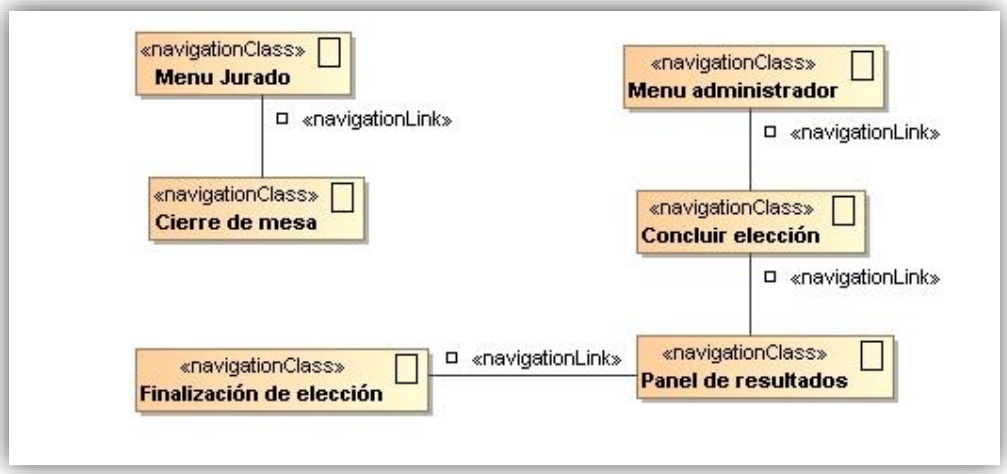


Figura 3.8 Diagrama de navegación post – votación.
Fuente: Elaboración propia.

3.4.4 Diagramas de actividades

Los siguientes diagramas de actividades modelan el comportamiento del sistema en sus tres etapas.

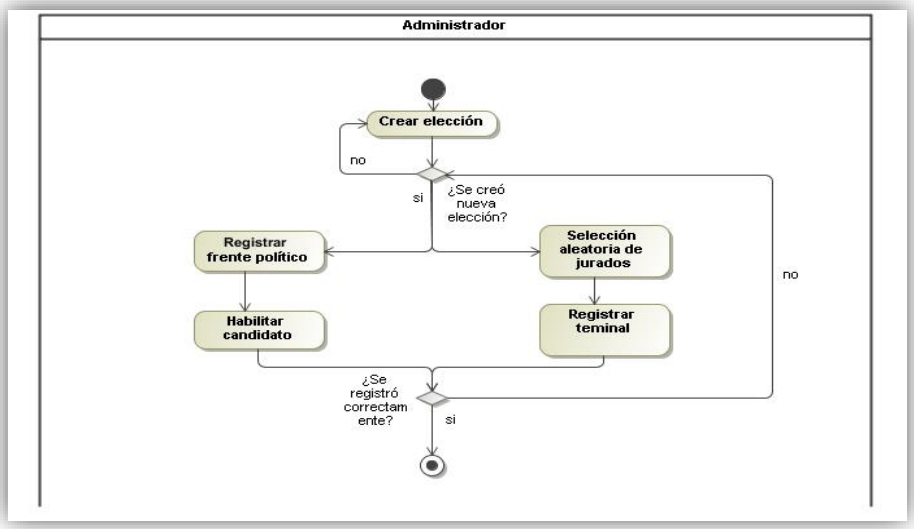


Figura 3.9 Diagrama de actividades pre – votación.
Fuente: Elaboración propia.

En la siguiente figura se muestra el flujo de actividades que con lleva la emisión de votos.

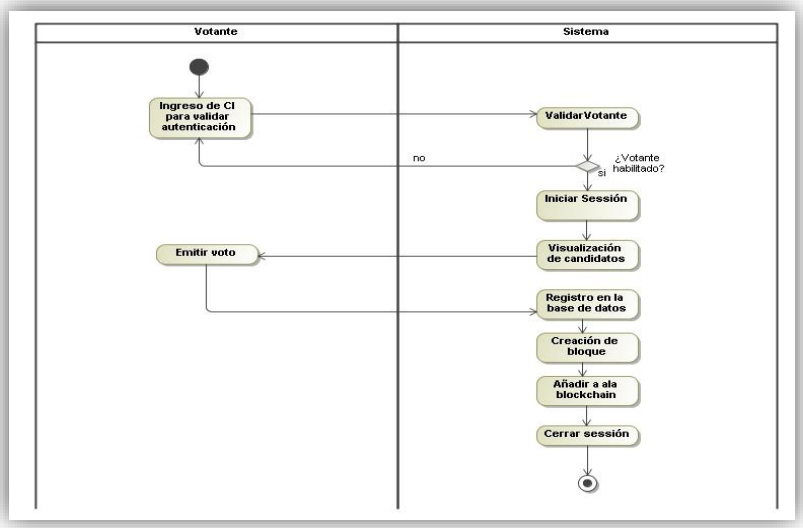


Figura 3.10 Diagrama de actividades votación.
Fuente: Elaboración propia.

En el siguiente diagrama de actividad se puede ver el flujo de actividades para el conteo y emisión de resultados finales de un proceso de elecciones.

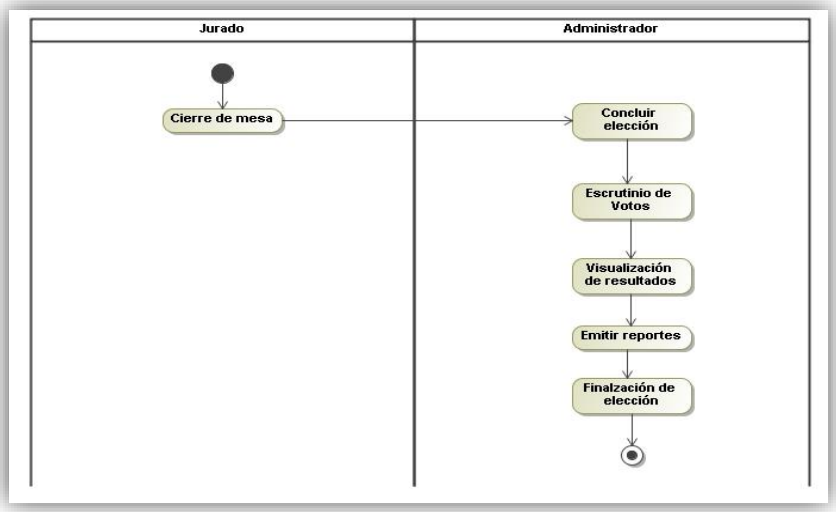


Figura 3.11 Diagrama de actividades post – votación.
Fuente: Elaboración propia.

3.4.5 Diagrama de presentación

Los diagramas de presentación muestran el diseño de un sistema, en la siguiente figura se puede observar el modelo de presentación para el tipo de usuario Administrador.

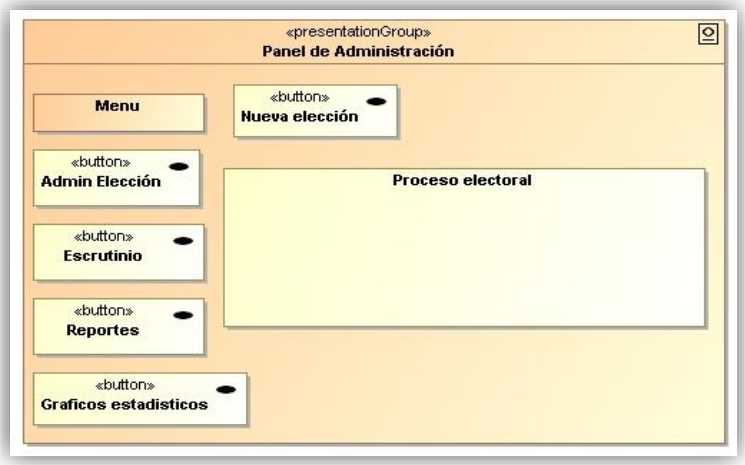


Figura 3.12 Diagrama de presentación panel de administración
Fuente: Elaboración propia.

A continuación, se puede observar el modelo de presentación para los jurados electorales.

Figura 3.13 Diagrama de presentación Panel de Jurado.
Fuente: Elaboración propia.

Los votantes podrán tener una vista en particular que consiste en el panel de sufragio, como se muestra en la figura 3.14

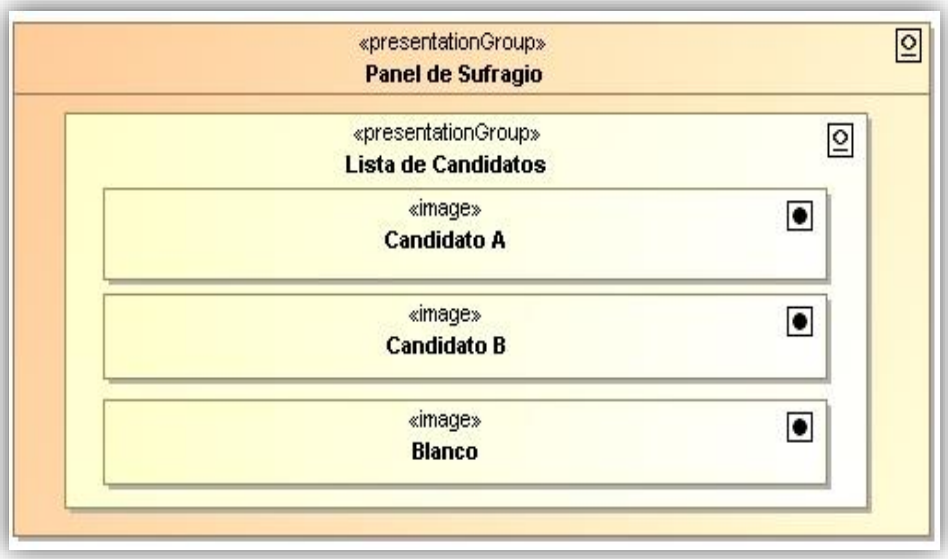


Figura 3.14 Diagrama de presentación panel de sufragio.
Fuente: Elaboración propia.

3.5 Implementación del Sistema

Se desarrolló un prototipo con las funcionalidades de un sistema de votación electrónica, a continuación, se describe la implementación de manera detallada:

a) Autenticación de usuario

El sistema cuenta con un Login para el acceso al sistema donde se solicitará credenciales como ser el usuario y contraseña. Mediante la autenticación, el sistema tiene la facultad de re direccionar a las vistas correspondientes según el tipo de usuario.

Figura 3.15 Autenticación de usuario
Fuente: Elaboración propia.

b) Panel de Administración

El Administrador del sistema (Comité electoral) obtendra el mayor privilegio, el control total del sistema. Se puede observar en la figura N° 3.16 el menu de opciones que tiene el administrador.

Figura 3.16 Vista administrador.
Fuente: Elaboración propia.

c) Creación de elecciones

El prototipo tiene la opción de crear una nueva elección mediante un formulario el cual debe ser llenado por el Administrador del sistema, los datos que requerirá el sistema son el nombre de la elección, observación (opcional), participantes los cuales serán habilitados para la elección y por último la programación de la fecha para la elección.

Figura 3.17 Formulario de Creación de elecciones nuevas.
Fuente: Elaboración propia.

d) Administración de Frentes políticos

El sistema cuenta con una opción para la administración de los partidos.

En la siguiente figura se puede observar el formulario para el registro de un nuevo frente o partido político.

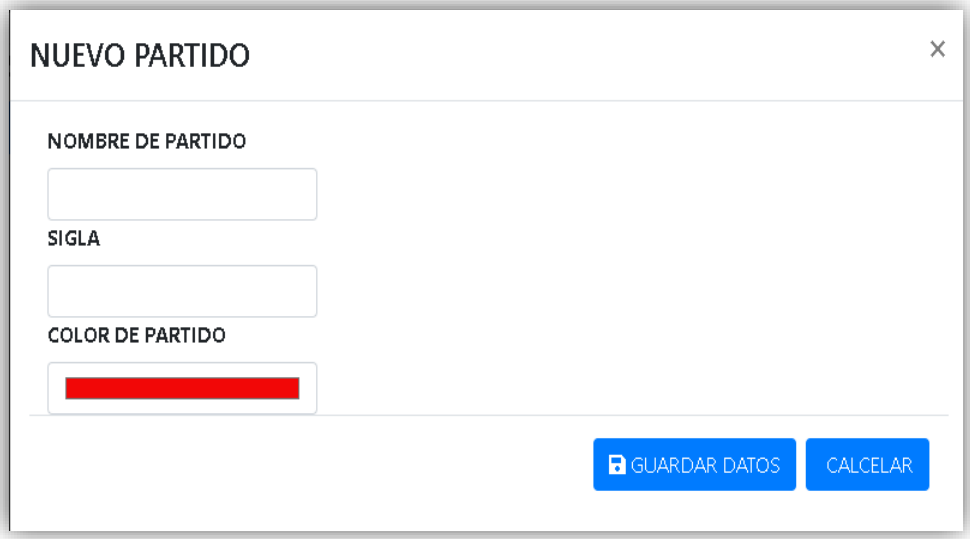


Figura 3.18 Registro de frente político.
Fuente: Elaboración propia.

En caso de que el partido o frente haya participado en anteriores elecciones el sistema ya contara con los datos del partido.

Figura 3.19 Listado de frentes políticos.
Fuente: Elaboración propia.

e) Administración de candidatos

La base de datos del sistema contará con el registro de estudiantes y docentes habilitados. En la figura 3.20 se muestra el listado para la habilitación de candidatos.

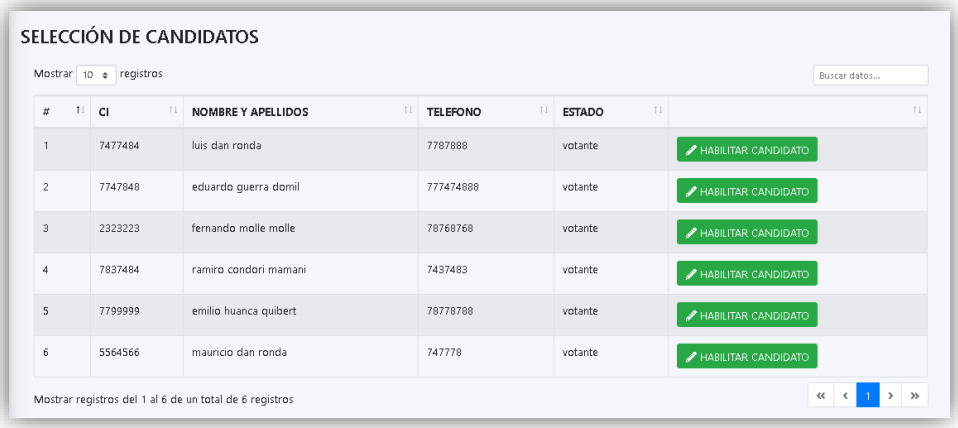


Figura 3.20 Lista de habilitados para selección de candidatos.
Fuente: Elaboración propia.

Mediante un formulario se completará la habilitación de los candidatos, los datos que el formulario requerirá es el partido al que representa y también permitirá adjuntar una foto del candidato.

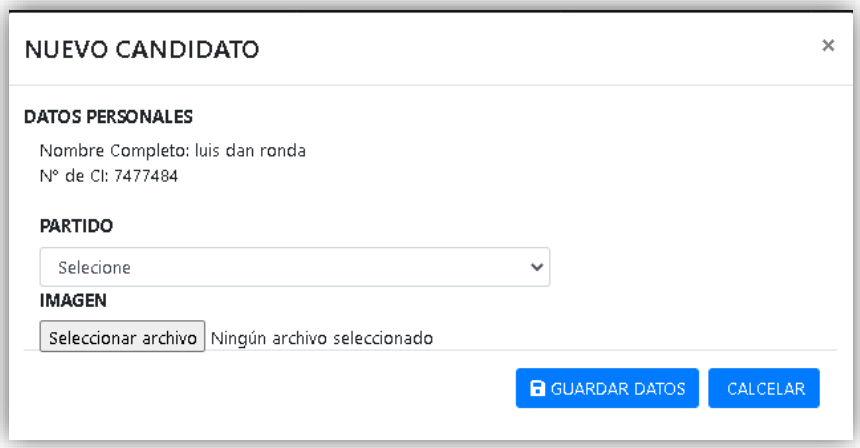


Figura 3.21 Formulario de habilitación de candidatos.
Fuente: Elaboración propia.

En la siguiente figura 3.22 se muestra el listado de candidatos que fueron habilitados.

Figura 3.22 Vista de Candidatos Habilitados.
Fuente: Elaboración propia.

f) Selección de Jurados y mesas electorales

La selección de jurados se realiza aleatoriamente por el sistema mediante una consulta SQL utilizando la función RAND (). Para realizar esta operación se debe llenar el formulario que se observa en la figura 3.23. donde se pide dos datos obligatorios, número de mesas y el número de jurados por mesa.

Figura 3.23 Formulario de Selección de jurados y creación de mesas.
Fuente: Elaboración propia.

g) Asignación de Terminales

Mediante un formulario se realiza la asignación de equipos computacionales para las diferentes mesas electorales, en la figura 3.24 se puede observar los datos que se requerirá para dicha tarea.

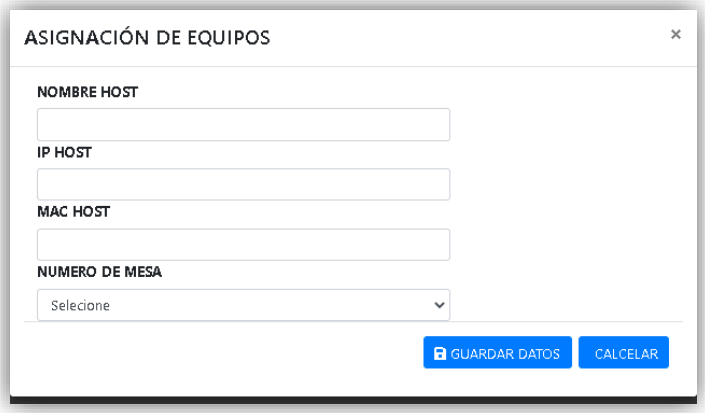


Figura 3.24 Formulario de Asignación de equipos.
Fuente: Elaboración propia.

h) Panel de Jurado

Los jurados obtendrán acceso al sistema donde podrán cumplir la función de iniciar la mesa de sufragio, otra de las funciones en el sistema es el habilitar al votante una vez que el Estudiante o Docente presente su carnet de identidad.

Figura 3.25 Vista Jurado para habilitación de votantes.
Fuente: Elaboración propia.

De forma que, al habilitar a un votante, el votante tendrá acceso al sistema para luego emitir su voto.

i) Ventana de Sufragio

Para evitar complejidad en el sufragio se desarrolló una vista que contiene el listado y la descripción de los candidatos y la opción Blanco. Para realizar la emisión de votos el votante solo debe presionar sobre el botón Votar que se encuentra al lado derecho de la imagen de los candidatos como se observa en la siguiente figura.

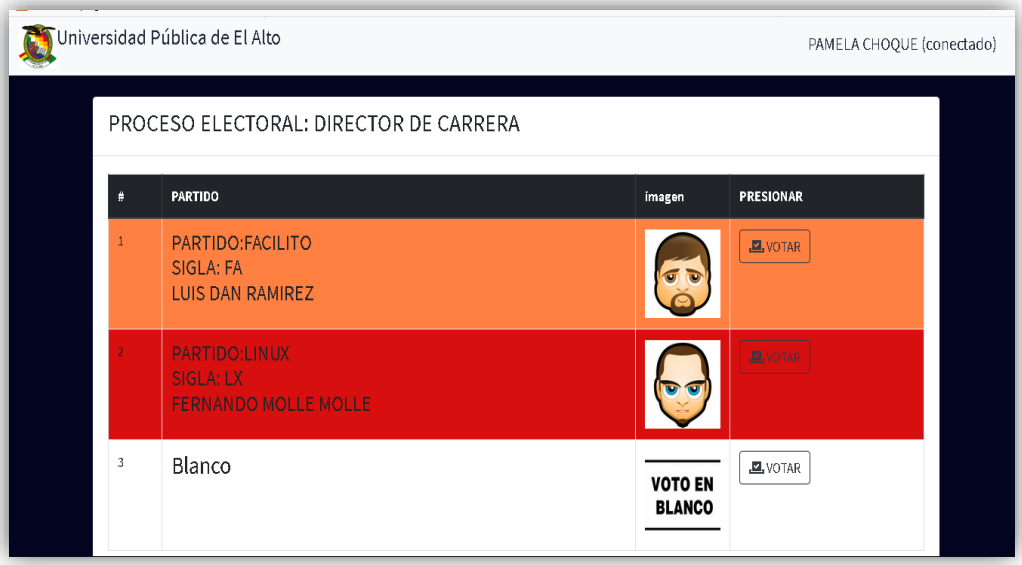


Figura 3.26 Ventana de Sufragio.
Fuente: Elaboración propia.

j) Escrutinio y conteo de votos

Para la realización del proceso de escrutinio y conteo de votos, el sistema ejecuta una consulta para recorrer cada registro de voto en la base de datos, posterior el sistema visualiza los resultados clasificándolos por mesa de sufragio.

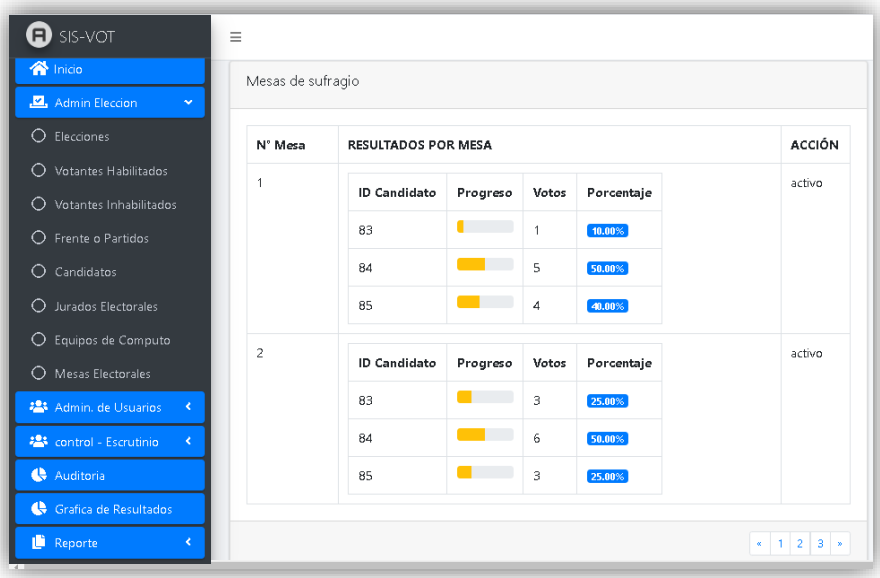


Figura 3.27 Visualización de conteo de votos por mesa.
Fuente: Elaboración propia.

k) Resultados de elección

Mediante un gráfico estadístico se muestra los resultados finales por cada candidato como se observa en la siguiente figura.

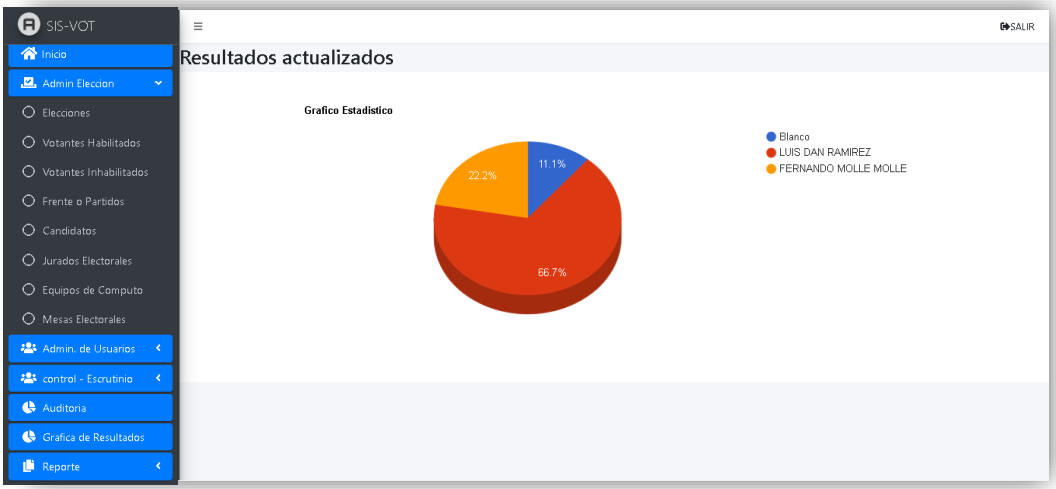


Figura 3.28 Grafico de Resultado General.
Fuente: Elaboración propia.

3.6 Aplicación de la Blockchain

En el anterior punto se pudo implementar un sistema de votación electrónica con las funcionalidades esenciales, sin embargo, para poder llegar a cumplir con el presente tema de investigación, en este punto se realiza la aplicación de los protocolos blockchain al sistema ya desarrollado.

Si bien el sistema de votación electrónica fue desarrollado en el lenguaje de programación PHP, la blockchain fue escrita en el lenguaje de programación Python, el cual funcionará de manera independiente gracias al servicio web que ofrece Flask.

3.6.1 Funciones en Python

En este punto se describen las funciones que hacen posible que la blockchain cumpla con las bases teóricas, además de que trabaje conjuntamente con el sistema de votación electrónica con el fin de resguardar los votos emitidos y brindar mayor seguridad.

- **Nuevo voto**

Para la creación de un nuevo voto que contendrá seis datos importantes (elección, votante, candidato, mesa, fecha y hora) se utilizó la siguiente función:

```
  # Genera un array con los datos para ser agregada en el bloque
def nuevo_voto(self,eleccion,votante,candidato,mesa,fecha,hora):

      self.current_votos.append({
          'eleccion': eleccion,
          'votante': votante,
          'candidato': candidato,
          'mesa': mesa,
          'fecha': fecha,
          'hora': hora,
      })

      return self.last_bloque['index'] + 1
```

➢ **Creación de nuevo bloque**

La función que realiza la creación de un nuevo bloque que contendrá los datos del voto emitido es la siguiente:

```
   # Genera un nuevo bloque
    def nuevo_bloque(self, proof, previous_hash):

        bloque = {
            'index': len(self.chain) + 1,
            'timestamp': time(),
            'votos': self.current_votos,
            'proof': proof,
            'previous_hash': previous_hash or self.hash(self.chain[-1]),
        }
        self.current_votos = []

        self.chain.append(bloque)
        return bloque
```

➢ **Generar Hash**

Un bloque contiene un Hash único, para generar este hash se aplicó la función Sha – 256, la función en Python que realiza esta tarea es la siguiente:

```
# Genera un hash de un bloque
    def hash(bloque):
        bloque_string = json.dumps(bloque, sort_keys=True).encode()
        return hashlib.sha256(bloque_string).hexdigest()
```

3.6.2 Peticiones mediante Rutas Flask

➢ **Registrar nuevo voto**

Para el registro de votos en la cadena de bloques se utilizó las rutas de Flask. Se realizó peticiones HTTP, se usó el método POST que se ejecuta desde el sistema de votación electrónica mediante la función cURL de PHP.

Por medio de una petición POST se envió datos cifrados en formato JSON que contiene seis datos importantes (elección, votante, opción elegida, mesa, fecha y hora), esta petición se la realiza a la siguiente ruta:

```
# Ruta para registrar nuevo voto
@app.route('/votos/nuevo', methods=['POST'])
def nuevo_voto():
    values = request.get_json()
    required = ['eleccion','votante', 'candidato', 'mesa', 'd', 'm',
'fecha', 'hora']
    if not all(k in values for k in required):
        return 'Valores faltantes', 400

    # Crear una nuevo voto
    index = blockchain.nuevo_voto(values['eleccion'],
      values['votante'],      values['candidato'], values['mesa'],
      values['d'], values['m'],
      values['fecha'], values['hora'])

    response = {'message': f'El voto será añadida al bloque {index}'}
    return jsonify(response), 201
```

- **Cadena de bloques**

Para poder observar y obtener la cadena de bloques completa se realiza una petición GET a la siguiente ruta:

```
@app.route('/chain', methods=['GET'])
def full_chain():
    response = {
        'chain': blockchain.chain,
        'length': len(blockchain.chain),
    }
    return jsonify(response), 200
```

3.6.3 Visualización de bloque

Los bloques contienen la información de manera cifrada, con el fin dar anonimato, además de mantener el voto de manera privada.

En la siguiente figura se puede observar un bloque creado con los datos de un voto emitido y que se encuentran totalmente cifrados.

Figura 3.29 Visualización de bloque.
Fuente: Elaboración propia.

Tabla 3.7
Descripción de Bloque

Componente	Descripción
Hash	Identificador de bloque
Index	Numeración del bloque
Previous_hash	Hash del anterior bloque
Proof	Prueba de trabajo que confirma los votos que ingresan
Timestamp	Fecha y hora de la creación del bloque
Votos	Contiene los datos importantes de un voto emitido

Fuente: Elaboración propia.

La función que cumplirá la blockchain es resguardar los datos de manera independiente y totalmente encriptadas con el fin de dar fiabilidad a los votos.

Para una auditoria de los votos, el sistema cuenta con una opción para realizar la desencriptación de los votos y verificar que coincidan con los votos del sistema de votación electrónica.

3.7 Seguridad

3.7.1 Cifrado de votos

Para dar mayor seguridad al proceso de votación, se propuso cifrar los votos en el lado del cliente para luego ser enviadas al servidor. Este proceso se realiza con el fin de que el tráfico de datos en la red sea anónimo y cumplir con la característica de la blockchain.

Para garantizar el voto que emite el votante, se aplicó la criptografía asimétrica o de clave pública RSA que consiste en la encriptación con una clave pública y luego ser desencriptada con una clave privada.

Tabla 3.8
Encriptación de votos en el Front-end.

VOTO	ALGORITMO	DATO ENCRIPTADO
Candidato A		38210161 10795576 30426803 20078069
hora y fecha		38210161 17350959 49997688 25353527
Elección	RSA	50807162 24177383 74053215 23793405
Mesa		50807162 14177383 64053251 33793405
Votante		43447162 64166758 66043291 13724458

Fuente: (Elaboración propia).

Para comprobar si realmente cumple con esta característica, se realizó el control del tráfico de datos en la red, gracias al software Wireshark que nos ayuda a poder observar el tráfico de datos, se pudo evidenciar que los votos enviados del cliente al servidor viajan cifrados.

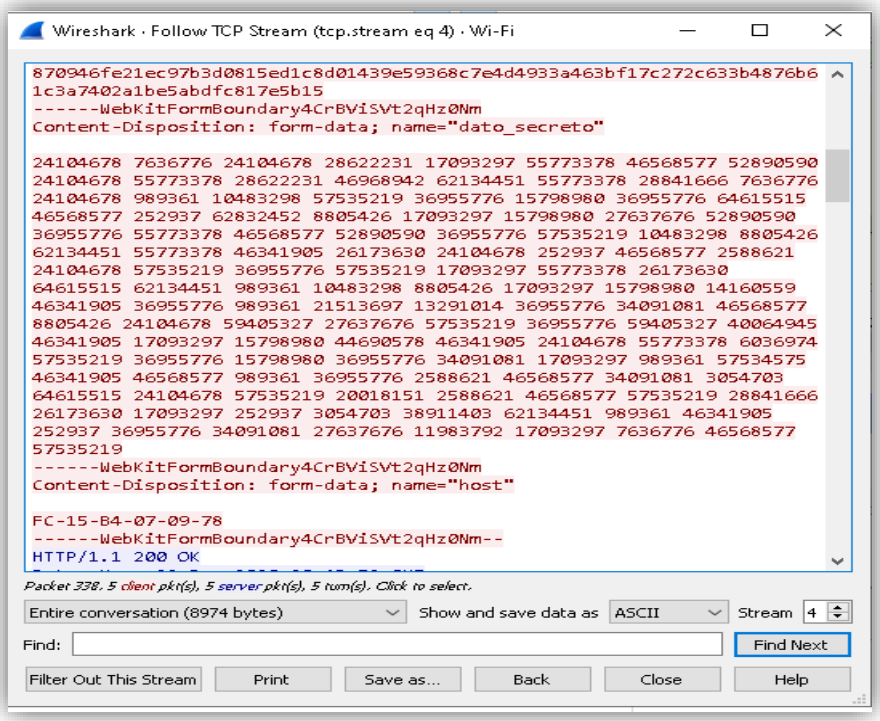


Figura 3.30 Tráfico de datos del cliente al servidor.
Fuente: Elaboración propia.

Como se pudo observar en la anterior figura, los datos que viajan del cliente al servidor están cifrados, mediante este método podemos evitar que los atacantes puedan ver los votos.

3.7.2 Inmutabilidad de votos

Para darle mayor nivel de seguridad al sistema se aplicó la teoría de la blockchain, donde cada bloque de datos contiene el voto emitido y se genera un Hash propio del bloque, adicionalmente hereda el Hash del último bloque de la cadena, el cual los mantiene conectado.

En la siguiente tabla se puede observar un ejemplo de 3 bloques que contiene un Hash en la parte superior, en el centro la información y al final un Hash del bloque anterior.

Tabla 3.9
Conexión de bloques mediante un Hash.

Bloque 100	Bloque 101	Bloque 102
Hash bloque 100	**Hash bloque 101**	**Hash bloque 102**
2EBE9F39B3BECAB9 75B286BA7197FFD7 D55129AB5642BA102 B7CFE1ADE55C261	63BE65064967695788 B19F0E80EAD2DDBA C1B5146771D79A499 0AACE25AD9B66	018F65038FACBD0527 6CAECDBBE34E5EA06 6F8F36C17E3489AD85 505F8DDA12C
• Voto • Elección • Mesa • Usuario	• Voto • Elección • Mesa • usuario	• Voto • Elección • Mesa • Usuario
Hash previo	**Hash previo**	**Hash previo**
8698F092008888EFA 83FB7F8724BE73F9F 42B423FAFF4E7FB39 8C84DF15A3815	2EBE9F39B3BECAB9 75B286BA7197FFD7D 55129AB5642BA102B 7CFE1ADE55C261	63BE65064967695788B 19F0E80EAD2DDBAC1 B5146771D79A4990AA CE25AD9B66

Fuente: Elaboración propia.

De esta forma hace inmutable los datos que se guarden en la cadena de bloques.

3.8 Pruebas

3.8.1 Pruebas de caja blanca

Para la aplicación de la prueba de caja blanca se utilizó la métrica de complejidad ciclomática, el cual nos mostrara los casos de prueba.

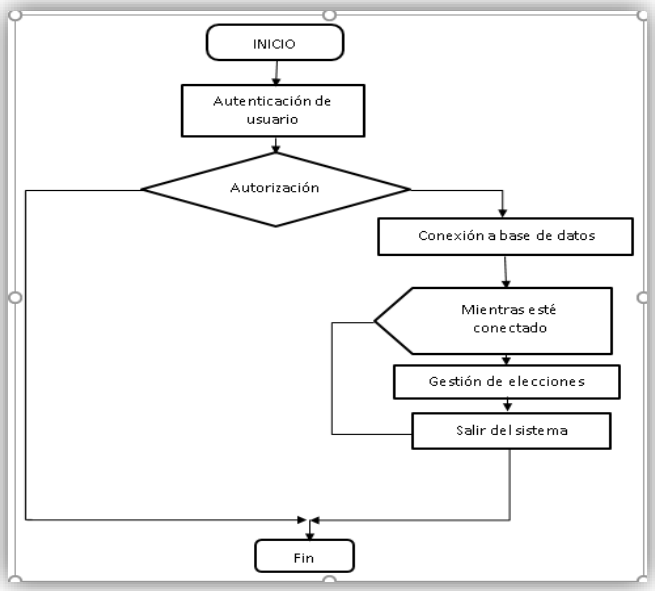


Figura 3.31 Diagrama de uso general.
Fuente: Elaboración propia.

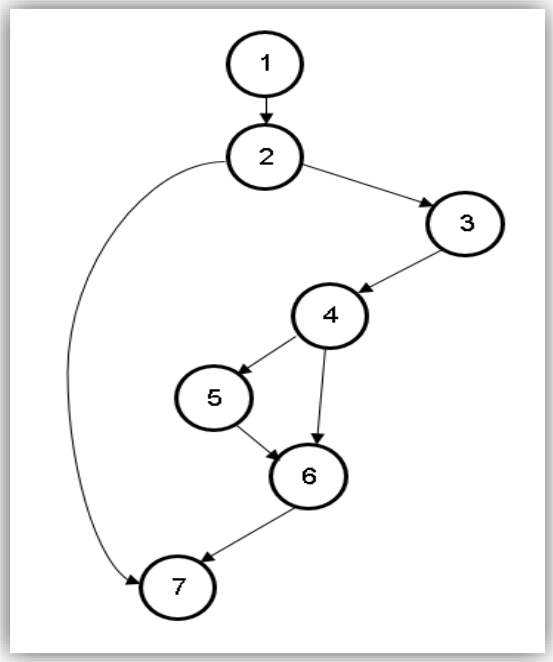


Figura 3.32 Grafo de flujo
Fuente: Elaboración propia.

- **Cálculo de la Complejidad Ciclomática**

Para calcular la complejidad ciclomática se hace uso de la siguiente formula:

$$\boldsymbol{V(G) = A - N + 2}$$

Dónde:

N = Numero de nodos

A = Numero de aristas

Reemplazando los valores tenemos:

$$\boldsymbol{V(G) = 8 - 7 + 2}$$

$$\boldsymbol{V(G) = 3}$$

El valor de complejidad ciclomática es igual a 3, el cual nos indica que son tres casos de pruebas que se deben ejecutarse.

- **Determinación de los casos de prueba**

A continuación, los tres caminos básicos:

Camino 1: 1-2-3-4-5-6-7

Camino 2: 1-2-3-4-6-7

Camino 3: 1-2-7

3.8.2 Pruebas de caja negra

Esta prueba consiste en realizar pruebas de la interfaz gráfica del sistema, a continuación, se muestran las pruebas más relevantes.

- **Pruebas de autenticación**

Cuando el usuario realice el ingreso de sus credenciales para iniciar sesión y estas sean correctas el sistema lo re direcciona a la vista correspondiente según

el tipo de usuario, en caso de que el usuario digite mal el usuario o contraseña el sistema arrojara un mensaje en rojo.

En la figura N° 3.33 se puede observar que el sistema emite un mensaje en caso de que el usuario llegue a utilizar credenciales falsas.

Figura 3.33 Prueba de autenticación.
Fuente: Elaboración propia.

- **Validación de inicio de elección**

Una de las funciones importantes que el administrador del sistema realiza es la inicialización de la elección donde habilita a los jurados para acceder al sistema y puedan realizar sus tareas respectivas.

Para poder realizar el inicio de elección debe cumplir con los requerimientos (Candidatos habilitados, Jurados asignados, mesas electorales y equipos de cómputo asignados), caso contrario el sistema arroja un mensaje de alerta para que complete estos requisitos para iniciar una elección.

Figura 3.34 Validación de inicio de elección.
Fuente: Elaboración propia.

3.8.3 Prueba de acceso de usuarios

Una de las pruebas importantes que se realizo fue el acceso al sistema de los distintos tipos de usuarios.

Se pudo comprobar que el sistema toma el identificador único de cada usuario y valida el tipo de usuario para luego re direccionar a la vista correspondiente, el sistema cuenta con 3 tipos de usuarios como se observa en la siguiente tabla.

Tabla 3.10
Tipos de usuario.

Tipo Usuario	Identificador	Acceso
Administrador(Comité electoral, encargado del sistema)	200	Administración de elecciones
Jurado Electoral	202	Gestión de mesa electoral
Votante	203	Panel de votación

Fuente: Elaboración propia.

3.9 Métrica de Calidad

La medición de un software es de gran importancia con el fin de tener un producto de calidad y estable, para realizar la medición se utilizó la ISO 9126.

3.9.1 Funcionalidad

A continuación, se muestra la ponderación de las características funcionales.

Tabla 3.11
Ponderación de la funcionalidad.

Característica	Ponderación
Adecuación	80%
Exactitud	80%
Interoperabilidad	90%
Seguridad	90%
Cumplimiento funcional	85%
Promedio	85%

Fuente: Elaboración propia.

Por tanto, se deduce que el prototipo tiene una funcionalidad del 85%.

3.9.2 Confiabilidad

Para hallar la confiabilidad se optó por aplicar la siguiente formula:

$$Confiabilidad = 1 - \frac{Número\ de\ errores}{Número\ de\ lineas\ de\ codigo} \times 100$$

Reemplazando los datos que obtuvimos en el punto de Costos para el cálculo de las líneas de código tenemos:

$$Confiabilidad = 1 - \frac{3}{8813.97} \times 100 = 99\%$$

El sistema tiene una confiabilidad del 99%.

3.9.3 Usabilidad

Mediante el siguiente test se logró obtener una ponderación sobre la usabilidad del prototipo desarrollado.

Tabla 3.12
Cuestionario sobre la usabilidad

N°	Pregunta	Valoración min.	Valoración máx.	Promedio
1	¿Es fácil de aprender el manejo de operaciones del sistema?	3	5	4
2	¿Las pantallas le parecen agradables?	4	5	4.5
3	¿La sistema responde rápido a las solicitudes?	2	5	3.5
4	¿Considera que el sistema es una herramienta útil?	3	5	4
5	¿El sistema llegó a cumplir son todas sus expectativas?	3	5	4

Fuente: Elaboración propia.

Con los datos obtenidos en la anterior tabla y reemplazándolas en la fórmula para hallar la usabilidad se tiene:

$$Usabilidad = \frac{\sum x_i}{n} \times \frac{100}{n}$$

$$Usabilidad = \frac{20}{5} \times \frac{100}{5} = 80$$

$$Usabilidad = 80\%$$

3.9.4 Eficiencia

Para poder obtener el cálculo de la eficiencia del sistema se consideró ponderar las características esenciales que el sistema desempeña.

Tabla 3.13
Evaluación de desempeño.

Característica de desempeño	Ponderación
Rapidez en Inicios de sesión	4
Proceso rápido en registrar	4
Proceso rápido en la búsqueda de votantes	5
Respuesta rápida a consultas	5
Fluidez en reportes y gráficos	4

Fuente: Elaboración propia.

En base a los datos de la anterior tabla se podría llegar a tener una idea de la eficiencia, para ello utilizamos la siguiente formula:

$$Eficiencia = \frac{\sum x_i}{n} \times \frac{100}{n}$$

$$Eficiencia = \frac{22}{5} \times \frac{100}{5} = 88 = 88\%$$

3.9.5 Mantenibilidad

Para calcular la mantenibilidad se utilizará la siguiente formula:

$$M = \frac{[Mt - (Fa + Fm + Fe)]}{Mt}$$

Donde:

Mt = número de módulos en la versión actual

Fm = número de módulos en la versión actual que han sido modificados

Fa = número de módulos en la versión actual que han sido añadidos

Fe = número de módulos de la versión anterior que se han eliminado en la versión

En la siguiente tabla se muestra los datos requeridos.

Tabla 3.14
Mantenibilidad del sistema

Información	Valor
Mt	4
Fa	0
Fb	0
Fc	0

Fuente: Elaboración propia

Con estos datos se procede al cálculo de la mantenibilidad.

$$M = \frac{[4 - (0 + 0 + 0)]}{4} = 1 \times 100\% = 100\%$$

3.9.6 Resultados

La ISO 9126 indica 6 características que debe evaluarse sin embargo para este prototipo se tomó 5, cabe recalcar que este prototipo es orientado a la Web, esto hace que sea 100% portable, es por eso que no se tomó la característica portabilidad.

Tabla 3.15
Calidad Global.

Parámetros	Porcentaje
Funcionalidad	85%
Confiabilidad	99%
Usabilidad	80%
Eficiencia	88%
Mantenibilidad	100%
Promedio	90.4%

Fuente: Elaboración propia.

Como se puede observar la referencia global de calidad es de 90.4%, esto quiere decir que el trabajo se encuentra en un nivel de aceptación satisfactorio.

3.10 Evaluación de Costos

En este punto se realiza la estimación del costo total del software desarrollado.

3.10.1 Puntos de función

Para realizar la estimación del tamaño del sistema se utilizó la técnica punto de función no ajustado.

Tabla 3.16
Calculo del punto de función no ajustado.

Parámetros de medición	Cuenta	Factor de ponderación	Total
Número de entradas de Usuarios	10	5	50
Número de salidas de usuario	9	5	45
Número de peticiones de Usuario	12	6	72
Número de archivos	13	6	78
Número de interfaces Externas	4	5	20
Cuenta total			265

Fuente: Elaboración propia.

En la siguiente tabla se muestra los 14 factores de ajuste donde se pondera con un puntaje que se encuentre entre 0 y 5.

Tabla 3.17
Calculo de ajuste de complejidad.

Fi	Factor	Valor
1	Mecanismos de recuperación	1
2	Comunicación de datos	3
3	Funciones de proceso distribuido	2
4	Rendimiento	2
5	Configuración usada rigurosamente	1
6	Entrada de datos en línea	3
7	Factibilidad operativa	3
8	Actualización en línea	3
9	Interfaces complejas	2
10	Proceso interno complejo	3
11	Reusabilidad de código	3
12	Fácil instalación	3
13	Instalaciones múltiples	2
14	Facilidad de cambios	3
	$\sum \boldsymbol{Fi}$	34

Fuente: Elaboración propia.

Procedemos a reemplazar los datos a la fórmula de punto de función ajustado:

$$PFA = cuenta\ total\ \times (0.65\ + 0.01\ \times \sum Fi\ )$$

$$PFA = 265 \times (0.65\ + 0.01\ \times\ 34\ )$$

$$PFA = 272.25$$

3.10.2 Aplicación de COCOMO

Para poder calcular las líneas de código, utilizamos el valor del punto de función ajustado, de igual forma utilizaremos el valor de Factor de líneas de código del lenguaje de programación utilizada para el desarrollo.

Tabla 3.18
Factor LCD/PF de lenguajes de programación.

Lenguaje	Nivel	Factor LDC/PF
C	2.5	128
ANSI basic	5	64
Java	6	53
PL/I	4	80
Visual Basic	7	46
ASP	9	36
PHP	11	29
Visual C++	9.5	34

Fuente: Pressman, 2002.

Reemplazamos los datos en la fórmula para calcular las líneas de código:

$$LDC = PFA \times Factor\ LDC/PF$$

$$LDC = 272.25 \times 29 = 7895.25$$

$$KLDC = \frac{7895.25}{1000} = 7.89525$$

Tabla 3.19
Tipos de proyecto de software.

MODO	A	b	c	d
Orgánico	2.4	1.05	2.5	0.38
Semiacoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.2	2.5	0.32

Fuente: elaboración propia en base a COCOMO.

Para obtener el valor de m(X), ponderamos en base a los 15 atributos de la tabla 2.6

$m(X) = 0.88 \times 1 \times 1.5 \times 1 \times 1.06 \times 1.15 \times 1.07 \times 0.86 \times 1 \times 0.86 \times 0.90 \times 1 \times 0.91 \times 0.91 \times 1.04 = 0.76$

Luego se procede aplicar la fórmula para hallar el esfuerzo que se planteó en el anterior capitulo:

$$E = a \times (KLDC)^b \times m(X)$$

$$E = 2.4 \times (7.89525)^{1.05} \times 0.76$$

$$E = 15.96 = 16\ personas/mes$$

Luego calculamos el tiempo de desarrollo

$$T = c \times (E)^{d}$$

$$T = 2.5 \times (15.96)^{0.38}$$

$$T = 7.16 = 7\ meses$$

Número de personas para el desarrollo

$$P = E/T$$

$$P = \frac{15.96}{7.16}$$

$$P = 2.23 = 2\ personas$$

Para el calcular el costo del software se considera el sueldo aproximado de un Ingeniero de Sistemas Junior de 2500 bs mensuales.

Por ultimo aplicamos la fórmula para hallar el costo del software

$$CT = 2500 \times (P \times T)$$

$$CT = 3000 \times (2 \times 7)$$

$$CT = 42000Bs. = 6122.49\ \$us.$$

3.10.3 Costo de elaboración del software

Se considera los otros costos que se requirió para el desarrollo del prototipo, a continuación, se muestran dichos costos.

Tabla 3.20
Costo de elaboración de prototipo.

Detalle	Importe
Análisis y diseño del prototipo	700 bs.
Material de escritorio	100 bs.
Conexión a internet	308 bs
Otros	50 bs.
Total	1158 bs.

Fuente: Elaboración propia.

3.10.4 Costo Total

Para calcular el Costo total se toma en cuenta el costo del software calculado anteriormente y el costo de elaboración.

Tabla 3.21
Costo total del prototipo.

Detalle	Importe
Costo del software	42000 bs.
Costo de elaboración	1158 bs
Total	43158 bs

Fuente: Elaboración propia.

Entonces se llega a la conclusión que el costo total del software es de 43158 Bs.

CAPÍTULO IV

PRUEBA Y RESULTADOS

PRUEBAS Y RESULTADOS

4.1 Introducción

En el presente capítulo se muestran las pruebas realizadas al sistema de votación electrónica como también a la blockchain con el fin de obtener resultados y verificar si se cumplió con los objetivos planteados, además en este capítulo se realiza la prueba de Hipótesis, que es fundamental para el presente trabajo de investigación.

4.2 Pruebas del Sistema

Se realizó una votación para poder ver el comportamiento del sistema, se tomó a 20 estudiantes para que puedan emitir su voto. Se habilitó a 2 candidatos más la opción voto en blanco, en total el votante tuvo 3 opciones para elegir.

4.2.1 Resumen de resultados

En la tabla 4.1 se muestran los resultados que se obtuvieron en la votación.

Tabla 4.1
Resultado general de la votación.

ID Candidato	Cantidad de votos
84	9
85	7
83	4
Total votos	20

Fuente: Elaboración propia.

Detalle de los Votos

A continuación, se observa a detalle los votos.

Tabla 4.2
Resultado especifico de votación.

N° voto	Opción elegida (id candidato)
1	84
2	85
3	85
4	84
5	83
6	84
7	84
8	83
9	85
10	84
11	83
12	85
13	84
14	84
15	85
16	84
17	85
18	85
19	84
20	83

Fuente: Elaboración propia.

4.2.2 Prueba de Inyección SQL al sistema

Se realizó una prueba de inyección SQL para realizar modificaciones en la base de datos con el fin modificar votos y beneficiar a un candidato.

A continuación, se puede observar los votos reales y el listado de votos con algunas modificaciones, aquellos números que se marcan con rojo son los que fueron modificados.

Tabla 4.3
Modificación de datos.

N°	Votos reales	Votos con modificación
1	84	84
2	85	84
3	85	85
4	84	84
5	83	85
6	84	85
7	84	85
8	83	83
9	85	85
10	84	84
11	83	83
12	85	85
13	84	85
14	84	84
15	85	85
16	84	85
17	85	85
18	85	85
19	84	84
20	83	83

Fuente: Elaboración propia.

Se modificó cinco votos quitándole al candidato 84 y beneficiado al candidato 85, esta es una de las debilidades de los sistemas de votación electrónica que son vulnerables a sufrir cambios y llegando a un fraude electoral.

4.2.3 Pruebas de modificación de votos en la blockchain

Cada voto se almacena en un bloque, y en cada bloque se genera un Hash único. La prueba realizada fue la desencriptación del Hash del bloque.

Se utilizó páginas web de desencriptación, en la siguiente figura podrá observar una de las pruebas de desencriptación realizada.

Figura 4.1 Desencriptación de Hash.
Fuente: Elaboración propia.

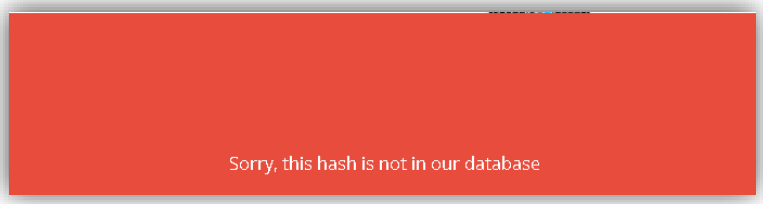


Figura 4.2 Resultado de la desencriptación de Hash..
Fuente Elaboración propia.

Se intentó de la misma forma cambiar los votos que se encuentran en la base de datos de la blockchain, sin embargo, se encontró con una base de datos con todos los datos cifrados, donde no se pudo realizar cambios porque cada voto tenía un cifrado distinto.

De todas formas, se utilizó algunas páginas web de desencriptación para intentar ver el dato y el voto, como se puede observar en las siguientes figuras no se tuvo éxito en el intento.

Para realizar la prueba de descifrado de votos de la blockchain, se tomó estos datos para luego copiarlos en varias páginas web de desencriptación.

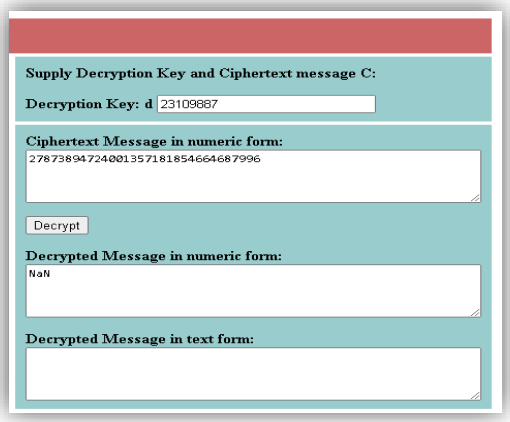


Figura 4.3 Descifrado RSA en cs.drexel.edu.
Fuente: Elaboración propia.

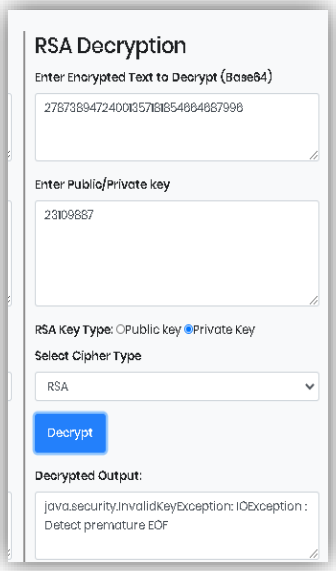


Figura 4.4 Descifrado RSA en Devglan.com.
Fuente: Elaboración propia.

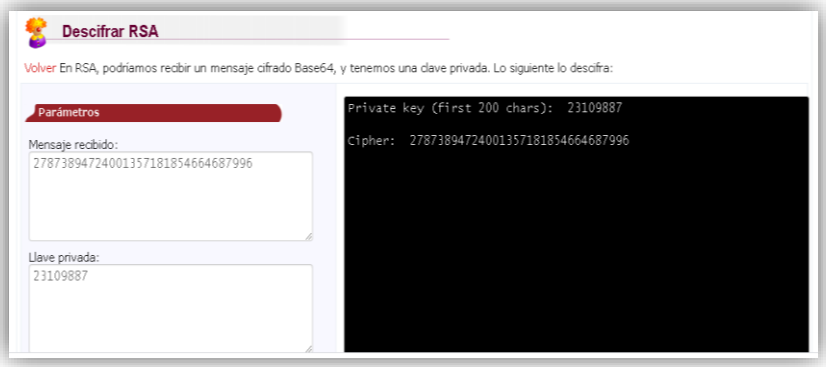


Figura 4.5 Descifrado RSA en Asecuritysite.com.
Fuente: Elaboración propia.

Las pruebas de desencriptación dieron un resultado satisfactorio comprobando que es difícil romper el algoritmo y descifrar el mensaje.

4.3 Resultados

El prototipo ayuda en minimizar el tiempo de las tareas que se realizan para llevar a cabo un proceso de elección, en la siguiente figura se observa la comparación.

Tabla 4.4
Comparación de Tiempos.

Funciones	Proceso de Elecciones Tradicional	Sistema de votación electrónica
Tareas post-elección (Inscripción de candidatos, selección de jurados, impresión de papeletas y otros)	24 - 72 horas	3 - 8 horas
Tareas post – votación (Escrutinio de votos, conteo de votos, emisión de resultados)	1 - 4 horas	5 - 30min

Fuente:(Elaboración propia.

4.3.1 Resultados del sistema de votación y la blockchain

Para poder observar los votos que contenía la blockchain se realizó el descifrado de los datos mediante la clave privada, a continuación, se muestra la comparación de votos que fueron modificados en el sistema y los votos que no fueron modificados en la blockchain.

Tabla 4.5
Comparación de resultados.

N°	Datos Reales	Datos del sistema de votación	Datos de la Blockchain
1	84	84	84
2	85	84	85
3	85	85	85
4	84	84	84
5	83	85	83
6	84	85	84
7	84	85	84
8	83	83	83
9	85	85	85
10	84	84	84
11	83	83	83
12	85	85	85
13	84	85	84
14	84	84	84
15	85	85	85
16	84	85	84
17	85	85	85
18	85	85	85
19	84	84	84
20	83	83	83

Fuente:(Elaboración propia)

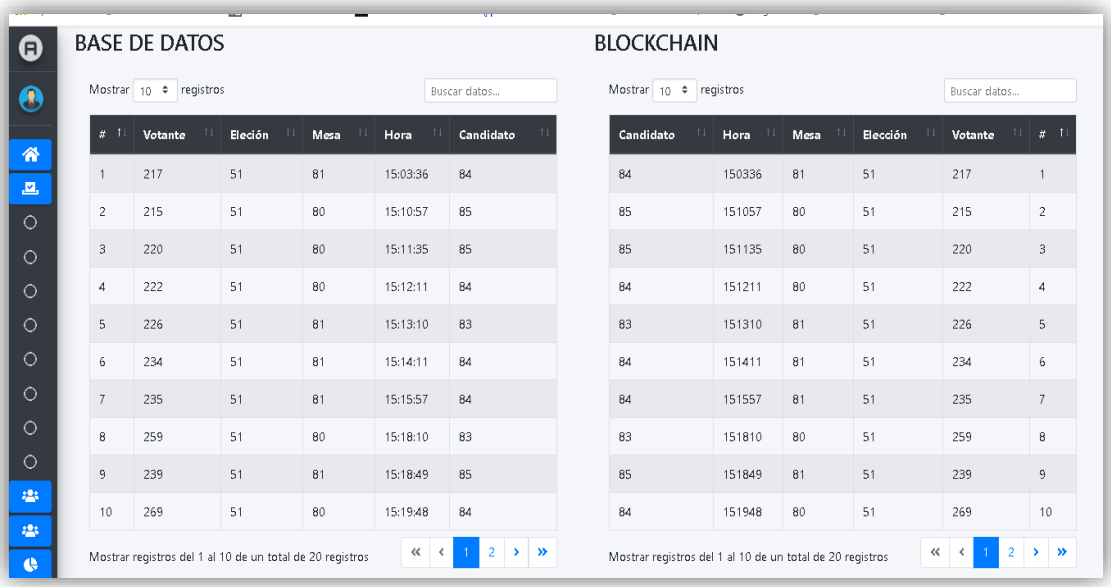


Figura 4.6 Visualización de datos del sistema y la blockchain Parte 1.
Fuente: Elaboración propia.

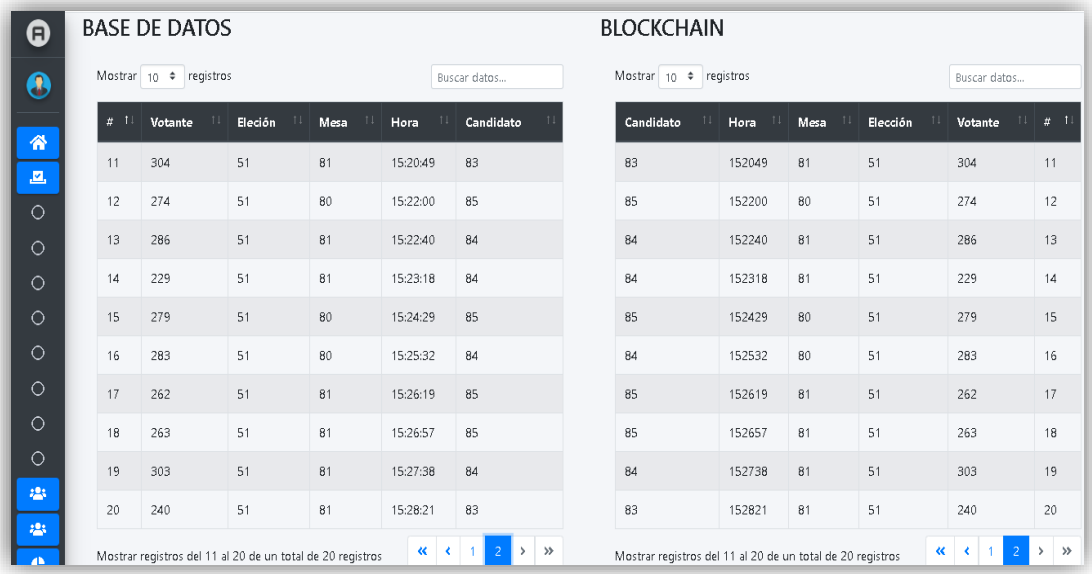


Figura 4.7 Visualización de datos del sistema y la blockchain. Parte 2.
Fuente: Elaboración propia.

Se obtuvo resultados satisfactorios de la blockchain, al no ser modificables y dando fiabilidad a los votos emitidos.

4.4 Prueba de Hipótesis

En este punto se realizará la prueba de hipótesis plantada en el Capítulo uno, demostrar si la hipótesis tiene una confianza del 95%.

4.4.1 Planteamiento de hipótesis

Nos planteamos una hipótesis nula (H_0) y la hipótesis de investigación (H_i)

- **Hipótesis Nula:**

H_0: La aplicación de protocolos blockchain a un sistema de votación electrónica no contribuirá a mejorar la seguridad y transparencia de los procesos electorales.

- **Hipótesis de Investigación:**

H_i: La aplicación de protocolos blockchain a un sistema de votación electrónica contribuirá a mejorar la seguridad y transparencia de los procesos electorales.

4.4.2 Tamaño de la muestra

Para demostrar la hipótesis de investigación planteada $H_{i,}$ se tomó los datos obtenidos de las pruebas realizadas al sistema y a la blockchain.

Se realizó la selección de 20 votos emitidos en el sistema y en la cadena de bloques, se pudo observar que se logró vulnerar el sistema de votación y realizar una manipulación de votos, mientras que en la blockchain no se logró dicho cometido.

De forma que se obtuvo dos muestras que corresponden al sistema de votación electrónica y a la cadena de bloques, se realiza la comparación de los votos, donde se le pondera con el numero 1 a los votos reales, mientras que a los votos modificados se le pondera con el número 0.

En la siguiente tabla se muestra la ponderación de votos.

Tabla 4.6
Comparación de votos validos

N°	Sistema de votación	Votos válidos Sistema X_i	Votos validos Blockchain X_j	Datos de la Blockchain
1	84	1	1	84
2	85	1	1	85
3	85	1	1	85
4	84	1	1	84
5	85	0	1	83
6	85	0	1	84
7	85	0	1	84
8	83	1	1	83
9	85	1	1	85
10	84	1	1	84
11	83	1	1	83
12	85	1	1	85
13	85	0	1	84
14	84	1	1	84
15	85	1	1	85
16	85	0	1	84
17	85	1	1	85
18	85	1	1	85
19	84	1	1	84
20	83	1	1	83
Total votos validos	15	20		
Media	0.75	1		

Fuente: Elaboración propia.

4.4.3 Prueba T de Student

La fórmula que se utilizara para la prueba la de hipótesis es la prueba t de Student.

$$t_0 = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{S^2 \times + \left[\frac{1}{n_1} + \frac{1}{n_2}\right]}}$$

Donde:

n_1 y n_2 son 2 los tamaños de muestra

$\overline{X_1}$ = Promedio de los votos válidos obtenidos del sistema de votación.

$\overline{X_2}$ = Promedio de los votos válidos obtenidos de la blockchain.

Varianza común estimada

La fórmula que realiza la estimación de la Varianza es la siguiente:

$$\boldsymbol{S^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{(n_1 + n_2 - 2)}}$$

Donde:

$n_1 \; y \; n_2$ son el tamaño de las muestras

$\overline{x_1} \; y \; \overline{x_2}$ son el promedio de las muestras

Cálculo de promedios

Anteriormente hallamos el promedio o media de las 2 muestras.

$$\boldsymbol{\overline{x_1} = 0.75; \;\; \overline{x_2} = 1}$$

Cálculo de las varianzas

En la siguiente tabla se muestra los datos de ambas muestras para el cálculo de la varianza.

Tabla 4.7
Cálculo de Varianzas.

N°	Muestra 1 X_i	$(X_i - \bar{x}_1)$	$(X_i - \bar{x}_1)^2$	Muestra 2 X_j	$(X_j - \bar{x}_2)$	$(X_j - \bar{x}_2)^2$
1	1	0.25	0.0625	1	0	0
2	1	0.25	0.0625	1	0	0
3	1	0.25	0.0625	1	0	0
4	1	0.25	0.0625	1	0	0
5	0	0.75	0.5625	1	0	0
6	0	0.75	0.5625	1	0	0
7	0	0.75	0.5625	1	0	0
8	1	0.25	0.0625	1	0	0
9	1	0.25	0.0625	1	0	0
10	1	0.25	0.0625	1	0	0
11	1	0.25	0.0625	1	0	0
12	1	0.25	0.0625	1	0	0
13	0	0.75	0.5625	1	0	0
14	1	0.25	0.0625	1	0	0
15	1	0.25	0.0625	1	0	0
16	0	0.75	0.5625	1	0	0
17	1	0.25	0.0625	1	0	0
18	1	0.25	0.0625	1	0	0
19	1	0.25	0.0625	1	0	0
20	1	0.25	0.0625	1	0	0
Total	15		3.75	20		0

Fuente: Elaboración propia.

Reemplazamos los datos obtenidos en la siguiente formula:

$$S_1{}^2 = \frac{\sum(x_i - \overline{x})^2}{n-1}$$

$$S_1{}^2 = \frac{3.75}{14} = 0.268$$

$$S_2{}^2 = \frac{\sum(x_j - \overline{x})^2}{n-1}$$

$$S_2{}^2 = \frac{0}{19} = 0$$

Calculamos la varianza común estimada

$$S^2 = \frac{(14) * 0.268 + (19) * 0}{33} = 0.114$$

Calculo del Estadístico “t”

$$t_{calculado} = \frac{0.75 - 1}{\sqrt{0.114 * \left[\frac{1}{15} + \frac{1}{20}\right]}} = -2.167$$

Nivel de confianza y grado de libertad

$$\infty = 0.05; 1 - 0.05 = 0.95 \rightarrow nivel\ de\ confianza$$

$$gl = 15 + 20 - 2 = 33 \rightarrow grados\ de\ libertad$$

Buscamos el valor crítico con $gl = 33$ y error estimado $\frac{\infty}{2} = 0.025$ en la tabla de valores críticos de la distribución t de Student.

Tabla 4.8

Valores críticos de la distribución t de Student

	área a la derecha de *t*															
	0,0005	0,0025	0,005	0,0075	0,01	0,015	0,02	0,025	0,05	0,1	0,15	0,2	0,3	0,4	0,45	
g.d.l																g.d.l
1	636,619	127,321	63,657	42,433	31,821	21,205	15,895	12,706	6,314	3,078	1,963	1,376	0,727	0,325	0,158	1
2	31,599	14,089	9,925	8,073	6,965	5,643	4,849	4,303	2,920	1,886	1,386	1,061	0,617	0,289	0,142	2
3	12,924	7,453	5,841	5,047	4,541	3,896	3,482	3,182	2,353	1,638	1,250	0,978	0,584	0,277	0,137	3
4	8,610	5,598	4,604	4,088	3,747	3,298	2,999	2,776	2,132	1,533	1,190	0,941	0,569	0,271	0,134	4
5	6,869	4,773	4,032	3,634	3,365	3,003	2,757	2,571	2,015	1,476	1,156	0,920	0,559	0,267	0,132	5
6	5,959	4,317	3,707	3,372	3,143	2,829	2,612	2,447	1,943	1,440	1,134	0,906	0,553	0,265	0,131	6
7	5,408	4,029	3,499	3,203	2,998	2,715	2,517	2,365	1,895	1,415	1,119	0,896	0,549	0,263	0,130	7
8	5,041	3,833	3,355	3,085	2,896	2,634	2,449	2,306	1,860	1,397	1,108	0,889	0,546	0,262	0,130	8
9	4,781	3,690	3,250	2,998	2,821	2,574	2,398	2,262	1,833	1,383	1,100	0,883	0,543	0,261	0,129	9
10	4,587	3,581	3,169	2,932	2,764	2,527	2,359	2,228	1,812	1,372	1,093	0,879	0,542	0,260	0,129	10
11	4,437	3,497	3,106	2,879	2,718	2,491	2,328	2,201	1,796	1,363	1,088	0,876	0,540	0,260	0,129	11
12	4,318	3,428	3,055	2,836	2,681	2,461	2,303	2,179	1,782	1,356	1,083	0,873	0,539	0,259	0,128	12
13	4,221	3,372	3,012	2,801	2,650	2,436	2,282	2,160	1,771	1,350	1,079	0,870	0,538	0,259	0,128	13
14	4,140	3,326	2,977	2,771	2,624	2,415	2,264	2,145	1,761	1,345	1,076	0,868	0,537	0,258	0,128	14
15	4,073	3,286	2,947	2,746	2,602	2,397	2,249	2,131	1,753	1,341	1,074	0,866	0,536	0,258	0,128	15
16	4,015	3,252	2,921	2,724	2,583	2,382	2,235	2,120	1,746	1,337	1,071	0,865	0,535	0,258	0,128	16
17	3,965	3,222	2,898	2,706	2,567	2,368	2,224	2,110	1,740	1,333	1,069	0,863	0,534	0,257	0,128	17
18	3,922	3,197	2,878	2,689	2,552	2,356	2,214	2,101	1,734	1,330	1,067	0,862	0,534	0,257	0,127	18
19	3,883	3,174	2,861	2,674	2,539	2,346	2,205	2,093	1,729	1,328	1,066	0,861	0,533	0,257	0,127	19
20	3,850	3,153	2,845	2,661	2,528	2,336	2,197	2,086	1,725	1,325	1,064	0,860	0,533	0,257	0,127	20
21	3,819	3,135	2,831	2,649	2,518	2,328	2,189	2,080	1,721	1,323	1,063	0,859	0,532	0,257	0,127	21
22	3,792	3,119	2,819	2,639	2,508	2,320	2,183	2,074	1,717	1,321	1,061	0,858	0,532	0,256	0,127	22
23	3,768	3,104	2,807	2,629	2,500	2,313	2,177	2,069	1,714	1,319	1,060	0,858	0,532	0,256	0,127	23
24	3,745	3,091	2,797	2,620	2,492	2,307	2,172	2,064	1,711	1,318	1,059	0,857	0,531	0,256	0,127	24
25	3,725	3,078	2,787	2,612	2,485	2,301	2,167	2,060	1,708	1,316	1,058	0,856	0,531	0,256	0,127	25
26	3,707	3,067	2,779	2,605	2,479	2,296	2,162	2,056	1,706	1,315	1,058	0,856	0,531	0,256	0,127	26
27	3,690	3,057	2,771	2,598	2,473	2,291	2,158	2,052	1,703	1,314	1,057	0,855	0,531	0,256	0,127	27
28	3,674	3,047	2,763	2,592	2,467	2,286	2,154	2,048	1,701	1,313	1,056	0,855	0,530	0,256	0,127	28
29	3,659	3,038	2,756	2,586	2,462	2,282	2,150	2,045	1,699	1,311	1,055	0,854	0,530	0,256	0,127	29
30	3,646	3,030	2,750	2,581	2,457	2,278	2,147	2,042	1,697	1,310	1,055	0,854	0,530	0,256	0,127	30
31	3,633	3,022	2,744	2,576	2,453	2,275	2,144	2,040	1,696	1,309	1,054	0,853	0,530	0,256	0,127	31
32	3,622	3,015	2,738	2,571	2,449	2,271	2,141	2,037	1,694	1,309	1,054	0,853	0,530	0,255	0,127	32
33	3,611	3,008	2,733	2,566	2,445	2,268	2,138	2,035	1,692	1,308	1,053	0,853	0,530	0,255	0,127	33
34	3,601	3,002	2,728	2,562	2,441	2,265	2,136	2,032	1,691	1,307	1,052	0,852	0,529	0,255	0,127	34
35	3,591	2,996	2,724	2,558	2,438	2,262	2,133	2,030	1,690	1,306	1,052	0,852	0,529	0,255	0,127	35
40	3,551	2,971	2,704	2,542	2,423	2,250	2,123	2,021	1,684	1,303	1,050	0,851	0,529	0,255	0,126	40
60	3,460	2,915	2,660	2,504	2,390	2,223	2,099	2,000	1,671	1,296	1,045	0,848	0,527	0,254	0,126	60
80	3,416	2,887	2,639	2,486	2,374	2,209	2,088	1,990	1,664	1,292	1,043	0,846	0,526	0,254	0,126	80
90	3,402	2,878	2,632	2,480	2,368	2,205	2,084	1,987	1,662	1,291	1,042	0,846	0,526	0,254	0,126	90
100	3,390	2,871	2,626	2,475	2,364	2,201	2,081	1,984	1,660	1,290	1,042	0,845	0,526	0,254	0,126	100
120	3,373	2,860	2,617	2,468	2,358	2,196	2,076	1,980	1,658	1,289	1,041	0,845	0,526	0,254	0,126	120
inf.	3,291	2,807	2,576	2,432	2,326	2,170	2,054	1,960	1,645	1,282	1,036	0,842	0,524	0,253	0,126	inf.

Fuente: UTN,2020.

Se pudo obtener como valor $t_{buscado}$: 2.035

Regla de decisión

$$S_i : t_{calculado} < t_{buscado} \rightarrow Se\ rechaza\ H_0\ y\ se\ acepta\ H_i$$

$$S_i : t_{calculado} > t_{buscado} \rightarrow Se\ acepta\ H_0\ y\ se\ rechaza\ H_i$$

Analisis de resultados

Anteriormente obtuvimos:

$$\boldsymbol{t_{calculado}} = -2.167$$

$$\boldsymbol{t_{buscado}} = 2.035$$

Según la regla de decisión

$$t_{calculado} < t_{buscado}$$

$$-2.167 < 2.035$$

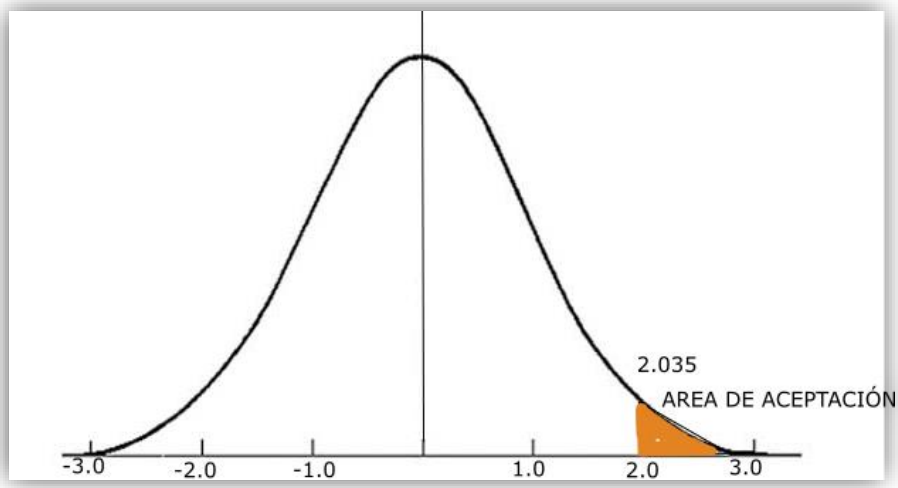


Figura 4.8 Gráfico de distribución t de Student.
Fuente: Elaboración propia.

Como el valor $t_{calculado}$ es menor al valor $t_{buscado}$, entonces se rechaza la hipótesis nula y se acepta la hipótesis de investigación, con una confianza del 95% por lo cual se llega a la conclusión: La aplicación de protocolos blockchain a un sistema de votación electrónica contribuirá a mejorar la seguridad y transparencia de los procesos electorales con una confianza del 95%.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES Y RECOMENDACIONES

5.1 Introducción

Para finalizar el presente trabajo de investigación, en este capítulo se describen las conclusiones a las que se llegó después de un arduo trabajo, además se describen las recomendaciones importantes y necesarias para una futura implementación o investigación en el área de la tecnología blockchain.

5.2 Conclusiones

Los métodos y la metodología de desarrollo utilizadas fueron fundamentales para el desarrollo del tema de investigación, para llegar a cumplir con los objetivos trazados, podemos decir que llegamos a cumplir satisfactoriamente el objetivo principal. A continuación, las conclusiones de acuerdo a los objetivos:

- Se analizó de manera rigurosa la tecnología blockchain, además de comprender las ventajas que contempla esta tecnología.
- Se creó un sistema de votación electrónica para poder agilizar un proceso electoral desde la creación de una elección hasta el escrutinio y conteo de votos.
- Se diseñó de una interfaz gráfica fácil y amigable para la interacción con el usuario final.
- Se demostró la aplicación de la tecnología blockchain a un sistema de votación electrónica.
- Se logró mostrar el nivel de seguridad que brinda la tecnología blockchain a los votos que se emiten a través del sistema de votación electrónica.

5.2 Recomendaciones

Los objetivos fueron cumplidos satisfactoriamente sin embargo se recomienda lo siguiente:

- Para un mejor control de votantes se recomienda considerar la implementación del sistema biométrico (Reconocimiento de huella dactilar).
- Se recomienda el uso de algoritmos de encriptación en más módulos del sistema de votación electrónica.
- Como alternativa se recomienda realizar futuros desarrollos de aplicaciones descentralizadas utilizando las plataformas Ethereum, Hyperledger, y otros.
- Por último se recomienda el estudio a profundidad de la tecnología blockchain para futuras aplicaciones en otros campos.

BIBLIOGRAFÍA

ACE. (20 de Abril de 2020). *Red de Conocimientos Electorales*. Obtenido de https://aceproject.org/ace-es/topics/es/onePage

Alexandre, A. (10 de mayo de 2018). *EE.UU: West Virginia Completa las Primeras Elecciones Estatales Apoyadas en la Cadena de Bloques.* Recuperado el 28 de septiembre de 2018, de cointelegraph: https://es.cointelegraph.com/news/us-west-virginia-completes-first-blockchain-supported-state-elections

Alvarez, M. A. (2017). Manual de CodeIgniter. *DesarrolloWeb*, 2.

Angel, J. d. (25 de abril de 2020). *unicauca.* Obtenido de http://seguridad.unicauca.edu.co/criptografia/CriptografiaParaPrincipiantes.pdf

Aparicio, C. (05 de mayo de 2020). *Escuela de Organización Industrial*. Obtenido de https://www.eoi.es/blogs/cesaraparicio/2012/05/06/el-modelo-cocomo-para-estimar-costes-en-un-proyecto-de-software/

Arias, F. (2012). *Proyecto de Investigación.* Caracas: Editorial Episteme.

Auren; ACCID; ALHOS; UPF; Barcelona School; Economistas contables. (2019). *Blockchain, bitcoin y criptomonedas: Bases conceptuales y aplicaciones prácticas.* España: Profit Editorial.

Barceló Ordinas, J. M., Íñigo Griera, J., Martí Escalé, R., & Peig Olivé, E. (2004). *Redes de computadores.* Barcelona: Fundació per a la Universitat Oberta de Catalunya.

Berni, P., & Gil de la Iglesia, D. (2010). *Laboratorio de PHP y MySql.* Barcelona: Eureca Media, SL.

Camacho, C. (12 de Octubre de 2017). Jóvenes usan el “blockchain” para votación y salud. *Los Tiempos*, págs. 5 - 6.

Chávez Abad, R. (2015). *Introducción a la Metodología de la Investigación.* Machala: Universidad Técnica de Machala.

Collell, J. (2013). *CSS3 y Javascript avanzado.* Barcelona: UOC.

Correo del sur. (20 de junio de 2018). El voto electrónico llegó a 16 colegios de la región. *Correo del Sur*, pág. 9.

Diaz, L. (2011). *LA OBSERVACION.* México.

Dolader, C., Bel, J., & Muñoz, J. L. (2017). *Universitat Politécnica de Catalunya.* Barcelona: Universitat Politécnica de Catalunya.

EcuRed. (10 de mayo de 2020). *EcuRed*. Obtenido de https://www.ecured.cu/MariaDB#cite_note-1

ECURED. (20 de marzo de 2020). *ECURED*. Obtenido de https://www.ecured.cu/CodeIgniter

Egea, F. J. (2000). *Servidores para Internet con Apache HTTPServer.* Madrid: Grupo EIDOS.

Eguiluz, J. (05 de julio de 2019). *uniwebsidad.* Recuperado el 20 de 11 de 2018, de https://librosweb.es/libro/javascript/capitulo-1.html

EL DIARIO. (17 de Mayo de 2018). Cinco tipos de voto electrónico. *El Diario*.

Escuela Penitenciaria Nacional. (10 de Abril de 2019). Obtenido de http://epn.gov.co/elearning/distinguidos/SEGURIDAD/1_conceptos_de_seguridad.html

Federico, A. (25 de Mayo de 2019). *Astec*. Obtenido de https://medium.com/astec/entendiendo-los-protocolos-de-consenso-de-blockchain-4858c71722d2

Gargantilla, P. (19 de 02 de 2020). *ABC Ciencia*. Obtenido de https://www.abc.es/ciencia/abci-metodo-cientifico-estos-cinco-pasos-201902170129_noticia.html?ref=https:%2F%2Fwww.google.com%2F

Gobierno de España. (12 de Enero de 2020). Obtenido de https://www.csd.gob.es/es/federaciones-y-asociaciones/federaciones-deportivas-espanolas/procesos-electorales-y-voto-electronico/sistema-de-voto

Gómez, A., López, M., Migani, S., & Otazú, A. (10 de mayo de 2020). Obtenido de https://blogadmi1.files.wordpress.com/2010/11/cocom0llfull.pdf

Granados, G. (10 de Julio de 2006). INTROUCCION A LA CRIPTOGRAFIA. México, México.

Hernandez, H. P. (2010). *Propuesta de análisis y diseño basada en UML y UWE para la migración de arquitectura de software centralizada hacia internet.* Guatemala: Universidad de San Carlos de Guatemala.

IBM. (18 de marzo de 2018). *IBM*. Recuperado el 20 de noviembre de 2018, de Aspectos básicos de blockchain: Introducción a los controladores distribuidos: https://www.ibm.com/developerworks/ssa/cloud/library/cl-blockchain-basics-intro-bluemix-trs/index.html

Instituto Científico de Gobierno Electrónico. (25 de Abril de 2020). *e-gobiernos.org*. Obtenido de https://e-gobiernos.org/2017/08/04/voto-electronico-blockchain/

Instituto Internacional para la Democracia y la Asistencia Electoral. (2011). *Una introducción al voto eletrónico.* Suecia.

Kendall, & Kendall. (2011). *Análisis y Diseño de Sistemas.* Mexico: Pearson Education.

López, G. (25 de Enero de 2020). *Gobierno de México.* Obtenido de http://www.ordenjuridico.gob.mx/Congreso/pdf/20.pdf

Lucena Lopez, M. J. (2011). *CRIPTOGRAFÍA Y SEGURIDAD EN COMPUTADORES.* Jaen: UNIVERSIDAD DE JAÉN.

Lujan, S. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web.* San Vicente: Club Universitario.

Mamani Ortiz, Y. (2014). *Introducción a la Metodología de Investigación.* Cochabamba.

Manual Web. (10 de marzo de 2020). Obtenido de http://www.manualweb.net/flask/introduccion-flask/

Maya, E. (2014). *Métodos y Técnicas de investigación.* México: Universidad Nacional Autónoma de México.

Minguez, D., & Garcia, E. (2010). Metodologías para el Desarrollo de aplicaciones Web: UWE. 10.

Molina Rios, J., & Zea Ordoñez, M. (2017). Metodología de Desarrollo en Aplicaciones Web. *ARJÉ*, 247.

Moreno, A. M. (2012). *Estimación de Proyectos Software.* Madrid.

Moreno, M., Toledo, A., Lopez, C., & Cruz, A. (10 de abril de 2020). *Universidad Tecnologica de la Selva*. Obtenido de http://iso9126uts.blogspot.com/

Navarro Sánchez, S. (2019). *Solución P2P embebida.* Madrid: UNIVERSIDAD AUTÓNOMA DE MADRID.

Navarro, B. Y. (2017). *Blockchain y sus aplicaciones.* Asuncion: Universidad Católica Nuestra Señora de la Asunción.

Nolivos, G. C. (2013). *ANÁLISIS, DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA WEB PARA EL CONTROL DE UN TALLER TÉCNICO AUTOMOTRIZ EN PLATAFORMA PHP - MYSQL UTILIZANDO METODOLOGÍA WEB UWE PARA LA EMPRESA.* Sangolqui: Universidad de Fuerzas Armadas - ESPE.

Pereira, B., Ayaach, F., Quintero, H., Granadillo, I., & Bustamante, J. (5 de Febrero de 2013). *https://es.slideshare.net/.* Obtenido de SlideShare: https://es.slideshare.net/isisparada/metricas-de-calidad-de-software

php. (14 de marzo de 2020). *php.net*. Obtenido de https://www.php.net/manual/es/intro-whatis.php

Plasencia, J. E. (2018). *Sistema de votación electrónica.* Santa Cruz de Tenerife - España.

Point, T. (10 de mayo de 2020). *tutorialspoint.* Obtenido de https://www.tutorialspoint.com/mariadb/mariadb_tutorial.pdf

Pressman, R. (2010). *Ingeniería de Software, Un enfoque práctico.* Mexico: McGraw-Hill.

Preukschat, A., Kuchkovsky, C., & Gómez, G. (2017). *BLOCKCHAIN: LA REVOLUCIÓN INDUSTRIAL.* Barcelona: Black Print.

Price, A. (2006). *Consideraciones, aportes y experiencias para el VOTO ELECTRÓNICO EN ARGENTINA.* Buenos Aires: Dunken.

Python. (23 de mayo de 2020). *Python Software Foundation.* Obtenido de https://docs.python.org/es/3.8/tutorial/index.html

Quishpe, J. P. (2013). *AUTOMATIZACIÓN DEL SEGUIMIENTO DE PROYECTOS Y CONTRATOS.* Quito.

Ramió Aguirre, J. (5 de mayo de 2020). *criptored.* Obtenido de http://www.criptored.upm.es/descarga/CursoCriptografiaAplicada2018.pdf

Rodriguez Aragón, L. (25 de Febrero de 2020). *Internet y Teleinformática*. Obtenido de Universidad de Castilla-La Mancha: https://previa.uclm.es/profesorado/licesio/Docencia/IB/IBTema4.pdf

Rodriguez Ojeda, L. (2015). *Python Programación.* Guayaquil: Escuela Superior Politécnica del Litoral.

Sánchez, H. C., Rodríguez, C. C., & Notario, A. C. (5 de mayo de 2020). *uah.* Obtenido de http://www3.uah.es/libretics/concurso2014/files2014/Trabajos/Criptografia%20y%20Metodos%20de%20Cifrado.pdf

Sánchez, S., Domínguez, P., & Velásquez, L. (2019). Hashing: Técnicas y Hash para la Protección de Datos. *Universidad Tecnológica de Panamá*, 3.

Sommerville, I. (2011). *Ingeniería de software.* Mexico: Pearson Education.

Sura F. Yousif. (2018). “ENCRYPTION AND DECRYPTION OF AUDIO SIGNAL BASED ON RSA ALGORITHN.” International Journal of Engineering Technologies and Management Research, 5(7), 57-64. DOI: 10.5281/zenodo.1341956.

Téllez, J. (2010). *El Voto electrónico.* México: Tribunal Electoral del Poder Judicial de la Federación.

Universidad Tecnológica Nacional Facultad Regional Buenos Aires. (2 de Octubre de 2015). Publicación de la Facultad Regional Buenos Aires. *Proyecciones*, 24. Obtenido de https://www.frba.utn.edu.ar/wp-content/uploads/2016/05/Proyecciones-oct-15.pdf#page=21

WIKIPEDIA. (25 de Febrero de 2020). Obtenido de https://es.wikipedia.org/wiki/Protocolo_de_comunicaciones

Wikipedia. (25 de Abril de 2020). *Wikipedia La enciclopedia ibre*. Obtenido de https://es.wikipedia.org/wiki/Servidor_HTTP_Apache

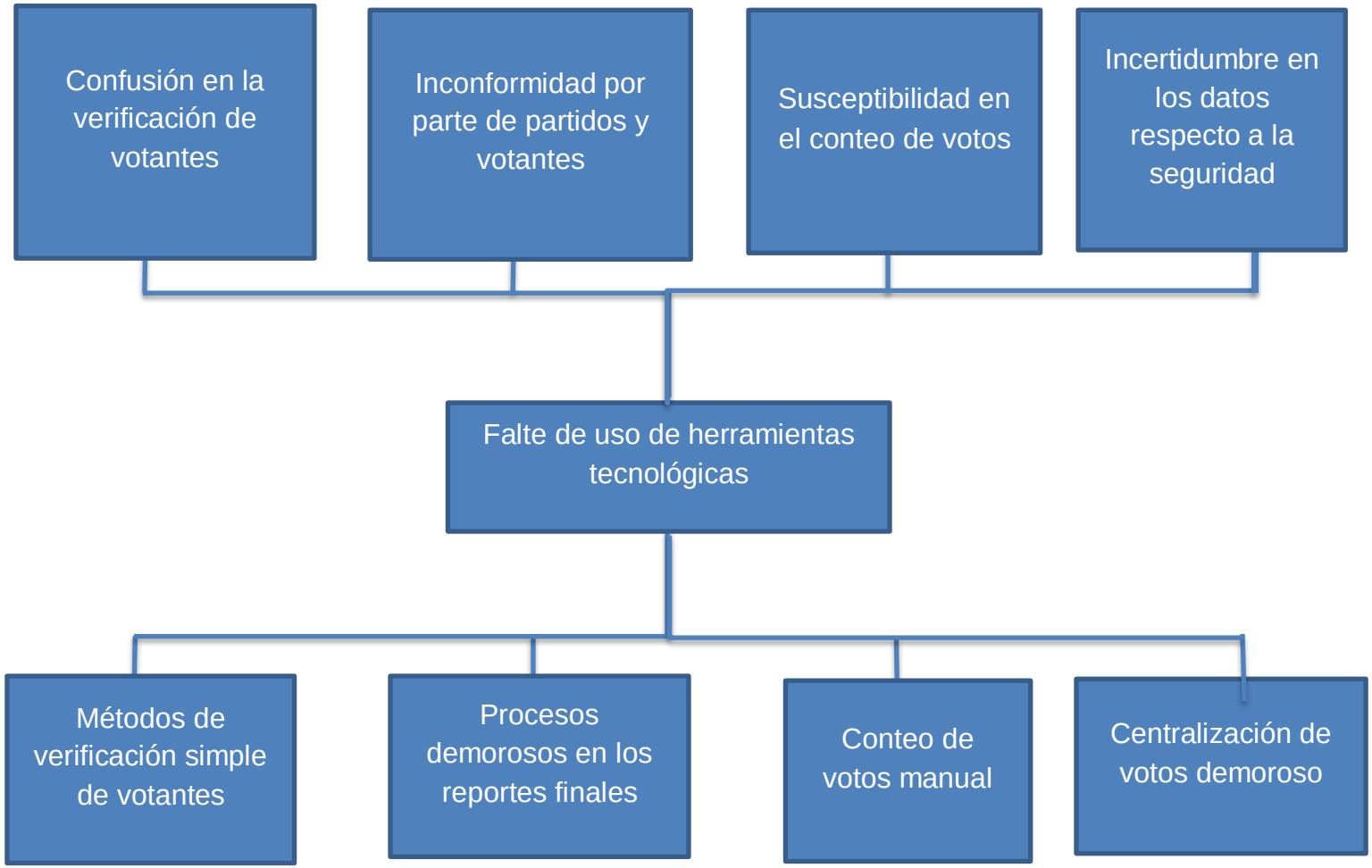
Wikipedia. (10 de Mayo de 2020). *Wikipedia La enciclopedia libre*. Obtenido de https://es.wikipedia.org/wiki/MySQL

Wikipedia. (10 de abril de 2020). *Wikipedia La enciclopedia libre*. Obtenido de https://es.wikipedia.org/wiki/COCOMO

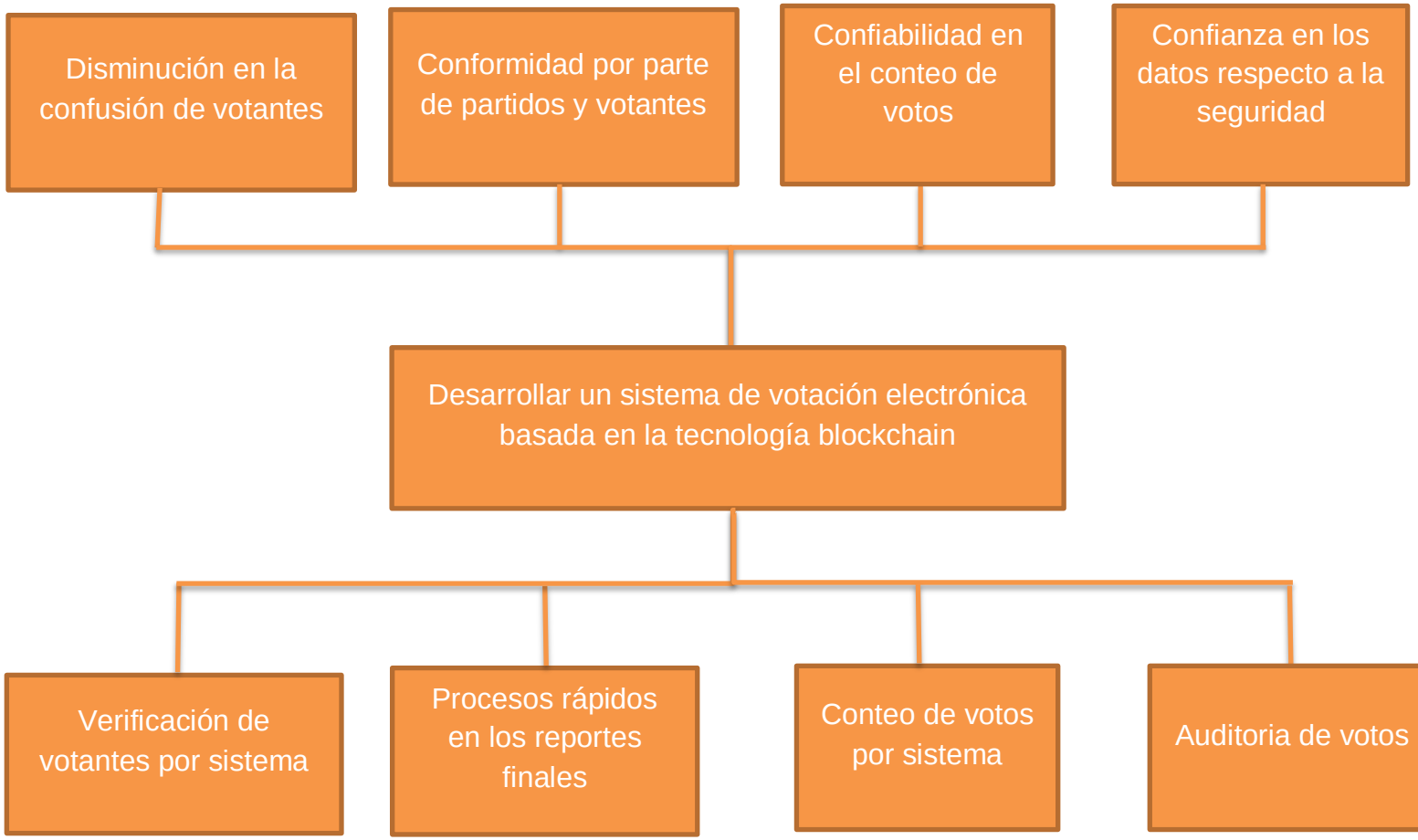
Wikipedia. (11 de Frebrero de 2020). *WIKIPEDIA La enciclopedia libre*. Obtenido de https://es.wikipedia.org/wiki/Cadena_de_bloques

Wikipedia. (01 de junio de 2020). *Wkipedia La enciclopedia libre.* Obtenido de https://es.wikipedia.org/wiki/Firma_digital#cite_note-2

ANEXOS



Anexo A – Árbol Problemas



Anexo B – Árbol Objetivos

El Alto, julio del 2020

Señor:
Ing. David Carlos Mamani Quispe
DIRECTOR DE LA CARRERA DE INGENIERÍA DE SISTEMAS
Presente. -

Ref.- AVAL DE CONFORMIDAD

Distinguido Ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad de la tesis de grado propuesto **"PROTOCOLOS BLOCKCHAIN APLICADOS A UN SISTEMA DE VOTACIÓN ELECTRÓNICA" CASO: CARRERA INGENIERÍA DE SISTEMAS - UPEA**, que propone el postulante Hector Churata Sonco, con cedula de identidad 9984083 L.P., para su defensa pública, evaluación correspondiente de la materia de Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente.

Ing. Enrique Flores Baltazar
TUTOR METODOLÓGICO

El Alto, 10 de julio del 2020

Señor:
Ing. Enrique Flores Baltazar
TUTOR METODOLÓGICO TALLER DE LICENCIATURA II
CARRERA INGENIERÍA DE SISTEMAS - U.P.E.A.
Presente. -

Ref.- AVAL DE CONFORMIDAD

Distinguido Ingeniero,

Mediante la presente tengo a bien comunicarle mi conformidad de la tesis de grado propuesto **"PROTOCOLOS BLOCKCHAIN APLICADOS A UN SISTEMA DE VOTACIÓN ELECTRÓNICA" CASO: CARRERA INGENIERÍA DE SISTEMAS – UPEA**, que propone el postulante Hector Churata Sonco, con cedula de identidad 9984083 L.P., para su defensa pública, evaluación correspondiente de la materia de Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingenieria de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente,

Ing. Ramiro Kantuta Limachi
TUTOR ESPECIALISTA

El Alto, 10 de julio del 2020

Señor:
Ing. Enrique Flores Baltazar
TUTOR METODOLÓGICO TALLER DE LICENCIATURA II
CARRERA INGENIERÍA DE SISTEMAS - U.P.E.A.
Presente. -

Ref.- AVAL DE CONFORMIDAD

Distinguido Ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad de la tesis de grado propuesto **"PROTOCOLOS BLOCKCHAIN APLICADOS A UN SISTEMA DE VOTACIÓN ELECTRÓNICA" CASO: CARRERA INGENIERÍA DE SISTEMAS – UPEA**, que propone el postulante Hector Churata Sonco, con cedula de identidad 9984083 L.P., para su defensa pública, evaluación correspondiente de la materia de Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente.

Ing. Elías Carlos Hidalgo Mamani
TUTOR REVISOR

Universidad Pública de El Alto

Manual de Administrador

SISTEMA DE VOTACIÓN ELECTRÓNICA

AUTOR: HECTOR CHURATA SONCO

2020

MANUAL DE ADMINISTRADOR

El sistema de votación electrónica basada en la tecnología blockchain está destinado a realizar las tareas fundamentales para llevar a cabo una nueva elección en sus tres fases: Pre- votación, votación, post- votación.

1. Objetivo

Establecer los pasos para administrar el sistema de votación electrónica de manera correcta para llevar a cabo un proceso de elección satisfactorio.

2. Requisitos de Hardware

- Pc con procesador equivalente a Intel Core i3 o superior.
- Memoria Ram de 4 GB. O superior.
- Disco duro de una memoria de 500 Megabytes o superior.
- Tarjeta de conexión a red inalámbrica.
- Dispositivos de entrada y salida.

3. Requisitos de Software

- Sistemas operativos: (Windows o Linux).
- Base de datos Maria DB.
- Servidor Apache.

4. Desarrollo del Manual de Administración

El administrador tendrá el mayor control sobre el sistema, las tareas que cumple son:

- Crear una nueva elección.
- El administrador tiene la capacidad de registrar partidos políticos.
- Habilitar a los Candidatos para la elección.
- Realizar la asignación de jurados.
- Asignar Terminales a las mesas habilitadas

- Iniciar la elección
- Finalizar elección
- Tendrá el control para realizar el conteo de votos.
- Por último, el administrador cumplirá la función más importante que es la auditoria de votos que consiste en la visualización de los votos registrados en el sistema y la blockchain con el fin de verificar los resultados.

4.1. Acceso al Sistema

Para acceder al sistema el administrador tendrá que ingresar su usuario y contraseña luego el sistema lo re direccionará al panel de administración.

4.2. Panel Principal del Administrador

Una vez que el administrador inicie sesión, el sistema lo re direcciona al panel principal donde tendrá a disponibilidad todas las funciones del sistema como se puede observar en la siguiente figura.

4.3. Tareas a Realizar en la Fase Pre – Votación

A continuación, los pasos a realizar para una nueva elección.

a) Crear nueva elección.

Para Crear una nueva elección tenemos la opción de Crear nueva elección.

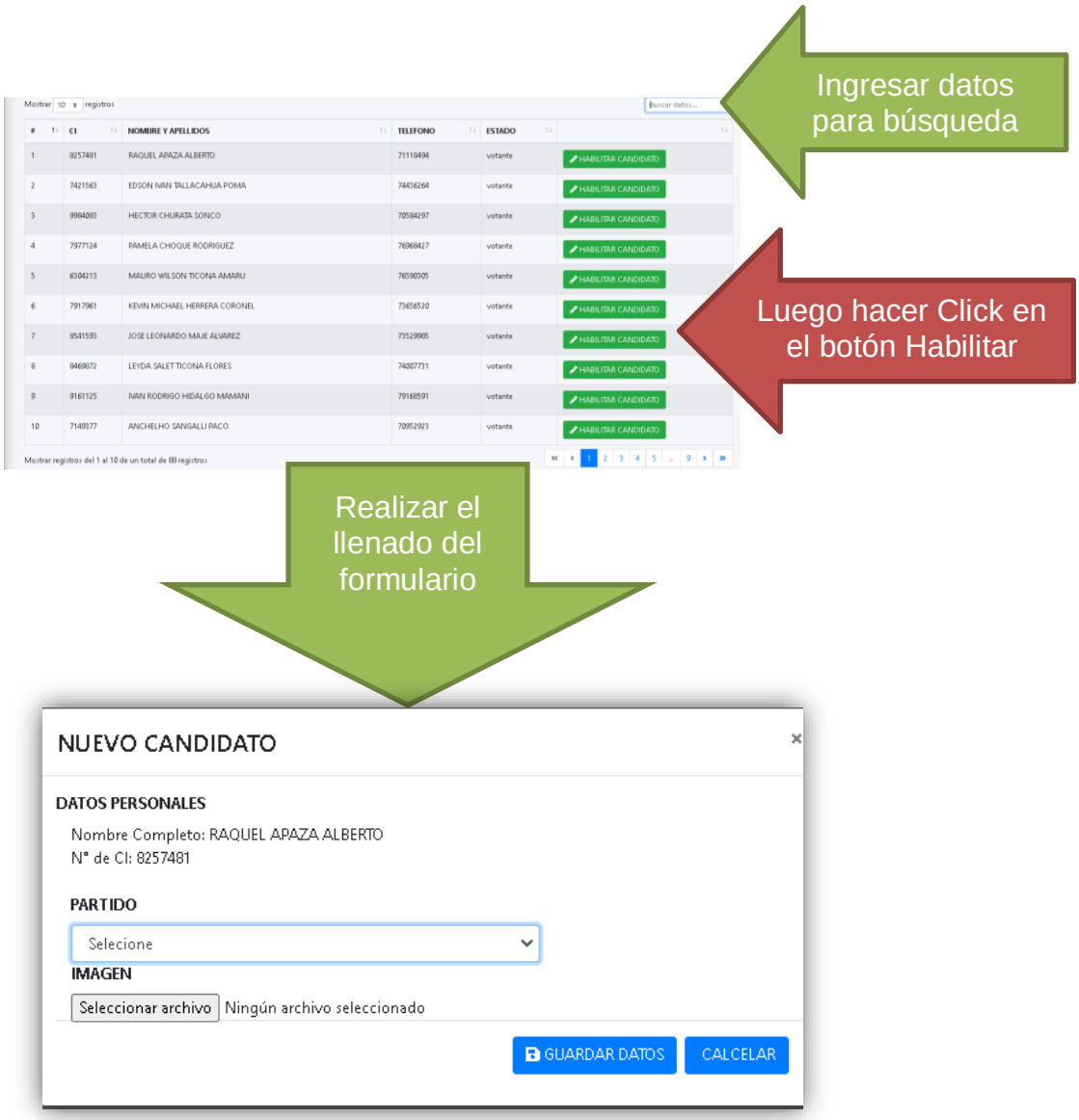
Luego se procede a llenar el siguiente formulario completando los datos que requiere el sistema para la realización de una nueva elección.

b) Registro de partidos

EL administrador realizara la verificación de los requisitos que deben presentar los partidos y frentes políticos para luego registrarlos al sistema mediante el siguiente formulario

c) Habilitación de candidatos

EL sistema visualiza el listado de personas que estén registradas en la base de datos. Para la habilitación de candidatos el administrador deberá realizar una búsqueda del postulante a candidato, para luego Habilitarlo y llenar el formulario de inscripción.



d) Selección de jurados y habilitación de mesas

La siguiente tarea es la selección de jurados, el sistema cuenta con una función que realiza el proceso de selección aleatoria, sin embargo, el administrador hará posible esto una vez llene el siguiente formulario.

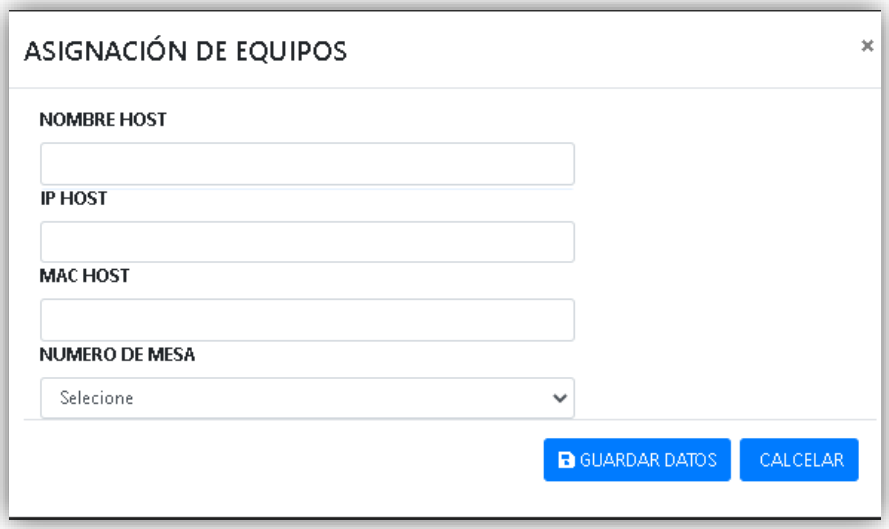
e) Generar Usuario y contraseña para jurados

Una vez que tengamos a los jurados seleccionados, se procede a generar Usuario y contraseña para que los jurados accedan al sistema.

Luego imprimir un PDF donde se mostrarán los Usuarios y contraseña de los jurados.

f) Asignación de equipos

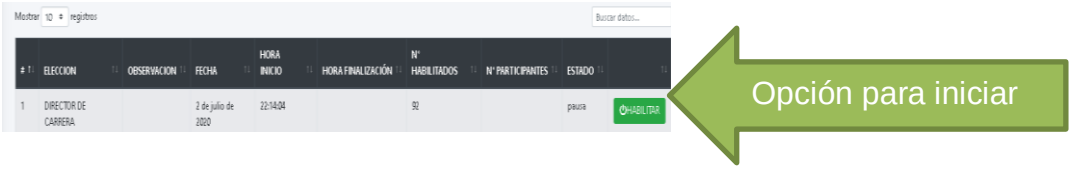
La ultima tarea que realizara el administrador en el registro de Terminales y asignación a las mesas electorales completando el siguiente formulario.



4.4. Tareas a Realizar en la Fase Votación

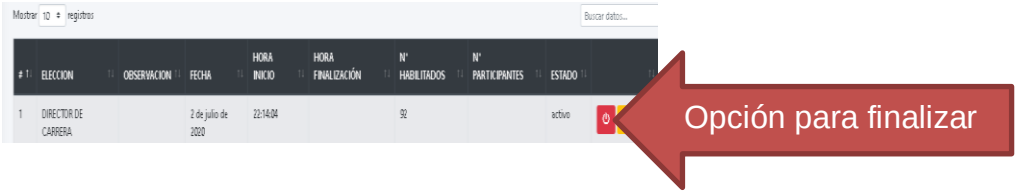
a) Iniciar elección.

El administrador cuenta con el control para iniciar una elección que consiste en dar permiso a los jurados y votantes para que comiencen a emitir votos.



b) Finalizar elección.

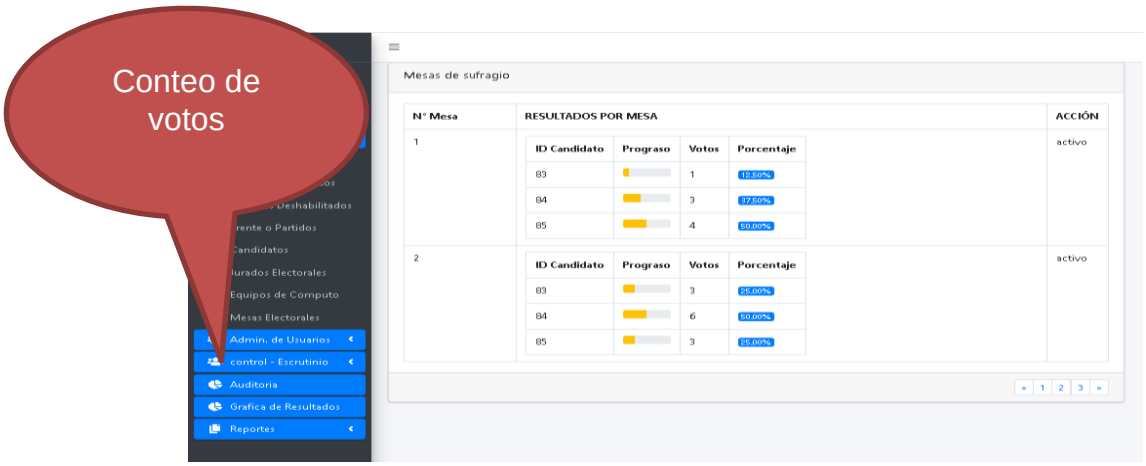
De la misma forma para la finalización.



4.5. Tareas a Realizar en la Fase Post – Votación

a) Conteo de votos

Una vez concluida la elección el administrador del sistema deberá realizar el escrutinio y posterior conteo de votos, para eso solo deberá ir a la opción Escrutinio y conteo de votos.



El sistema recorrerá registro por registro en la base de datos para luego visualizar los resultados clasificándolos por mesa de sufragio.

b) Visualización de resultados finales

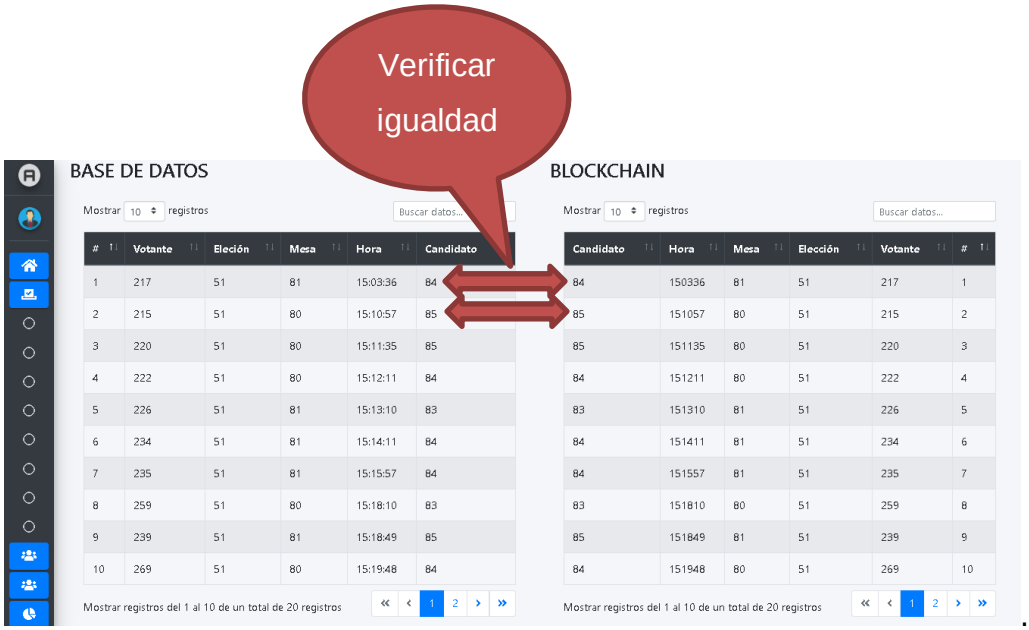
El administrador deberá ir a la opción “Grafica de resultados”, luego el sistema visualizará en una gráfica los resultados generales por candidato.

c) Reportes

Una vez el administrador conforme con los resultados, el administrador emitirá un reporte sobre los resultados que se obtuvieron.

d) Auditoria de Votos

En caso de que existe inconformidad por parte de los candidatos o los veedores, el administrador tendrá una opción llamada “Auditoria”, lo único que debe realizar es presionar sobre esta opción y luego se listara todos los votos registrados en la base de datos y otra lista de los votos registrados en la blockchain, el administrador deberá verificar que los votos coincidan



Una vez realizada la verificación se podrá dar validez a los votos y finalizar completamente la elección.

5. Conclusión

El sistema brinda una interfaz amigable, de fácil manejo en las diferentes tareas de administración, además de que ayuda al administrador en realizar fácilmente la auditoria de votos.

Universidad Pública de El Alto

Manual de Usuario

SISTEMA DE VOTACIÓN ELECTRÓNICA

AUTOR: HECTOR CHURATA SONCO

2020

MANUAL DE USUARIO

1. Objetivo

Establecer los pasos para el proceso de emisión de votos en el sistema de votación electrónica, para una buena experiencia en la interacción con el sistema.

2. Desarrollo del Manual de Usuario

Existen dos tipos de Usuarios que solo participaran en la etapa de la votación, estos son el tipo de usuario Jurado y Votante.

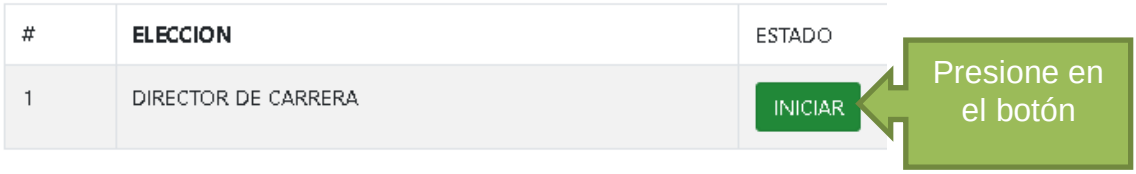
2.1. Usuario Jurado

a) Autenticación de usuario

El usuario(jurado) deberá ingresar su número de carnet de identidad y una contraseña que será proporcionada por el administrador.

b) Iniciar Mesa

Una vez que ingresó al Sistema deberá iniciar votación, esta acción requiere presionar sobre el botón Iniciar.

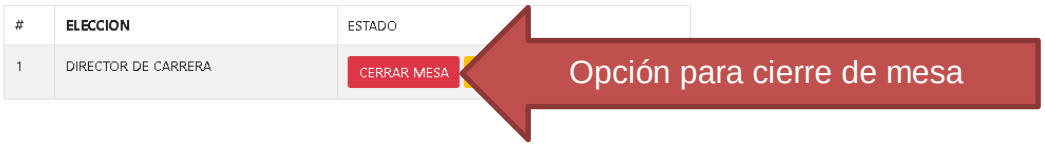


c) Habilitar votante

El sistema otorgara al jurado la opcion para habilitar a los votantes. Debera realizar la busqueda mediante el numero de cedula de identidad y habiltar al votante que se presente en la mesa.

d) Cerrar mesa de sufragio.

Una vez concluida las 8 horas, prosigue cerrar la mesa.



Estas son las tareas fundamentales que el Usuario Jurado deberá realizar en el sistema.

2.2. Usuario Votante

a) Autenticación de usuario

Una vez habilitado por el jurado para emitir el voto, deberá ingresar su número de identidad para acceder al sistema.

b) Emisión de voto

El sistema lo re direccionará al panel de votación donde el votante deberá emitir el voto presionando sobre el botón Votar.

c) Confirmar voto

El sistema pedirá la confirmación de la opción elegida o la posibilidad de cancelar y cambiar de decisión.

d) Cierre de sesión

Una vez que se realice la confirmación de voto, el sistema automáticamente cerrara sesión y no permitirá que vuelva a ingresar al sistema.

3. Conclusión

El sistema cuenta una interfaz fácil para realizar la emisión de votos, evitando dificultad en la interacción con el usuario.