Universidad Internacional de La Rioja

Escuela Superior de Ingeniería y Tecnología

Máster Universitario en Ciberseguridad

Desarrollo de software de almacenamiento de resultados electorales con tecnología Blockchain

Trabajo fin de estudio presentado por:	David Leonardo Espinosa Ospina Darwin Alfonso Rosas Quiroz Sebastián Guevara Sánchez
Tipo de trabajo:	Desarrollo de software
Director/a:	Dr. María Yoldi Sangüesa
Fecha:	16/07/2024

Resumen

Este proyecto desarrolla e implementa un software para almacenar resultados electorales usando tecnología Blockchain, destacando la seguridad y descentralización debido a la sensibilidad de la información electoral. Blockchain garantiza inmutabilidad y transparencia de los datos mediante bloques interconectados criptográficamente, ideales para datos seguros y verificables. El software almacena resultados de votaciones, incluyendo candidatos y votos, cumpliendo con estándares legales y técnicos para asegurar datos precisos y confiables. Los beneficios de Blockchain incluyen seguridad, transparencia, descentralización, escalabilidad, eficiencia y reducción de costos. Utiliza herramientas como Solidity, C++, Java, Python y JavaScript para crear aplicaciones seguras. El desarrollo se realizó con metodologías ágiles como SCRUM. El análisis económico muestra que el proyecto es viable y rentable, con un costo total de $125,794.31 y costos operativos anuales de $45,800.00. El retorno de inversión proyectado es del 169% en cinco años.

Palabras clave: Blockchain, Seguridad, Criptografía, Python, MongoDB.

Abstract

This project develops and implements software to store electoral results using Blockchain technology, emphasizing security and decentralization due to the sensitivity of electoral information. Blockchain ensures data immutability and transparency through cryptographically interconnected blocks, making it ideal for secure and verifiable data. The software stores voting results, including candidates and votes, adhering to legal and technical standards to ensure accurate and reliable data. The benefits of Blockchain include security, transparency, decentralization, scalability, efficiency, and cost reduction. It uses tools like Solidity, C++, Java, Python, and JavaScript to create secure applications. The development was conducted using agile methodologies like SCRUM. The economic analysis shows that the project is viable and profitable, with a total cost of $125,794.31 and annual operating costs of $45,800.00. The projected return on investment is 169% over five years.

Keywords: Blockchain, Security, Cryptography, Python, MongoDB.

Índice de contenidos

Índice de ilustraciones

Índice de tablas

Organización del trabajo en grupo

El trabajo de fin de máster se divide en 2 partes importantes, el desarrollo del proyecto de almacenamientos de resultados electorales con tecnología Blockchain y el desarrollo de la memoria del TFE.

Desarrollo del software:

- Planteamiento de casos de uso – Se analiza el problema de la metodología usada actualmente y las ventajas que se generarán al implementar el proyecto.
- Diseño de interfaces y datos – Se identifica que tipo de información se puede recopilar del proceso electoral y la información que se tendrá que almacenar en la generación de los bloques en el blockchain.
- Implementación de código fuente- Se desarrollo el código utilizando buenas prácticas de desarrollo y de seguridad.
- Pruebas unitarias y de integración - Se realizan pruebas al código como al funcionamiento, y validando la no vulnerabilidad del proyecto.

Desarrollo del Trabajo:

- Introducción – Se investigo la problemática con la que se cuenta, la actual forma de recopilación de los resultados en los procesos electorales. y nos brinda un mayor conocimiento de cómo se puede abordar y encontrar una mejor solución al problema principal que sería la vulnerabilidad de los resultados.
- Estado del Arte – En este apartado identificamos diferentes puntos de vista de los miembros del trabajo y de otros autores de la tecnología usada, y así poder encontrar nuevas tecnologías la cuales se pondrán en práctica en el proyecto para mejorar en la seguridad del proceso electoral.
- Objetivos y Metodología de trabajo – Nos planteamos el Objetivo general y los específicos, para poder medir la mejora de los resultados, y tener un enfoque más realista de la mejora del proceso.

- Desarrollo especifico de la contribución – Desarrollamos los diferentes componentes del proyecto como la seguridad de la información almacenada y gestionada por medio de Blockchain.
- Conclusiones y trabajo a futuro – Es el segmento en donde plasmamos los resultados obtenidos durante toda esta etapa del proyecto tanto como del desarrollo del software y la mejora del flujo del proceso electoral.

Partes que aborda el TFE, distribución y estructura de la memoria

Tabla 1 - Organización del trabajo en grupo.

Organización del trabajo en grupo - Desarrollo de la memoria	
Apartado de la memoria	**Responsables**
Introducción	Darwin Rosas
Marco Teórico	Sebastián Guevara
Motivación	David Espinosa
Planteamiento del problema	Sebastián Guevara
Estructura del trabajo	David Espinosa
Estado del arte	Darwin Rosas, Sebastián Guevara
Objetivos concretos y metodología de trabajo	Darwin Rosas, Sebastián Guevara, David Espinosa
Desarrollo especifico de la contribución	Darwin Rosas, Sebastián Guevara, David Espinosa
Creación de interfaces	Darwin Rosas

Estructura de Base de Datos	David Espinosa
Creación del JSON con la información	Sebastián Guevara
Pruebas de software	Sebastián Guevara, David Espinosa
Análisis económico	David Espinosa
Metodología de Análisis Económico	David Espinosa
Costos Iniciales de Desarrollo	David Espinosa, Sebastián Guevara
Conclusiones y trabajo futuro	Darwin Rosas, Sebastián Guevara
Referencias bibliográficas	Darwin Rosas, Sebastián Guevara, David Espinosa
Anexo A	Darwin Rosas, Sebastián Guevara, David Espinosa

Objetivo del TFE desde el punto de vista de la adquisición de conocimientos

El presente proyecto tiene como objetivo el desarrollo y la implementación de un software para el almacenamiento de resultados electorales utilizando la tecnología Blockchain. La seguridad y la descentralización son puntos clave en este proceso, ya que la información electoral es extremadamente sensible y debe ser protegida contra ataques y manipulaciones. Blockchain, como tecnología emergente, ofrece una solución robusta a estos desafíos al proporcionar un sistema que garantiza la inmutabilidad y la transparencia de los datos almacenados.

Blockchain es una tecnología que permite almacenar información en bloques interconectados mediante criptografía, creando una cadena de datos que es prácticamente inalterable sin que se detecten cambios en todos los bloques subsiguientes. Esta característica hace que Blockchain sea ideal para aplicaciones que requieren altos niveles de seguridad y veracidad de los datos, como el almacenamiento de resultados electorales.

En este proyecto, se utiliza Blockchain para almacenar los resultados de las votaciones, incluyendo información crítica como el nombre de los candidatos y los resultados registrados. Esta información se almacena de manera que cumple con los estándares legales y técnicos del lugar donde se realizan las elecciones, asegurando así que los datos sean precisos y confiables para auditorías y otros procesos administrativos.

Entre los beneficios más destacados de Blockchain se encuentran la seguridad, la transparencia, la descentralización, la escalabilidad, la eficiencia y la reducción de costos

La seguridad se logra a través de mecanismos criptográficos avanzados que aseguran que la información no pueda ser alterada sin que se detecte. La transparencia es garantizada por la naturaleza pública de la cadena de bloques, que permite a cualquier parte interesada verificar los datos almacenados. La descentralización reduce el riesgo de fallos y ataques, ya que no existe un único punto de fallo. La escalabilidad permite que el sistema maneje grandes volúmenes de datos sin degradar el rendimiento. La eficiencia y la reducción de costos se derivan de la eliminación de intermediarios y la automatización de procesos.

El software desarrollado en este proyecto utiliza herramientas y lenguajes de programación específicos para Blockchain, como Solidity para contratos inteligentes en Ethereum, así como

otros lenguajes como C++, Java, Python y JavaScript. Estas herramientas permiten la creación de aplicaciones robustas y seguras que pueden manejar la complejidad del almacenamiento de datos electorales.

El desarrollo de este software se llevó a cabo siguiendo metodologías ágiles como SCRUM, que aseguran una implementación eficiente y adaptable a cambios. El proceso de desarrollo incluyó la adquisición de nombres de candidatos y votos, la creación de un hash criptográfico para asegurar la información, y la organización de estos datos en formato JSON que luego se almacena en una base de datos NoSQL.

El análisis económico del proyecto muestra que la implementación de Blockchain en el almacenamiento de resultados electorales es no solo técnicamente viable, sino también económicamente rentable. El costo total estimado para el desarrollo del software, incluyendo un factor de riesgo del 30% y los impuestos, asciende a $125,794.31, con costos operativos anuales proyectados en $45,800.00. El retorno de inversión (ROI) proyectado del 169% en un período de cinco años sugiere que la inversión es altamente rentable, recuperando más del costo inicial en beneficios económicos y operativos.

Mecanismos de coordinación empleados

Para el desarrollo de nuestro Trabajo de Fin de Estudios (TFE) grupal sobre la implementación de Blockchain en sistemas electorales, empleamos diversos mecanismos de coordinación y comunicación que reflejan la realidad del entorno laboral actual.

- **Herramientas de Comunicación**

Plataforma Campus UNIR

Utilizamos esta plataforma para acceder a recursos educativos y mantenernos al día con las fechas de entrega, asi mismo nos ayudó a coordinar las reuniones, plazos y eventos importantes del proyecto, asegurando que todos estuvieramos al tanto de las fechas de entrega.

WhatsApp

Empleamos WhatsApp para la coordinación de reuniones y asegurar que todos los miembros del equipo estuvieran alineados y puntuales en las entregas de los hitos definidos.

Reuniones Virtuales

Realizamos reuniones virtuales en Jitsi y Google Meet para discutir el progreso del proyecto, asignar tareas y resolver problemas en tiempo real.

Versionamiento de Código

Empleamos Github para el control de versiones del código fuente, permitiendo un desarrollo colaborativo y ordenado del software.

Compartición de Documentos

Google Drive nos permitió a almacenar y compartir documentos en tiempo real, asegurando que todos los miembros del equipo tuvieran acceso a la información más reciente.

Revisión y Retroalimentación

Implementamos un ciclo regular de revisión y retroalimentación mediante comentarios en Google Docs y reuniones de revisión, garantizando la calidad y coherencia del trabajo entregado.

Glosario

- **Blockchain:** Tecnología que permite almacenar información en bloques interconectados mediante criptografía, creando una cadena de datos prácticamente inalterable sin que se detecten cambios en todos los bloques subsiguientes. Es ideal para aplicaciones que requieren altos niveles de seguridad y veracidad de los datos.

- **Seguridad:** Medidas y protocolos utilizados para proteger la información contra accesos no autorizados, alteraciones y ataques, asegurando que los datos sean confidenciales, íntegros y disponibles cuando se necesiten.

- **Ataques:** Intentos malintencionados de acceder, alterar, robar o destruir información y sistemas. En el contexto de Blockchain, estos pueden incluir ataques de red, ataques de doble gasto, y otros métodos de compromiso de la integridad del sistema.

- **Resultados electorales:** Datos que reflejan el número de votos obtenidos por cada candidato o propuesta en una elección. Es crucial que estos resultados sean precisos, verificables y estén protegidos contra manipulaciones.

- **Transparencia:** La capacidad de un sistema para permitir que sus procesos y datos sean visibles y verificables por las partes interesadas, asegurando así confianza y responsabilidad.

- **Descentralización:** Estructura de un sistema donde no existe un único punto de control o fallo, distribuyendo las responsabilidades y el poder de decisión entre múltiples nodos o participantes.

- **Escalabilidad**: La capacidad de un sistema para manejar un aumento en la carga de trabajo o la demanda de manera eficiente, sin degradar el rendimiento.

- **Eficiencia**: La habilidad de un sistema para maximizar la productividad con el menor uso posible de recursos, tiempo y esfuerzo.

- **Almacenamiento:** Proceso y tecnología utilizada para guardar datos de manera segura y accesible para su posterior recuperación y uso.

- **Criptografía**: Ciencia y práctica de proteger la información mediante el uso de códigos y cifrados, asegurando que solo las partes autorizadas puedan acceder y modificar los datos.

- **Votaciones:** Proceso mediante el cual los electores expresan sus preferencias sobre candidatos o propuestas en una elección, que posteriormente se convierten en resultados electorales.

- **Tecnología**: Conjunto de conocimientos y herramientas utilizados para diseñar, crear y mejorar productos, servicios y procesos.

- **Desarrollo de software:** Proceso de diseñar, crear, probar y mantener programas y aplicaciones que cumplen con las necesidades específicas de los usuarios.

- **Integridad de datos:** La exactitud y consistencia de los datos almacenados y transmitidos, asegurando que no han sido alterados o corrompidos.

- **Auditoría:** Proceso de revisión y evaluación sistemática de registros y actividades para asegurar el cumplimiento de estándares, regulaciones y políticas establecidas.

- **Procesos administrativos**: Conjunto de actividades y procedimientos organizativos utilizados para gestionar y operar una entidad, asegurando que se alcancen los objetivos establecidos.

- **NoSQL:** Tipo de sistema de gestión de bases de datos que no utiliza el modelo de tabla relacional tradicional, permitiendo un almacenamiento y recuperación de datos más flexible y escalable.

- **JSON**: Formato de texto ligero para el intercambio de datos, fácil de leer y escribir para los humanos y de interpretar y generar para las máquinas.

- **Inmutabilidad:** Característica de un sistema donde los datos, una vez escritos, no pueden ser alterados ni borrados, garantizando su permanencia y fiabilidad.

- **Herramientas de programación:** Conjunto de aplicaciones y lenguajes utilizados por los desarrolladores para escribir, probar y mantener software.

- **Metodologías ágiles:** Enfoques de desarrollo de software que promueven la flexibilidad, la colaboración y la capacidad de respuesta rápida a los cambios mediante ciclos de trabajo iterativos e incrementales.

- **ROI (Retorno de Inversión):** Métrica financiera que evalúa la rentabilidad de una inversión comparando el beneficio neto obtenido con el costo total de la inversión.

- **Pruebas de software:** Proceso de evaluar y verificar que un software funcione como se espera, identificando y corrigiendo errores y defectos.

- **Interfaces de usuario:** Componentes gráficos y de interacción que permiten a los usuarios comunicarse y operar un sistema o aplicación de manera efectiva.
- **MongoDB:** Base de datos NoSQL orientada a documentos que almacena datos en formato BSON (una extensión de JSON), ofreciendo alta escalabilidad y flexibilidad para gestionar grandes volúmenes de datos.

1. Introducción

En la era digital, la confianza en los sistemas electorales es fundamental para la estabilidad de las democracias modernas. La manipulación de datos electorales y la falta de transparencia en el proceso de votación han sido preocupaciones recurrentes en todo el mundo. Estos desafíos han llevado a la búsqueda de soluciones tecnológicas que puedan garantizar la integridad, la seguridad y la transparencia de los resultados electorales. En este contexto, la tecnología Blockchain se presenta como una solución prometedora.

Blockchain, o cadena de bloques, es una tecnología de registro distribuido que permite almacenar datos de manera segura y transparente. Cada bloque en la cadena contiene un registro de transacciones, y estos bloques están interconectados de manera que cualquier cambio en un bloque afecta a todos los bloques posteriores. Esta característica hace que la información almacenada en una Blockchain sea inmutable y resistente a manipulaciones (Swan, 2015). La descentralización inherente de Blockchain, donde la información no está controlada por una única entidad, sino que se distribuye a través de una red de nodos, añade una capa adicional de seguridad y confianza (Nakamoto, 2008).

La implementación de la tecnología Blockchain en los sistemas electorales plantea una serie de desafíos y consideraciones legales que deben ser cuidadosamente analizados para asegurar su viabilidad y conformidad con el marco legal existente. A continuación, se presenta un análisis de los aspectos legales en España, Colombia y a nivel internacional, incluyendo referencias y citas de legislaciones y estudios relevantes.

El objetivo principal de este proyecto es desarrollar e implementar un software para el almacenamiento de resultados electorales utilizando tecnología Blockchain. La importancia de este proyecto radica en su capacidad para abordar los problemas críticos de seguridad y manipulación de datos que han plagado los sistemas electorales tradicionales. Al utilizar Blockchain, se puede garantizar que los resultados electorales sean precisos, inmutables y fácilmente verificables por cualquier parte interesada (Nofer et al., 2017).

1.1. MARCO TEÓRICO

1.1.1. Python

Python es un lenguaje de programación creado por Guido van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses “Monty Python”. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible. (Raúl Gonzáles Duque, 2011).

Python es un lenguaje de programación potente y fácil de aprender. Tiene estructuras de datos de alto nivel eficientes y un simple pero efectivo sistema de programación orientado a objetos. La elegante sintaxis de Python y su tipado dinámico, junto a su naturaleza interpretada lo convierten en un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en muchas áreas, para la mayoría de las plataformas (Python Software Fundation, 2001).

La característica de tipado dinámico se refiere a que no es necesario declarar el tipo de dato que va a contener una determinada variable, sino que su tipo se determinará en tiempo de ejecución según el tipo del valor al que se asigne, y el tipo de esta variable puede cambiar si se le asigna un valor de otro tipo (Raúl Gonzáles Duque, 2011).

“Python ha sido parte importante de Google desde el principio, y lo sigue siendo a medida que el sistema crece y evoluciona. Hoy día, docenas de ingenieros de Google usan Python y seguimos buscando gente diestra en este lenguaje”. Peter Norvig, director de calidad de búsquedas de Google Inc. (Andrés Marzal, 2014).

“Python juega un papel clave en nuestra cadena de producción. Sin él, un proyecto de la envergadura de «Star Wars: Episodio II» hubiera sido muy difícil de sacar adelante. Visualización de multitudes, proceso de lotes, composición de escenas. . . Python es lo que lo une todo.” Tommy Brunette, director técnico senior de Industrial Light & Magic. (Andrés Marzal, 2014).

“Python está en todas partes de Industrial Light & Magic. Se usa para extender la capacidad de nuestras aplicaciones y para proporcionar la cola que las une. Cada imagen generada por

computador que creamos incluye a Python en algún punto del proceso.” Philip Peterson, ingeniero principal de I+D de Industrial Light & Magic. (Andrés Marzal, 2014).

1.1.2. MongoDB

NoSQL o "No solamente SQL" (Not Only SQL) es un término acuñado por Carlo Strozzi en 1998 y nuevamente retomado por Eric Evans en 2009 y se refiere a un conjunto de bases de datos que se diferencian en gran parte de las bases de datos convencionales, en características tanto de uso como de implementación. (Yohan Graterol, 2014).

“Es imposible para un sistema distribuido garantizar simultáneamente las siguientes tres características:

- Consistency (Consistencia): todos los nodos ven la misma data al mismo tiempo.
- Availability (Disponibilidad): una garantía de que todos los requerimientos recibirán una respuesta de que el requerimiento fue exitoso o fallido.
- Partition Tolerance (Tolerancia a la Partición): el sistema continúa operando a pesar de la pérdida arbitraria de mensajes, o la falla de parte del sistema.” (Eric Brewer, 2000).

MongoDB es un sistema de bases de datos no relacionales, multiplataforma e inspirada en el tipo de bases de datos documental y clave/valor, su nombre proviene del término en inglés "humongous". Está liberada bajo licencia de software libre, específicamente GNU AGPL 3.0. MongoDB usa el formato BSON (JSON Compilado) para guardar la información, dando la libertad de manejar un esquema libre. Este motor de bases de datos es uno de los más conocidos y usados, pudiéndolo comparar en popularidad con MySQL en el caso de las bases de datos relacionales. (Yohan Graterol, 2014).

1.1.3. Pymongo

Pymongo es una distribución de Python que proporciona herramientas para trabajar con MongoDB, es la forma más preferida de comunicarse con la base de datos MongoDB desde Python.

Para poder conectarte a MongoDB con Python, necesitas instalar el paquete del controlador PyMongo. En Python, la mejor práctica es crear lo que se conoce como un "entorno virtual" en el cual instalar tus paquetes. Esto los aísla limpiamente de cualquier paquete "del sistema" que tengas instalado y ofrece la ventaja adicional de no requerir privilegios de root para instalar paquetes adicionales de Python. La herramienta para crear un "entorno virtual" se llama virtualenv. (Niall O'Higgins. 2011)

1.1.4. Flask

Flask se destaca de otros marcos porque permite a los desarrolladores tomar el control y tener control creativo total de sus aplicaciones. Quizás hayas escuchado la frase "luchar contra el marco" antes. Esto sucede con la mayoría de los marcos cuando decides resolver un problema con una solución que no es la oficial. Podría ser que quieras utilice un motor de base de datos diferente, o tal vez un método diferente para autenticar a los usuarios. (Miguel Grinberg, 2014).

Flask es un marco web Python liviano que proporciona herramientas y funciones útiles para crear aplicaciones web en el lenguaje Python.

Al desarrollar una aplicación web, es importante separar la lógica empresarial de la lógica de presentación. La lógica empresarial es lo que maneja las solicitudes de los usuarios y se comunica con la base de datos para crear una respuesta adecuada. (DigitalOcean, 2024). En Flask, puedes usar el lenguaje de plantillas Jinja para generar plantillas HTML. Una plantilla es un archivo que puede contener contenido tanto fijo como dinámico. Cuando un usuario solicita algo de tu aplicación (como una página de índice o una página de inicio de sesión), Jinja te permite responder con una plantilla HTML donde puedes usar muchas características que no están disponibles en HTML estándar, como variables, if declaraciones, for bucles, filtros y herencia de plantillas. Estas características te permiten escribir de manera eficiente páginas HTML fáciles de mantener. Jinja también escapa automáticamente el HTML para evitar ataques de secuencias de comandos entre sitios (XSS). (DigitalOcean, 2024).

1.1.5. JavaScript

JavaScript es un lenguaje de programación de scripts (secuencia de comandos) orientado a objetos. Como cualquier otro lenguaje de programación, JavaScript tiene algunas características especiales: sintaxis, modelo de objetos, etc. Claramente, cualquier cosa que diferencia un lenguaje de otro. Además, descubrirás rápidamente que JavaScript es un lenguaje relativamente especial en su acercamiento a las cosas. Esta parte es esencial para cualquier principiante de programación e incluso para aquellos que ya conocen un lenguaje de programación debido a que las diferencias con otros lenguajes de programación son numerosas. (Rafael Menéndez, 2008).

La inigualable popularidad de JavaScript como lenguaje de programación de aplicaciones web se ha extendido a otras aplicaciones y otros entornos no relacionados con la web. Herramientas como Adobe Acrobat permiten incluir código JavaScript en archivos PDF. Otras herramientas de Adobe como Flash y Flex utilizan ActionScript, un dialecto del mismo estándar de JavaScript. Photoshop permite realizar pequeños scripts mediante JavaScript y la versión 6 de Java incluye un nuevo paquete (denominado javax.script) que permite integrar ambos lenguajes. Por último, aplicaciones como Yahoo Widgets (http://widgets.yahoo.com/) y el Dashboard de Apple (http://www.apple.com/downloads/dashboard/) utilizan JavaScript para programar sus widgets. (Rafael Menéndez, 2008).

1.1.6. Blockchain

Blockchain permite almacenar información en bloques interconectados mediante criptografía, creando una cadena inmutable de información que garantiza la integridad de los datos (Swan, 2015). La descentralización es una característica clave, ya que distribuye la información a través de una red de nodos, aumentando la resistencia a ataques y fallos (Yli-Huumo et al., 2016). Existen diferentes tipos de Blockchain, como pública, privada y de consorcio, cada una con ventajas y desventajas específicas en términos de seguridad, velocidad y control de acceso (Buterin, 2014).

“Blockchain es una tecnología que permite almacenar información en bloques interconectados mediante criptografía, creando una cadena inmutable de información (Swan, 2015). Esta estructura garantiza que cualquier alteración en un bloque invalide todos los

bloques subsiguientes, haciendo prácticamente imposible modificar la información sin ser detectado (Yli-Huumo et al., 2016)."

1.1.7. Herramientas y Lenguajes Usados en Blockchain

“Para desarrollar aplicaciones Blockchain se utilizan herramientas y lenguajes como Solidity para contratos inteligentes en Ethereum, así como C++, Java, Python y JavaScript” (Dannen, 2017).

1.1.8. Almacenamiento de Resultados Electorales

“El almacenamiento de resultados electorales requiere altos niveles de seguridad y transparencia. Los métodos tradicionales centralizados son vulnerables a manipulaciones y ataques cibernéticos, mientras que Blockchain ofrece una mayor transparencia y seguridad” (Hjalmarsson et al., 2018).

1.1.9. Criptografía

“La criptografía es esencial para la seguridad en Blockchain. Algoritmos de hashing como SHA-256 son fundamentales para crear identificadores únicos para cada bloque de información, asegurando la integridad y autenticidad de los datos” (Katz y Lindell, 2007).

1.1.10. SHA-256

Un hash de n bits es un mapa de mensajes de longitud arbitraria a valores hash de n bits. Un hash criptográfico de n bits es un hash de n bits que es unidireccional1 y resistente a colisiones.2 Estas funciones son primitivas criptográficas importantes que se utilizan para cosas como firmas digitales y protección con contraseña.

Los hashes populares actuales producen valores hash de longitud n = 128 (MD4 y MD5) y n = 160 (SHA-1), y por lo tanto no puede proporcionar más de 64 u 80 bits de seguridad, respectivamente, contra ataques de colisión. Desde el objetivo del nuevo Advanced El estándar de cifrado (AES) ofrecerá, en sus tres tamaños de criptovariables, 128, 192, y 256 bits de seguridad, existe la necesidad de algoritmos hash complementarios que proporcionar niveles similares de seguridad mejorada.

La función de compresión SHA-256 opera en un bloque de mensajes de 512 bits y en un bloque de 256 bits. valor hash intermedio de bits. Es esencialmente un algoritmo de cifrado de

bloques de 256 bits que cifra el valor hash intermedio utilizando el bloque de mensajes como clave. Por lo tanto, hay dos componentes principales para describir: (1) la función de compresión SHA-256, y (2) el calendario de mensajes SHA-256 (Andrew W.Appel, 2015)

1.2. Motivación

El presente proyecto aborda el problema crítico de la seguridad y transparencia en la gestión de resultados electorales. Actualmente, los sistemas tradicionales utilizados para registrar y almacenar los resultados de las votaciones presentan vulnerabilidades, como manipulaciones internas, ciberataques y errores humanos. Estas vulnerabilidades no solo comprometen la integridad del proceso electoral, sino que también disminuyen la confianza pública en los resultados, afectando la legitimidad de las elecciones.

La motivación principal de este estudio radica en la necesidad urgente de mejorar la seguridad y la transparencia en los procesos electorales. La tecnología Blockchain se presenta como una solución innovadora y robusta para abordar estos desafíos. Al utilizar Blockchain, se puede garantizar que los datos electorales sean inmutables y verificables por todas las partes interesadas, lo que incrementa significativamente la confianza pública en los sistemas electorales.

Investigaciones previas han demostrado el potencial de Blockchain para transformar diversos sectores mediante la mejora de la seguridad y la transparencia de los datos. Sin embargo, su aplicación en el ámbito electoral aún es incipiente y requiere un análisis profundo y una implementación cuidadosa. Este proyecto se basa en estas investigaciones previas para explorar cómo Blockchain puede ser integrado eficazmente en los sistemas electorales y proporcionar un marco sólido para futuras implementaciones.

La integridad y seguridad en los sistemas electorales es un desafío crucial, evidenciado por las denuncias y preocupaciones en varios países. En Venezuela, las elecciones presidenciales de 2024 están marcadas por la desconfianza y el riesgo de fraude debido a la manipulación política y las inhabilitaciones de candidatos opositores. Este escenario agrava la transparencia electoral en un contexto de conflicto territorial y represión de derechos humanos (Guzmán, 2023).

En Estados Unidos, aunque las denuncias de fraude electoral son comunes, estudios como los del Centro Brennan y la Heritage Foundation demuestran que el fraude es extremadamente

raro, gracias a robustas medidas de seguridad y auditoría. Aun así, la desinformación persiste, generando desconfianza en el proceso electoral (DW, 2024).

En Sudáfrica, la oposición denunció fraude electoral y planea acciones legales tras las recientes elecciones, reflejando la necesidad de sistemas más transparentes y seguros (France24, 2024).

En México, según el exsecretario general de la OEA, José Miguel Insulza, es difícil que ocurra un fraude significativo debido a la vigilancia estricta de los observadores internacionales y las medidas de seguridad implementadas en el proceso electoral (CNN Español, 2024).

En España, las denuncias sobre posibles fraudes en los resultados electorales de Junts fuera de Cataluña fueron desmentidas, atribuyendo las irregularidades a errores de comunicación de datos provisionales que no afectaron el resultado final (EFE Verifica, 2024).

Estas situaciones nos muestran un panorama global de la situación electoral en diversas naciones, donde se genera una urgencia de implementar un sistema electoral que garantice la seguridad y la transparencia, otorgando tranquilidad y confianza para desvirtuar cualquier situación de duda e incertidumbre. La tecnología Blockchain ofrece una solución innovadora, proporcionando inmutabilidad y verificabilidad en la gestión de datos electorales. Este proyecto se justifica por la necesidad de mejorar los sistemas electorales, basándose en investigaciones que validan el potencial de Blockchain para asegurar procesos críticos y restaurar la confianza pública.

1.3. Planteamiento del problema

El problema detectado es la vulnerabilidad de los sistemas actuales de gestión de resultados electorales frente a manipulaciones y ciberataques, comprometiendo la integridad y la transparencia del proceso electoral. Esta vulnerabilidad genera desconfianza en los ciudadanos y puede llevar a crisis de legitimidad en los gobiernos elegidos.

Para solucionar este problema, se propone el desarrollo e implementación de un sistema de almacenamiento de resultados electorales basado en tecnología Blockchain. Este sistema buscará garantizar la inmutabilidad y verificabilidad de los datos electorales, eliminando la posibilidad de manipulación y aumentando la transparencia del proceso.

La finalidad de este Trabajo de Fin de Estudios (TFE) es proporcionar un sistema seguro y transparente que pueda ser utilizado en futuras elecciones, asegurando así la integridad del proceso electoral y restaurando la confianza pública en los resultados.

1.4. Estructura del trabajo

Este trabajo se estructura en varios capítulos que abordan diferentes aspectos del proyecto:

- *Introducción*

Se presenta una visión general del proyecto, incluyendo la motivación, los objetivos y la estructura del trabajo.

- *Marco Teórico*

Se revisa la literatura existente sobre Blockchain, sistemas electorales, y tecnologías relacionadas como Python, MongoDB, y criptografía. También se aborda la historia y evolución de las votaciones universales y electrónicas.

- *Objetivos*

Se definen el objetivo principal y específicos del proyecto, detallando qué se espera lograr con la implementación del sistema basado en Blockchain.

- *Metodología*

Se describe el enfoque metodológico utilizado para desarrollar el sistema, incluyendo el diseño de interfaces de usuario, la configuración de bases de datos, y los procesos de encriptación y pruebas de software.

- *Aspectos Legales*

Se analiza el marco legal en España, Colombia y a nivel internacional, destacando las regulaciones y normativas que impactan la implementación de Blockchain en sistemas electorales.

- *Resultados*

Se presentan los resultados obtenidos durante el desarrollo del proyecto, incluyendo la creación de interfaces de usuario, la estructura de la base de datos, y las pruebas de encriptación y seguridad

.

- *Análisis Económico*

Se realiza un análisis económico detallado, evaluando los costos iniciales, los costos operativos anuales, y el retorno de inversión (ROI) proyectado.

- *Conclusiones*

Se resumen los hallazgos del proyecto, se discuten las implicaciones y se proponen áreas para futuras investigaciones.

- *Referencias*

Se listan todas las fuentes y referencias utilizadas a lo largo del trabajo, siguiendo las normas APA 7.

2. Estado del arte

Varios estudios han demostrado la efectividad de Blockchain para prevenir el fraude y mejorar la trazabilidad en el almacenamiento de resultados electorales. Implementaciones en Estonia y Suiza han mostrado mejoras significativas en la seguridad y confianza pública en el sistema electoral (Glaser, 2017).

Numerosos estudios han explorado el uso de Blockchain en el almacenamiento de resultados electorales, destacando su capacidad para ofrecer una solución más segura y transparente. Implementaciones en países como Estonia y Suiza han mostrado mejoras significativas en la seguridad y confianza pública (Glaser, 2017).

2.1. METODOLOGÍA DE DESARROLLO

El desarrollo de software utilizando Blockchain requiere metodologías ágiles como SCRUM para asegurar una implementación eficiente y segura. Es esencial una revisión exhaustiva de la literatura, la identificación de herramientas y lenguajes adecuados, y la realización de pruebas exhaustivas para validar la robustez del sistema desarrollado (Beck et al., 2017).

El desarrollo de software utilizando Blockchain requiere un enfoque especializado y el uso de metodologías ágiles como SCRUM para asegurar una implementación eficiente y segura (Beck et al., 2017).

2.2. HISTORIA DE LAS VOTACIONES UNIVERSALES

Los primeros sufragios que de los que hay documentación son las asambleas de la antigua Grecia y Roma 590ac la cual tenía algunas restricciones alguna de ellas era que solo los hombres podían participar en estas elecciones, luego este tipo de democracia fue creciendo en la edad media hasta volverse necesaria para cada localidad, pero el momento histórico de la revolución francesa 1793 fue el que amplio la posibilidad a democracias más participativas, luego de tener monarquías absolutistas donde solo la nobleza y la iglesia católica tenían poder político, con esta revolución llego la separación de la iglesia de la política pública, y llegaron los derechos de hombre junto a la posibilidad de escoger los representantes en los congresos estatales (J. W. Lamare, 1981, p.256-257).

Entre los años 1800 y los años 1900 le empezó a impulsar el derecho femenino de votación por políticas públicas, de los ejemplos más recientes de elección femenina de lideres políticos se dio en 1869 en estados unidos, en 1893 lograron el derecho las mujeres en nueva Zelanda, en 1895 en Australia pudieron votar las mujeres, posteriormente a los avances sociales anteriores, otros países se animaron a continuar con estas decisiones entre 1906 y 1921Finlandia, Noruega, Dinamarca y Suecia, habilitaron el sufragio femenino. Los países que decidieron darle la oportunidad de votar a las mujeres luego de la segunda guerra mundial 1943 fue Italia, Bélgica y Francia (J. W. Lamare, 1981, p.256-257).

En Colombia se aprobó el voto femenino en 1954 durante la dictadura de Gustavo Rojas Pinilla, y algunos de los países donde no se avanzado en la democracia femenina son Afganistan, Arabia Saudita, Kuwait y Líbano (J. W. Lamare, 1981, p.256-257).

2.3. HISTORIA DE LAS VOTACIONES ELECTRÓNICAS

La votación electrónica se empezó a implementar a mediados de los 1990 con la llegada del software brasilero DRE, pero en el 2000 el partido estadounidense democrático de Arizona realizo por primera vez sus elecciones por medio de internet (Business Wire, 1999).

En estos últimos 20 años las votaciones de manera electrónica han ido en aumento, en el año 2013 más de 31 países empezaron a usar formas electrónicas de votaciones. (ACE Project, 2024).

DRE en ingles direct-recording electronic voting machine (DRE) en español voto electrónico recogido directo, es una aplicación que ofrece el servicio de intermediario entre el votante y la base de datos, donde se guarda el voto del elector, creada en Brasil en el año 1996 (United States Election Assistance Commission, 2005). Esta aplicación organiza en los votos y los muestra al final de cada tiempo de votación, imprime en pantalla los votos totales y los de cada individuo, esta información pude ser extraída en un disco duro y llevada a una central donde se hará el conteo 1996 (United States Election Assistance Commission, 2005). Este aplicativo DRE es muy amigable con el usuario, facilitándole la votación a personas con discapacidad.

En Brasil se ha implementado rápidamente el e-voting que es la votación electrónica, donde se empezó a realizar desde 1996 y en la actualidad se sigue usando, este software se sostiene en sistemas operativos Linux (Tribunal Superior Electoral, 2005). Esta herramienta guarda la información relacionada al voto y el votante, desde el año 2012 se creó la posibilidad de validación biométrica (Tribunal Superior Electoral, 2005). La DRE tiene su código fuente a disposición de los partidos políticos que participaran en la elección con el interés de que haya transparencia, pero también ha sido criticado porque no responde con un papel que sirva de comprobante de votación.

En la india es otro país donde se ha implementado la aplicación DREs desde 1999 en algunas elecciones, en este país, tomaron la decisión de crear un pequeño recibo llamado VVPAT que sirve de comprobante de la votación, esta aplicación también tiene la particularidad de que el candidato puede ver las personas que votaron por el (India Today. New counting method for Assembly polls, 2008).

Estonia escogió la opción de que sus elecciones estatales se den por medio de internet, más del 30% de la votación total es hecha por internet, aun cuando tienen la posibilidad de votar en papel (e-Estonia, 2024). La arquitectura que se usa para el software es de llave publica, pero al final de la votación se borra la identidad del voto, para validar la identidad de la persona, se comprueba por medio de la SIM del celular, En estonia los ciudadanos tienen una tarjeta de identificación con chip integrado que sirve para que ellos realicen sus votos sobre la política de su país.

En estonio han surgido algunas críticas por posibles vulnerabilidades en el código fuente de la aplicación de votaciones. (Tribunal Superior Electoral, 2005).

La aplicación Follow My Vote es un software de votación que utiliza almacenamiento en Blockchain y fue creada por una empresa con beneficios públicos en estados unidos, este software valida la identidad del votante y también tiene su código abierto al público, para mayor transparencia en su información, fue escrita en el lenguaje de programación C++ .

En el territorio noruego entre el 2011 y 2013 se empezó el proceso de digitalización de las votaciones parlamentarias, su proveedor de software fue Scytl que es una empresa española fundada en 2001, las personas que tuvieron interacción con el software informan que es muy parecido al sistema de estonia, este software noruego tuvo malas noticias relacionada a posibles ciberataques (J. Barrati-Esteve et al., 2012).

En el territorio de China tienen el sistema de votación electrónica con tecnología Blockchain Liu y Wang, el cual cuenta con una estructura centrada en la privacidad o el anonimato por parte de los votantes, también cuentan con llave publica y firmas digitales de cada voto, cada grupo de personas relacionadas al software tienen roles específicos que sirve para ver los resultados en tiempo real, si se tiene alto privilegio (Liu, Yi et al., 2017).

2.4. HISTORIA DE JAVASCRIPT Y ECMA

A principios de los años 90, la mayoría de los usuarios que se conectaban a Internet lo hacían con módems a una velocidad máxima de 28.8 kbps. En esa época, empezaban a desarrollarse las primeras aplicaciones web y, por tanto, las páginas web comenzaban a incluir formularios complejos. Con unas aplicaciones web cada vez más complejas y una velocidad de navegación tan lenta, surgió la necesidad de un lenguaje de programación que se ejecutara en el navegador del usuario. De esta forma, si el usuario no rellenaba correctamente un formulario, no se le hacía esperar mucho tiempo hasta que el servidor volviera a mostrar el formulario indicando los errores existentes. (Javier Eguíluz Pérez, 2008).

Brendan Eich, un programador que trabajaba en Netscape pensó que podría solucionar este problema adaptando otras tecnologías existentes (como ScriptEase) al navegador Netscape Navigator 2.0, que iba a lanzarse en 1995. Inicialmente, Eich denominó a su lenguaje LiveScript. Posteriormente, Netscape firmó una alianza con Sun Microsystems para el desarrollo del nuevo lenguaje de programación. Además, justo antes del lanzamiento Netscape decidió cambiar el nombre por el de JavaScript. La razón del cambio de nombre fue exclusivamente por marketing, ya que Java era la palabra de moda en el mundo informático y de Internet de la época. La primera versión de JavaScript fue un completo éxito y Netscape Navigator 3.0 ya incorporaba la siguiente versión del lenguaje, la versión 1.1. Al mismo tiempo, Microsoft lanzó JScript con su navegador Internet Explorer 3. JScript era una copia de JavaScript al que le cambiaron el nombre para evitar problemas legales. (Javier Eguíluz Pérez, 2008).

Para evitar una guerra de tecnologías, Netscape decidió que lo mejor sería estandarizar el lenguaje JavaScript. De esta forma, en 1997 se envió la especificación JavaScript 1.1 al organismo ECMA (European Computer Manufacturers Association).

ECMA ha publicado varios estándares relacionados con ECMAScript. En junio de 1997 se publicó la primera edición del estándar ECMA-262. Un año después, en junio de 1998 se realizaron pequeñas modificaciones para adaptarlo al estandar ISO/IEC-16262 y se creó la segunda edición. La tercera edición del estándar ECMA-262 (publicada en diciembre de 1999) es la versión que utilizan los navegadores actuales y se puede consultar gratuitamente en

http://www.ecma-international.org/publications/standards/Ecma-262.htm Actualmente se encuentra en desarrollo la cuarta versión de ECMA-262, que podría incluir novedades como paquetes, namespaces, definición explícita de clases, etc. ECMA también ha definido varios estándares relacionados con ECMAScript, como el estándar ECMA-357, que define una extensión conocida como E4X y que permite la integración de JavaScript y XML.

3. Objetivos concretos y metodología de trabajo

3.1. Objetivo general

Desarrollar e implementar un software de registro de resultados de votaciones con tecnología Blockchain.

3.2. Objetivos específicos

Investigar y seleccionar la infraestructura tecnológica adecuada para la implementación del software de almacenamiento de resultados electorales basado en Blockchain.

Establecer protocolos de cifrado y seguridad para proteger la privacidad y confidencialidad de los datos electorales almacenados en la Blockchain.

Desarrollar interfaces de usuario intuitivas y accesibles que permitan a los usuarios visualizar y verificar los resultados electorales almacenados en la Blockchain de manera transparente.

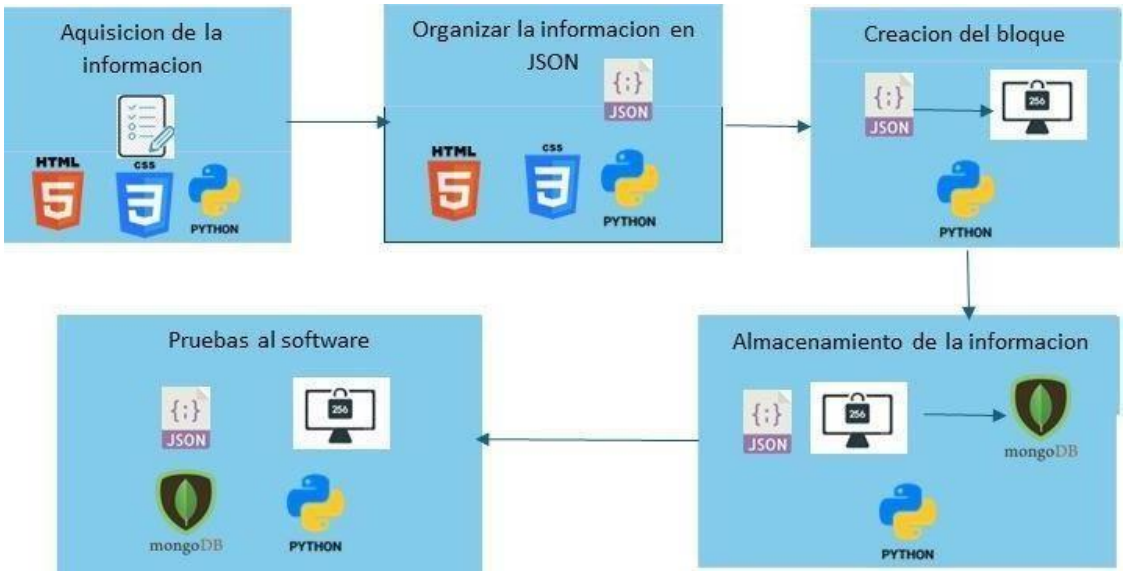
Realizar pruebas exhaustivas de funcionalidad, seguridad y rendimiento del sistema de almacenamiento de resultados electorales para garantizar su adecuado funcionamiento en condiciones reales de uso.

3.3. Metodología del trabajo

En esta parte del documento se describe la metodología para diseñar y desarrollar un software que funciona de manera web que consiste en almacenar información de elecciones regionales con tecnología Blockchain. Este proceso empieza con la adquisición del nombre de los candidatos junto a la cantidad de votos recibidos, después se crea un hash criptográfico con la información que se va a almacenar, y luego esta información se organiza en un JSON junto a un hash criptográfico para cuidar la legitimidad de los datos. Teniendo este JSON se crea el bloque y se almacena en una base de datos NoSql, se implementará diferentes lenguajes de programacion y tambien de etiquetas que son Python, javascript y html para la adquisicion de la informacion y tambien la creacion del bloque con el hash criptografico, tambien se usa la base de datos mongodb por medio de la libreria pymongo.

La metodología anterior se representa en la siguiente grafica.

Ilustración 1 - Metodología del flujo funcional del software



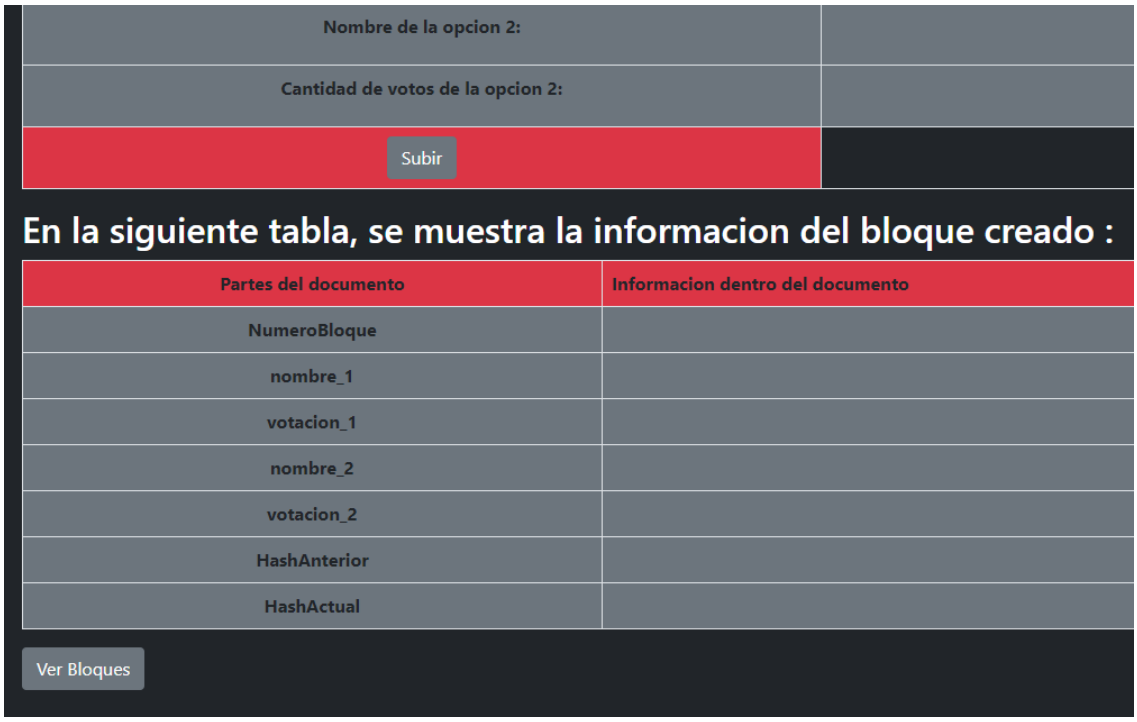
3.1. DISEÑO DE LAS INTERFACES DE USUARIO

Las interfaces creadas en esta investigación son amigables con el usuario y fáciles de utilizar, La interfaces inicial es la de subir información de votaciones, esta página web tendrá una barra de inicio en la parte superior de la pantalla, donde se puede navegar por la aplicación, y en la parte del medio un formulario donde se ingresan la información de las votaciones, esta información tendrá nombre de candidato y cantidad de votos, en la parte de abajo de la interface se puede ver una tabla que confirma la información que se subió a la base de datos y al final, un botón donde se puede acceder al listado de las cadenas de bloque creadas. La estructura del frontend de todo el software, está compuesto por HTML, CSS, bootstrap y Javascrip, los colores se escogieron con el fin de evitar el desgaste de la vista con colores brillantes de fond, por eso se escogieron de fondo colores opacos, también ayudan a economizar la energía del computador. El diseño de la interface se puede observar en la siguiente imagen.

Ilustración 2 - Registro de resultados de votaciones

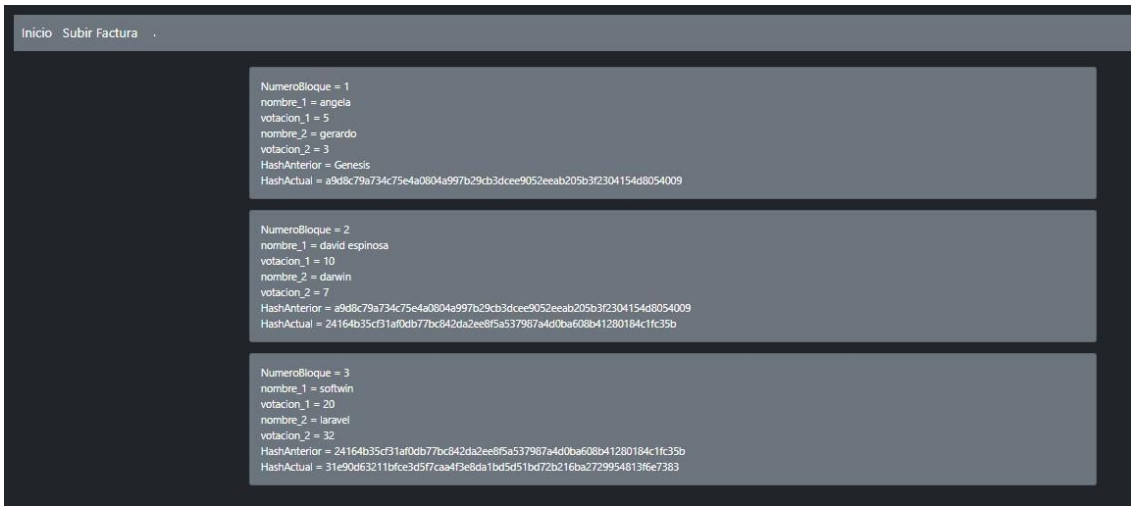
En la parte de abajo del software, se muestra los datos que ya se ingresaron a la base de datos, y al final de la tabla se habilita un botón para ver el bloque creado en la pestaña 2 de la página web.

Ilustración 3 - Registro de resultados de votaciones



La segunda interfaz que creamos en este software es el de ver bloques creados, el botón para acceder a esta página está en la página inicial, en la parte de abajo o también en la barra de inicio donde al dar click en “ver bloques” también te lleva a esta 2 pestaña de la aplicación, esta segunda parte de la aplicación web tendrá una barra de inicio en la parte superior, donde se podrá elegir devolverse, en esta página se vera la cadena de bloques donde cada bloque tiene la información de las votaciones relacionado a el nombre del candidato y su cantidad de votos, anexándole el hash del bloque anterior y el hash del bloque actual, las herramientas usadas en esta interfaces son HTML, CSS, bootstrap y Javascrip, los colores que escogimos fue con el fin de economizar energía de la pantalla de los computadores con colores opacos y evitar el cansancio que se da con fondos muy brillantes. como se muestra en la figura siguiente.

Ilustración 4 - Registro de resultados de votaciones

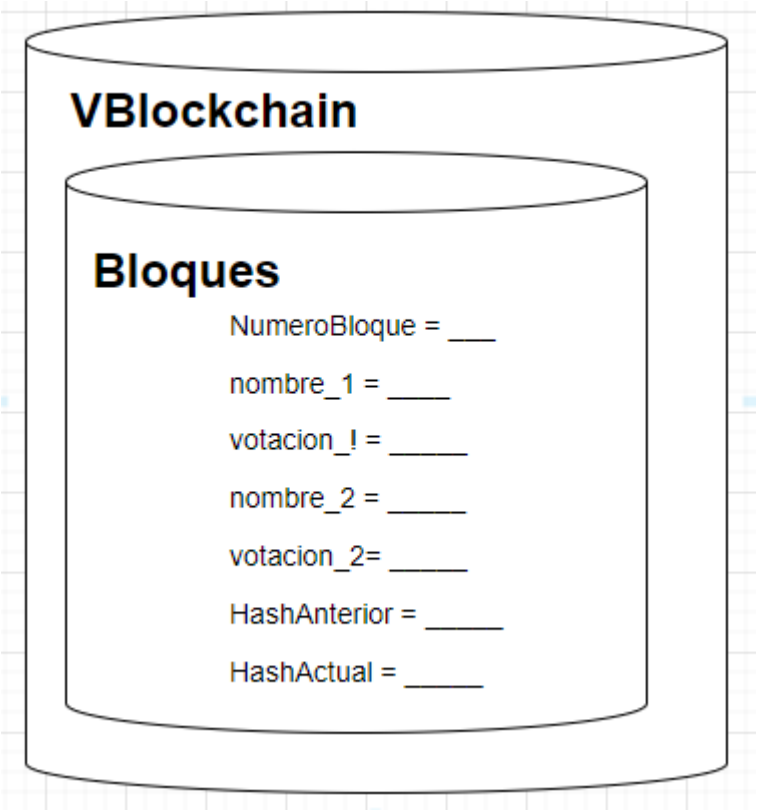


La integración entre las 2 interfaces de esta página web, consiste en los botones que tiene la barra de inicio, con estos botones, se puede pasar de una interfaz a la otra interfaz como se muestra en la siguiente grafica.

3.2. DISEÑO DE BASE DE DATOS

En este diseño de base de datos usaremos el motor NoSql llamado MongoDb el nombre de la base de datos que usaremos será “VBlockchain” y dentro de esa base de datos, tendremos la colección llamada “bloques” que es donde se guardara la información de los candidatos y sus votos. Esta colección llamada bloque, tendrá un ID que se guardara con todo el bloque, este id es autoincrementar cada vez que se cree una colección, cada bloque tendrá el nombre del candidato y su respetiva votación general, y al final tendrá el hash anterior, que junto al resto de datos servirá para crear el hash actual de cada bloque, toda esta información se guardara en forma de JSON y se envia a guardar en la base de datos por medio del lenguaje de programacion python. En la siguiente imagen se muestra un diagrama de base de datos donde nos muestra cómo se guardarán los datos.

Ilustración 5 - Diseño base de datos



3.3.INGRESO DE INFORMACIÓN DE VOTACIONES EN LA PÁGINA WEB

En este software se ingresarán los datos por medio de un formulario estos datos serán limitados, el nombre del candidato será un string y la cantidad de votos será un entero, los hashes criptográficos serán string también, primero se crea un JSON con toda esta información y posteriormente se guardará en la base de datos, la adquisición y organización de la información se hará en la parte del frontend por medio de HTML, CSS y Bootstrap con un componente de PYTHON en la parte del backend, como se muestra en la siguiente imagen.

Ilustración 6 - Registro de resultados de votaciones

3.4. ENCRIPTACIÓN DE LA INFORMACIÓN

La encriptación de la información se realizará con el lenguaje de programación PYTHON y su librería hashlib, de esta librería usaremos la opción SHA256 la cual nos servirá para convertir toda la información del bloque de datos en 1 hash criptográfico de 32 bits llamado "hash actual" que posteriormente se guardará junto a la información, se escogio el SHA256 por la dificultad de descencriptar esa variable encriptada.

Ilustración 7 - Registro de resultados de votaciones

3.5.CONFIGURACIÓN DE LA BASE DE DATOS MONGODB

La información luego de ser procesada por en lenguaje PYTHON, procedemos a usar la librería PyMongo para crear el insert y almacenarlo en la base de datos, antes de ejecutar el insert, se validará si este es el primer bloque creado, y si ese es el caso, se pondrá el texto "Genesis" en la variable hash anterior, del primer bloque creado. Dado el caso de que ya haya un primer bloque, se traerá el hash actual de ese, para ponerlo en el HashAnterior del nuevo bloque.

Ilustración 8 - Registro de resultados de votaciones

3.6. DISEÑO DE PRUEBAS DEL SOFTWARE

En esta parte del desarrollo comprobaremos la seguridad del software al momento de mantener la integridad de los datos almacenados tomando de ejemplo un cambio mínimo en los datos del bloque para ver cómo cambia el hash actual en los 2 ejemplos, lo que se hará en esta primer prueba es que se guardará un bloque de información y nos fijaremos en el hash actual que crea este bloque, luego se hará 1 cambio en 1 solo digito de todo el bloque y validaremos como el hash actual del bloque de prueba creado, cambiara en su totalidad, esto nos ayudara a identificar si hubo un cambio en la información.

En la segunda prueba que se hará al software, es ingresar 3 bloques de información, y capturaremos la imagen de los bloques creados, luego cambiaremos solo 1 dato de los bloques iniciales y en esta oportunidad validaremos que el hash actual del bloque final cambia totalmente, así el cambio de dato se haya realizado en los bloques iniciales, y de esta manera veremos cómo se afecta toda la cadena de bloques, con un solo digito cambiado.

La tercera prueba y ultima que se realizara es la de la modificación de la totalidad de los datos en 1 bloque, con el fin de demostrar que cuando los cambios realizados de manera no autorizada así sean de todos los datos, cambiara el hash criptográfico actual del último bloque creado y de esta manera nos daremos cuenta de que hubo una modificación no deseada, por la diferencia en los dígitos del hash criptográfico.

Las herramientas que se usaran en el diseño y desarrollo de este software funcionaran así:

3.6.1. Lenguaje de programación Python

con este lenguaje se crea toda la estructura del software desde la adquisición de la información hasta la creación del bloque de datos que sera enviado a almacenarse en la base de datos, como la gran parte del software esta creado con python, se usara librería flask para ejecutar la aplicación de manera web, con el comando “python nombredelmain” y en la página de “ver bloques” se imprimen todos los bloques de datos que se han creado por medio de una consulta a la base de datos, organizada enel lenguaje python. Las librerías del lenguaje python que se usaran en este proyecto son fask, hashlib, pymongo. La librería flask sirve para ejecutar el main del software y ejecutar la aplicación, la librería hashlib ayuda a crear el hash criptografico que tendra cada bloque de informacion para poder tener la seguridad de un Blockchain, y la librería pymongo nos facilitara la comunicación del software con la base de datos mongodb que recibe informacion JSON y es una base de datos NoSql.

3.6.2. Lenguaje de programación Javascript

Se usa en esta oportunidad con el fin de mostrar en pantalla los diferentes bloques creados en la página “ver bloques” del software, ahí se vera la información de cada bloque, el número del bloque, los nombres de los participante de la eleccion y la cantidad de votos que saco cada uno, organizado desde el más bloque más antiguo al bloque más reciente, la impresión de esta información dentro de la pestaña “verbloques.html” se realiza con una consulta a la base de datos por medio de python y luego javascript la recibe con un for que recorre la informacion recibida en un JSON y va consultando cada parte del JSON para luego irla publicando en la página con el codigo de etiqueta HTML.

3.6.3. Lenguaje de etiquetado HTML

Se utiliza en este desarrollo para darle estructura a el frontend del software el cual este compuesto por 2 pestañas de página web, una para ingresar información de elecciones compuesta por nombre de candidato junto a la cantidad de votos logrados, que luego se guardara en Blockchain y la otra pestaña que sirve para ver los bloques de información creados en orden de creación.

El orden de la pestaña por donde se ingresa la información está compuesta por una barra de opciones, donde se encuentra la posibilidad de cambiar de pestaña web, debajo de esta barra de tareas, hay un formulario compuesto por una etiqueta tipo from que recopila diferentes input en este caso cada input es de tipo texto y al final del formulario se encuentra un input de tipo submit que al darle click el envía toda la información puesta en los input de tipo text y los lleva a el backend que esta creado en Python, en esta misma página pero en la parte de abajo se encuentra otro formulario, donde los input mostraran el bloque de datos, con la información organizada y lista para ingresar a la base de datos, luego de dar el click en subir la información, en el botón submit, se toma la información en el formulario, con el hash ya creado y se envía al motor de base de datos, en este caso mongodb, este motor de base de datos puede usarse de manera web y en la web se almacena la información de manera distribuida, no centralizada. Lo descrito anteriormente es la parte HTML de la pestaña 1 de la página web, por otra parte se encuentra la descripción de la pestaña 2 de la página web, que es donde se visualizan los bloques creados, para llegar a esa página podemos dar click en la barra de tareas donde dice “ver bloques” esto nos llevara a una página, donde la información mostrada en pantalla son los bloques creados, cada bloque tiene los datos que se le ingresaron y están organizados desde más antiguo a más recientemente creado, la estructura HTML se complementa con el lenguaje javascript que tiene la opción de crear una condicional for para recorrer una lista e ir mostrando en pantalla cada bloque de datos dentro de la lista, esa lista se solicita por medio de una consulta a la base de datos, el HTML tambien se combina con el CSS para darle color y mejor ubicación a cada parte del HTML.

3.6.4. CSS (Cascading Style Sheets)

Lenguaje de diseño utilizado para describir la presentación de un documento escrito en HTML o XML. CSS permite controlar el aspecto visual de las páginas web, definiendo estilos para elementos como colores, fuentes, márgenes, líneas, alturas, anchos, imágenes, y la disposición general de los elementos en una página. Al separar la estructura del contenido (HTML) del diseño (CSS), se facilita el mantenimiento y la actualización del sitio web, ya que los cambios en el estilo pueden realizarse desde un único archivo CSS sin necesidad de modificar el HTML de cada página.

El CSS es un lenguaje de diseño gráfico, y nos ayuda a darle color, ubicación y que sus componentes sean más amigables para el usuario, en este software el CSS estará compuesto por colores de fondo como el gris y el negro con la intención de que la vista descanse y que las pantallas gasten la menor energía posible, ya que los colores oscuros en los fondos de pantalla requieren de menos electricidad para su funcionamientos y estos colores ayudan a que el ojo este mas descansado y se esfuerce menos, lo anterior descrito es todo lo contrario a los fondos blancos o de colores claros.

El color que se usa en la letra es el blanco y el rojo, con la intención de que se visualice mejor y de manera definida los textos del software con tecnología Blockchain, los botones tienen fondo gris y letra blanca, para que se diferencien de los campos de escritura que tienen letra negra con fondo gris. Lo anterior es la descripción general de los colores que lleva el software y también se usó bootstrap para realizar el responsive en la página web, esta característica nos ayuda a que cuando se use un celular o una Tablet o un televisor para acceder a este software, los componentes HTML se organicen de la mejor manera y no se amontonen en una parte, también evita que se distorsione la información en este programa, para el responsive se importó sus librerías por medio de una etiqueta HTML con conexión a la página de Bootstrap, luego en cada etiqueta HTML del software se le añadió las clases de Bootstrap que ya tiene definido el responsive, las clases son una característica en la que uno puede especificar las propiedad CSS de cada componente HTML, las clases de CSS que se usaron son: en la pestaña 1 de la página web, se usó el “bg- danger” para darle responsive a la etiqueta HTML del formulario y también darle un color de letra rojo a una parte de este. Mas adelante en el código también se usó bg-secondary para las partes del formulario tengan resposive y color gris en sus componentes, también se usó en los botones btn-secondary para que el color de estos submit sea gris, el fondo de toda la página se escogió de color negro y se escogió la característica CSS siguiente “p-3 mb-2 bg-dark text-white" para darle responsive y un background o fondo de pantalla color negro con el color de texto blanco. También se uso el navbar que en otras palabras es la barra de tareas el cual lleva un color gris de fondo con unas letras blancas y con el siguiente CSS se configura esta característica sin dejar de ser responsive, "navbar navbar-expand-lg navbar-secondary bg-secondary".

En la pestaña 2 de la página web con Blockchain se usó la característica llamada card para darle forma a los bloques de información que se traen desde la base de datos, para que este tuviera responsive y un fondo de color gris se escogió la siguiente característica “card bg-secondary mb-3” y también la propiedad CSS “card-body”.

3.6.5. MongoDB

El motor de base de datos MongoDB se escogió para este desarrollo de software con tecnología Blockchain. con el fin de que este paso a paso sirva para usarse en la nube de MongoDB sin necesidad de hacerle cambios importantes en los pasos explicados en este documento, en este motor de base de datos iniciaremos descargando la versión local en el computador y posteriormente en el código Python anexaremos la sección de conexión con la base de datos, en este caso se guarda en una variable client el JSON que tiene la informacion necesaria para la creacion de la conexión con la base de datos, usando la librería pymongo y su funcion MongoClient donde ingresamos los parametros que son necesarios, luego de guardar en una variable bd el nombre de la base de datos llamada “VBlockchain” creada en el cliente anteriormente nombrado, luego se crea la colección llamada “Bloques”. La base de datos MongoDB recibe estos códigos y crean los bloques en la base de datos que puede estar en una nube. Luego de crear la conexión se procede a enviar los datos recolectados en el frontend y que llegan al motor de base de datos, en este caso se usa el col.insert_one el cual realiza un ingreso de informacion a la base de datos, teniendo ya programado la conexión, luego realizamos el insert y por ultimo usamos una consulta para traer todas las colecciones creada o tambien llamadas cadenas de bloques, para traer toda la informacion se usa el col.find() que es funcion de pymongo de python para realizar una consulta en la base de datos y traer todas los bloques creados.

La encriptación de la información y la creación del hash criptográfico se realizó con el componente hashlib y su opción de sha256 que nos ayudara a crear los 32 hexadecimales con toda la información obtenida de ese primer ingreso de datos y con ese hash criptográfico se comprueba que los datos en ese bloque no han sido modificados. Porque si 1 solo dato se cambia, entonces el hash criptográfico cambia totalmente.

4. Desarrollo específico de la contribución

En esta parte del documento mostraremos los resultados obtenidos luego de realizar el paso a paso de la metodología ya explicada anteriormente, en estos resultados podemos ver la creación de los interfaces que verán los usuarios, la estructura de la base de datos y su creación, también como se ingresarán la información de las votaciones al software, la creación del JSON con los datos, la encriptación de la información y las pruebas de software.

El código fuente completo de la aplicación se encuentra en el siguiente github:

https://github.com/blockchaintfmunir/blockchain-electoral

4.1.CREACIÓN DE INTERFACES DE USUARIO

El software creado, tiene 2 interfaces en las que se puede navegar, la primera consiste en el ingreso de los datos a la aplicación, y en la segunda vista se muestran los diferentes bloques que se han creado anteriormente. En la vista 1 hay un formulario con método POST y diferentes input que sirven para ingresarle la información a la aplicación, se usó clases de CSS que están en el framework Boostrap para darle color a la aplicación, se usó HTML para la estructura del software, usando un form para el ingreso de la información, dentro de estas etiquetas form hay unos inputs que reciben la información y por medio del boton submit envía esta información hacia el backend en Python, estos inputs tienen un name relacionado a la información que se le ingresa, y también tienen la class de css que llevara esa etiqueta y un type de texto, porque la información se digitara en estos inputs.

También se usó CSS para darle los colores a las interfaces y la ubicación de la letra con la características del formulario que uno quiera, para lograr las anteriores características se escogen las class de bootstrap por su fácil codificación del responsive, En la siguiente imagen se muestra el código del formulario de ingreso de información que tiene la primera página.

Ilustración 9 - Código Fuente - Formulario de registro

```
<h1>Votacion electronica con Blockchain</h1>
<h2>Por favor ingrese la informacion de la votacion : </h2><br>
<form action="/upload" method="POST" enctype="multipart/form-data">
  <table class="table table-responsive table-bordered">
    <thead class="bg-danger">
      <tr>
        <th scope="col" > Caracteristicas de la informacion: </th>
        <td scope="col" > Datos que se van a ingresar: </th>
      </tr>
    </thead>
    <tbody>
      <tr>
          <th scope="row" class="bg-secondary">Nombre de la opcion 1:</th>
          <td scope="row" class="bg-secondary"><input type="text" name="nombre_1" class="btn btn-secondary" /></td>
      </tr>
      <tr>
        <th scope="row" class="bg-secondary">Cantidad de votos de la opcion 1:</th>
        <td scope="row" class="bg-secondary"><input type="text" name="votacion_1" class="btn btn-secondary" /></td>
      </tr>
      <tr>
        <th scope="row" class="bg-secondary">Nombre de la opcion 2:</th>
        <td scope="row" class="bg-secondary"><input type="text" name="nombre_2" class="btn btn-secondary" /></td>
      </tr>
      <tr>
        <th scope="row" class="bg-secondary">Cantidad de votos de la opcion 2:</th>
        <td scope="row" class="bg-secondary"><input type="text" name="votacion_2" class="btn btn-secondary" /></td>
      </tr>
      <tr>
        <th scope="row" class="bg-danger"><input type="submit" value="Subir" class="btn btn-secondary"/></th>
      </tr>
```

En la vista 2 antes de mostrar la información en pantalla, la página realiza una consulta en MongoDB en la cual se piden todos los bloques creados con su respectiva información, en la siguiente imagen, se muestra un ciclo creado en el lenguaje javascript el cual empieza con el for y termina con el endfor donde al principio del codigo se especifica la condicional que permitira dividir los blosques de datos uno a uno y el codigo que está en medio del for y el endfor es lo que se repite, de esa manera se muestra en pantalla cada bloque de la cadena con su respectiva información, en HTML se usó un tbody que sirve como cuerpo de los bloques de información que se imprimen en pantalla, también usamos para cada bloque de la cadena, unas etiquetas llamadas card que ayudan a separar cada bloque del siguiente y dentro de cada bloque hay unos td y th, estas etiquetas nos ayudan a diferenciar entre fila y columna dentro de cada card o bloque de la cadena de bloques, también se usó CSS con el framework boostrap para darle color a la muestra de la información y se escogió colores de fondo opacos de luz con el fin de evitar el brillo en la pantalla, cada etiqueta de HTML con su respectivo CSS debe estar abierta y luego cerrada en esta pestaña de la página web. En la siguiente imagen se muestra el form que recorre los bloques para mostrarlos en pantalla.

lustración 10 - Código Fuente - Formulario de registro

```
46      {% block result %}
47
48    <div class="container" id="test">
49                <tbody id="body">
50                    {% for i in listabloques%}
51                    <div class="card bg-secondary mb-3" >
52                      <div class="card-body">
53                    <tr>
54                        <td> <th> NumeroBloque = </th> {{ i["NumeroBloque"] }}</td>
55                        <br>
56                        <td> <th>nombre_1 =</th> {{ i["nombre_1"] }}</td>
57                        <br>
58                        <td> <th>votacion_1 = </th>{{ i["votacion_1"] }}</td>
59                        <br>
60                        <td> <th>nombre_2 = </th> {{ i["nombre_2"] }}</td>
61                        <br>
62                        <td> <th>votacion_2 = </th>{{ i["votacion_2"] }}</td>
63                        <br>
64                        <td> <th>HashAnterior = </th> {{ i["HashAnterior"] }}</td>
65                        <br>
66                        <td> <th>HashActual = </th> {{ i["HashActual"] }}</td>
67                    </tr>
68                  </div>
69                </div>
70                    {% endfor %}
71    </div>
72
73  {%endblock%}
```

4.2. ESTRUCTURA DE BASE DE DATOS

En esta aplicación, el motor de base de datos que se va a usar es MongoDB se escogio por lo innovador que traen los motores de base de datos NoSql y el libiano peso de los datos guardados en los NoSql a diferencia de los motores SQL, tambien se escogio este motor de base de datos porque ofrece la opcion en la nube para completar las caracteristicas del Blockchain, en el codigo de Python se usa la librería pymongo para el manejo de las bases de datos, con esta librería, se crea la comunicación con la base de datos, tambien con esta librería se crea la base de datos y por último se hacen las consultas de la informacion guardada, en este caso para mostrarla en la segunda pestaña de “ver bloques” para ver la cadena de bloques. Luego de tener comunicación con mongoDB se crea una base de datos llamada “Blockchain” que tendrá diferente colección llamadas “Bloque” como se ve en la siguiente imagen.

Ilustración 11 - Registro en la Base de Datos

La información que se ingresa a el software queda almacenada en MongoDB como un JSON, MongoDB queda en espera de la creación del siguiente bloque, ya que se le hará la consulta del hash anterior, para luego crear el siguiente hash actual, con.

4.3.ADQUISICIÓN DE INFORMACIÓN DE VOTACIONES

En esta parte del software se usó el lenguaje de etiquetas llamada HTML un form que usa el método POST que enviara la información en los inputs hacia una función llamada /upload como se muestra en la siguiente imagen

Ilustración 12 - Código Fuente - Form

```
35          <h1>Votacion electronica con Blockchain</h1>
36          <h2>Por favor ingrese la informacion de la votacion : </h2><br>
37          <form action="/upload" method="POST" enctype="multipart/form-data">
```

```
46              <tr>
47                  <th scope="row" class="bg-secondary">Nombre de la opcion 1:</th>
48                  <td scope="row" class="bg-secondary"><input type="text" name="nombre_1" class="btn btn-secondary" /></td>
49              </tr>
```

4.4.CREACIÓN DE JSON CON LA INFORMACIÓN

Para crear el JSON que se va a enviar a la base de datos, primero hay que crear la función que va a recibir la información dentro de los inputs explicados anteriormente, esto se ejecuta luego de dar click en el boto subir, y la información de los inputs se almacenan en las variables indicadas en la siguiente imagen.

Ilustración 13 - Código Fuente - Form

```
23  #Este codigo se ejecuta cuando se activa guardar los datos
24  @app.route("/upload", methods=['POST'])
25  def uploader():
26      if request.method == 'POST':
27          #Se crean las variables y se guarda lo que viene del primer formulario
28          nombre_1 = request.form['nombre_1']
29          votacion_1 = request.form['votacion_1']
30          nombre_2 = request.form['nombre_2']
31          votacion_2 = request.form['votacion_2']
32          #Conexion a la base de datos "Blockchain" y seleccion de la coleccion "Bloques"
33          client = MongoClient('localhost', port=27017, username='', password='')
34          db = client['VBlockchain']
35          col = db['Bloques']
```

Luego de tener la información en su respectiva variable, procedemos a organizar el JSON que posteriormente se insertara en la base de datos, como se muestra en la siguiente imagen.

Ilustración 14 - Código Fuente – JSON del registro de Votaciones

```
51          col.insert_one({'NumeroBloque': NumeroBloque, 'nombre_1': nombre_1, 'votacion_1': votacion_1,
52                  'nombre_2': nombre_2, 'votacion_2': votacion_2, 'HashAnterior': HashAnterior, 'HashActual': HashActual
53          })
```

4.5. ENCRIPTACIÓN DE LA INFORMACIÓN

En la encriptación de la información de los votantes, se usará la librería de Python llamada hashlib y de esta librería usaremos sha26 para encriptar la información y cuando ya tenemos esta encriptación, le damos la opción hexdigest para convertirlo en hexadecimales y podemos guardar con más facilidad, como se muestra en las siguientes imágenes.

Ilustración 15 - Código Fuente – JSON del registro de Votaciones

```
8   #hash para la encriptacion de la cadena de bloques
9   import hashlib
```

```
46          #Se encripta toda la informacion con sha256 anexando el hash
47          encri = hashlib.sha256(NumeroBloque.encode()+nombre_1.encode()+votacion_1.encode()
48                                 +nombre_2.encode()+votacion_2.encode()+HashAnterior.encode())
49          HashActual = encri.hexdigest()
```

4.6. FUNCIONAMIENTO DE LA BASE DE DATOS

La conexión con la base de datos se realiza con el lenguaje de programación Python y la librería pymongo. Como se evidencia en la imagen siguiente.

Ilustración 16 - Código Fuente – JSON del registro de Votaciones

```
10  #Se guarda en una base de datos mongodb
11  from pymongo import MongoClient
```

De la librería pymongo se usa la función MongoClient donde se le pone el string de conexión a la base de datos, y esto se guarda en la variable client, que luego se usara para crear la base de datos y la colección llamada bloques, como se muestra en la siguiente imagen.

Ilustración 17 - Código Fuente – JSON del registro de Votaciones

```
32          #Conexion a la base de datos "Blockchain" y seleccion de la coleccion "Bloques"
33          client = MongoClient('localhost', port=27017, username='', password='')
34          db = client['VBlockchain']
35          col = db['Bloques']
```

Luego de tener la conexión base de datos establecida, se realiza una consulta find_one a la variable que llamamos NumeroBloque : 1 , si la respuesta es que esa variable, no está en ningún boques entonces se crea la base de datos, entonces la variable HashAnterior, que vamos a crear la nombraremos “Genesis” y el HashActual se crea con toda la información del bloque que se va a crear.

Ilustración 18 - Código Fuente – JSON del registro de Votaciones

```
37        #Se guarda en la variable "colvacia" si la coleccion "Bloques" esta vacia
38        colvacia = col.find_one({"NumeroBloque": "1"})
39
40        #Si "colvacia" es "None" el HashAnterior es "Genesis" y el NumeroBloque es 1
41        if(colvacia == None):
42            HashAnterior = "Genesis"
43            NumeroBloque = 1
44            NumeroBloque = str(NumeroBloque)
```

Si ya había un NumeroBloque : 1 los pasos son similares a los pasos de si no hubiera colección anterior, con la diferencia de que el bloque anterior se consulta a la base de datos, con un find().sort(‘NumeroBloque’, -1).limit(1) el cual me trae toda la información del bloque anterior y luego, de esa colección de base de datos en una variable, la recorremos con un for, para ir seleccionando el campo hashactual de la lista anterior a la que se va a crear, con este hexadecimal podemos crear el siguiente bloque. También se adquiere el NumeroBloque del último bloque creado, también cogemos el dato numbloqant y lo volvemos número para anexarle un 1 y dejarlo como NumeroBloque, para ser usado en la siguiente creación de bloque de informacion y el hashanterior se guardará al final como una parte importante de la cadena de bloques, como se muestra en la siguiente figura.

Ilustración 19 - Código Fuente – JSON del registro de Votaciones

```
55        #Si "colvacia" es DIFERENTE de "None" el HashAnterior es varia y el NumeroBloque es depende de cuantos se hayan creado
56        if (colvacia != None):
57            #Se guarda en la variable "bloquecodificado" el ultimo bloque que se inserto
58            bloquecodificado = col.find().sort('NumeroBloque', -1).limit(1)
59            #se recorren los items dentro de ese ultimo bloque
60            #cada item se guarda en la variable "bloqueanterior"
61            for item in bloquecodificado :
62                bloqueanterior = item
63            #Dentro de la lista que se guardo en "bloqueanterior" se selecciona el campo "NumeroBloque"
64            numbloqant = bloqueanterior['NumeroBloque']
65            #Dentro de la lista que se guardo en "bloqueanterior" se selecciona el campo "HashActual"
66            hashbloqant = bloqueanterior['HashActual']
67            numbloqant= int(numbloqant)
68            NumeroBloque = numbloqant + 1
69            HashAnterior = hashbloqant
70            #Se convierten las variables en string para poderla guardar en la base de datos
71            NumeroBloque = str(NumeroBloque)
72            HashAnterior = str(HashAnterior)
```

Luego se usa la función insert_one para ingresarle todo el JSON con la información de las votaciones en la base de datos, en este software se guarda primero el NumeroBloque, que depende de que bloque creados anteriormente habia y luego el nombre_1 que es el nombre del candidato que participa en las elecciones, el cual tendra una cantidad de votos llamados votacion_1, luego el nombre del segundo candidato y su cantidad de voto obtenidos, tambien se agregara el hashanterior que es del bloque anterior al que se va a crear y el hash actual, ya creado co la librería hashlib, todos estos datos se ingresan a mongodb por medio de esta consulta como se ve en la siguiente imagen.

Ilustración 20 - Código Fuente – JSON del registro de Votaciones

```
51          col.insert_one({'NumeroBloque': NumeroBloque, 'nombre_1': nombre_1, 'votacion_1': votacion_1,
52                          'nombre_2': nombre_2, 'votacion_2': votacion_2, 'HashAnterior': HashAnterior, 'HashActual': HashActual
53          })
```

4.7. PRUEBAS DE SOFTWARE

La prueba que se realizó fue ingresar unos nombres de candidatos y sus votos en el primer bloque y verificar el hash que se creó, posteriormente se ingresan la misma información, con la diferencia de que 1 solo digito se cambió y con esto podemos evidenciar que el mínimo cambio, modifica completamente el hash criptográfico indicándonos que hubo un cambio en la información y protegiendo la integridad de los datos. En la siguiente imagen podemos ver el primer ingreso.

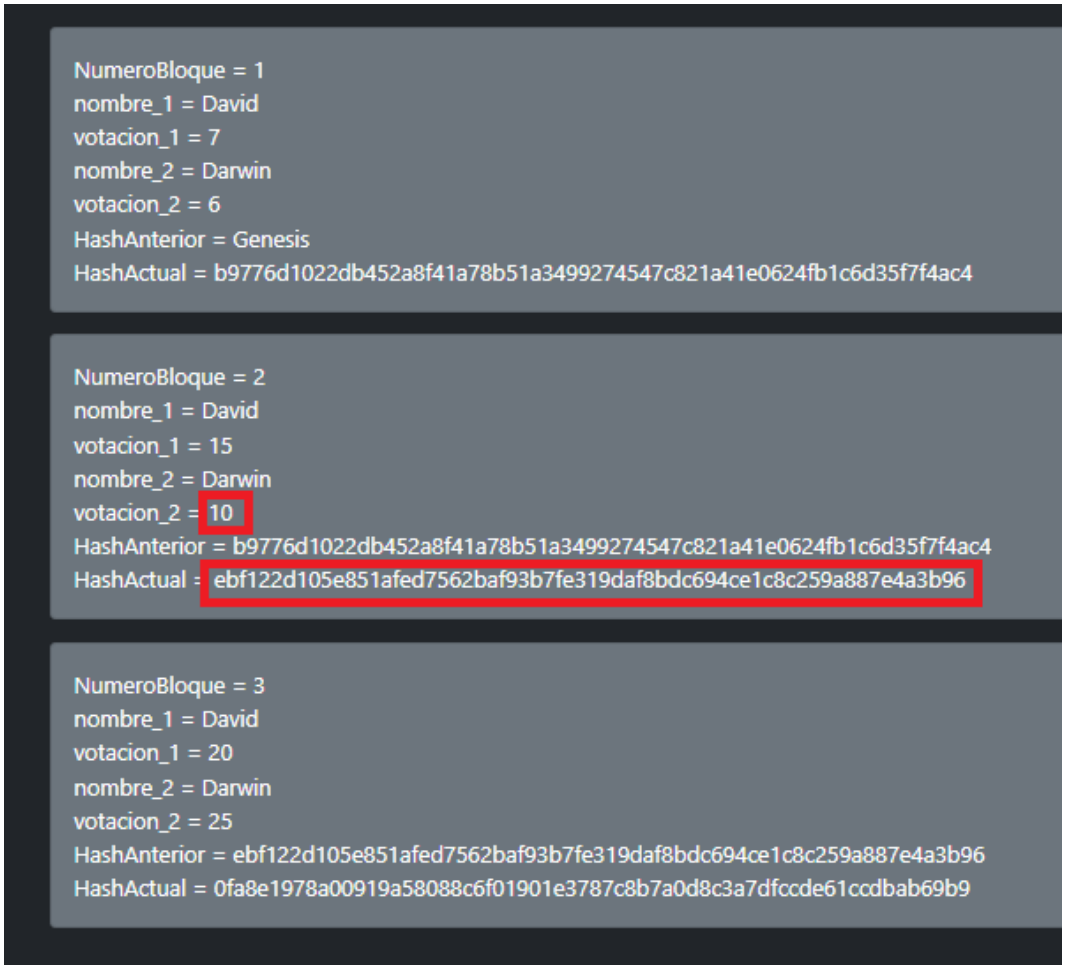
Ilustración 21 - Registro de nuestra prueba

Luego de tener la evidencia del primer insert, procedemos a eliminar el bloque y volver a ingresar la información, en esta oportunidad la votación 2, no será 9 sino 6, como se muestra en la siguiente imagen.

Ilustración 22 - Registro de nuestra prueba

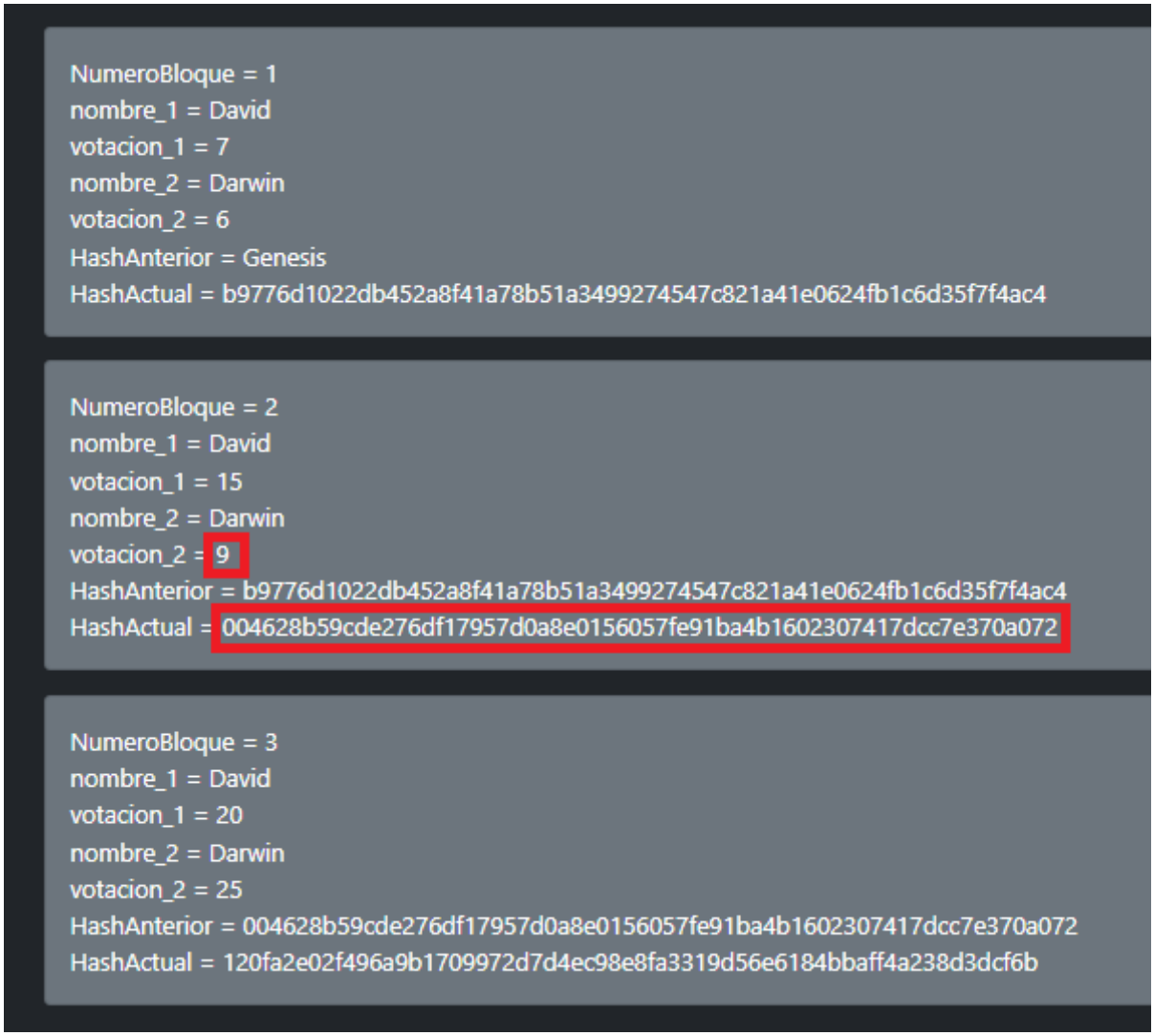
En la prueba 2 En se realizará una inserción de 3 bloques con los mismos datos en la mayoría de veces con excepción de 1 dato que será cambiado a la mitad del segundo bloque y lograremos analizar que el hash actual del bloque final cambia totalmente y de esta manera veremos cómo se afecta el hashactual de los bloques siguientes bloques, con un solo digito cambiado cambian totalmente los siguientes hash. Como se muestra en la siguiente imagen

Ilustración 23 - Registro de nuestra prueba



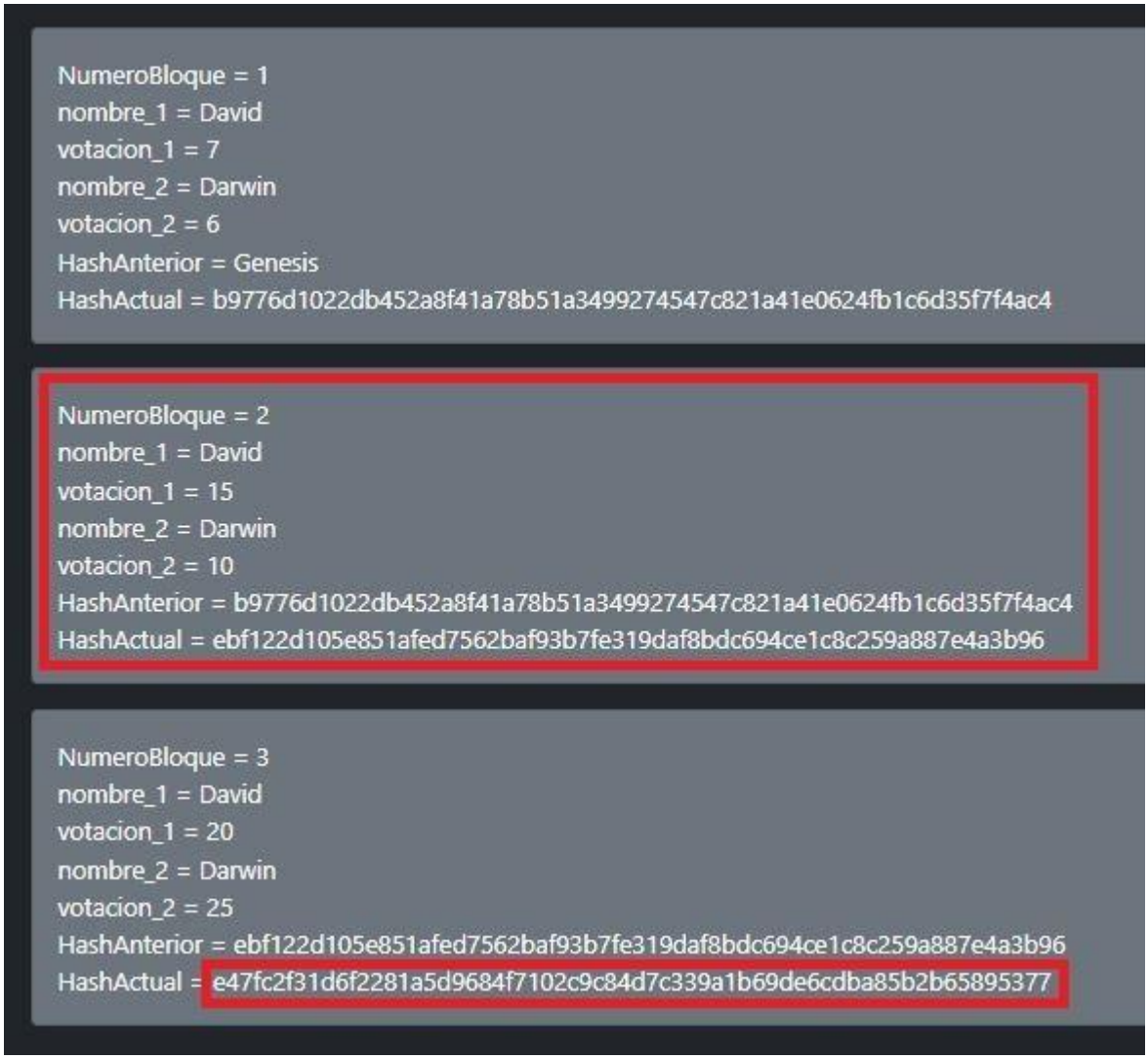
Luego se borraron los registros anteriores y se volvieron a realizar los mismos registros, con la diferencia de que el campo votación_2 se cambió el digito de 10 a 9 y el resto de datos iguales, eso afecto todo el bloque de información que su hash final es diferente y todos los siguientes hashes son diferentes por el cambio de información que hubo, en la siguiente imagen se señala el cambio en el digito y el hash actual como sale de diferente al hash actual del intento anterior de ingreso de información.

Ilustración 24 - Registro de nuestra prueba



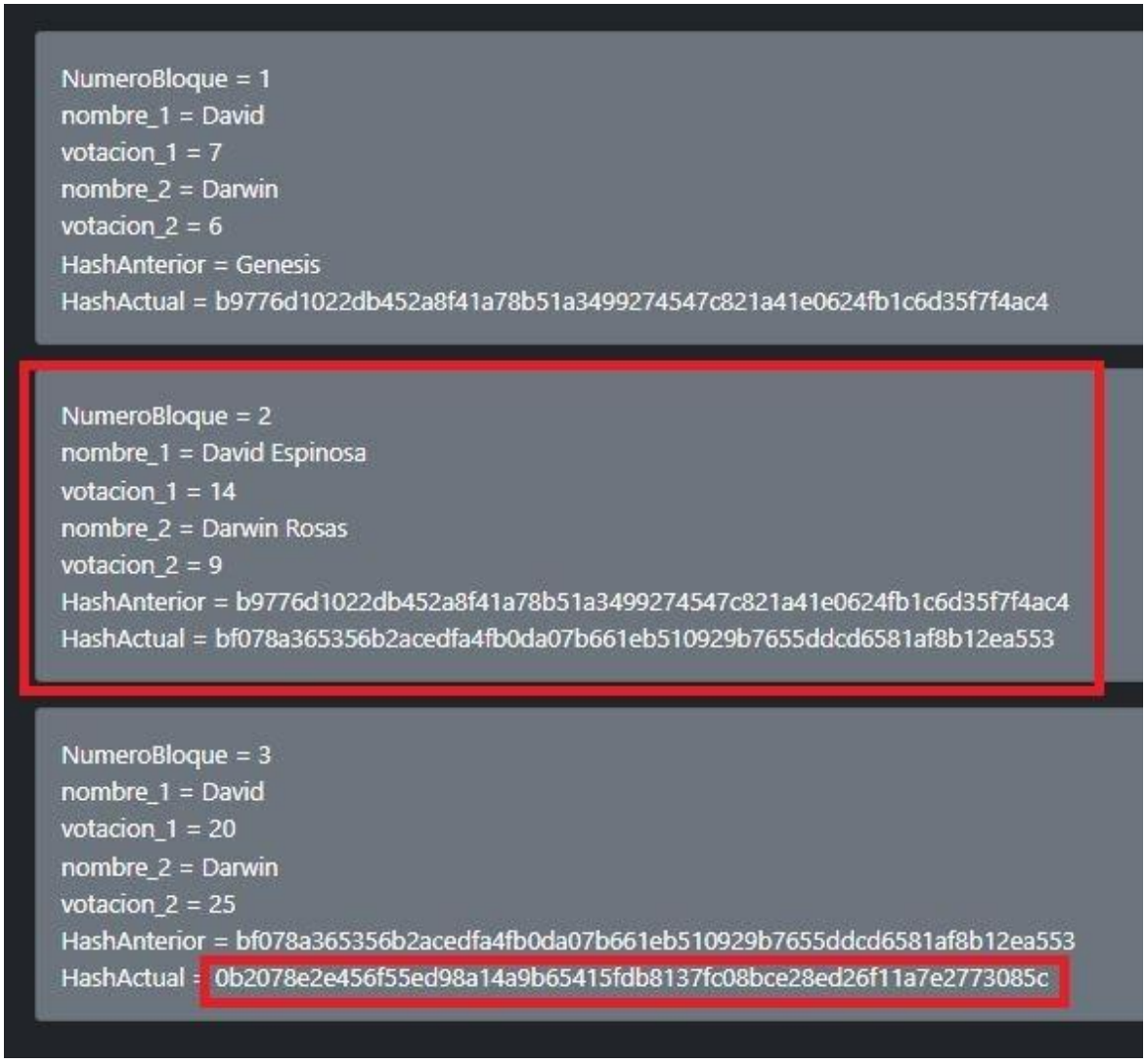
La tercera prueba y ultima que se realizara es la de la modificación de la totalidad de los datos en 1 bloque, con el fin de demostrar que cuando los cambios realizados de manera no autorizada así sean de todos los datos, cambiara el hash criptográfico actual del último bloque creado y de esta manera nos daremos cuenta que hubo una modificación no deseada, por la diferencia en los dígitos del hash criptográfico, como se ve en la siguiente imagen.

Ilustración 25 - Registro de nuestra prueba



Luego de ver la imagen anterior, esta vez escogimos el segundo bloque para modificar los datos electorales totalmente, poniéndole el apellido a cada candidato y cambiando el número de los votos, con esto validamos que los hashes a continuación empiezan a cambiar a como seria normalmente en la imagen anterior, y en la siguiente imagen, se muestra el bloque con la información totalmente cambiada y el siguiente hash empiezan a ser diferentes.

Ilustración 26 - Registro de nuestra prueba



5. ANÁLISIS ECONÓMICO

El análisis económico es esencial para evaluar tanto la viabilidad como la rentabilidad de cualquier proyecto, especialmente en ciberseguridad y Blockchain. Este análisis ofrece una visión clara de los costos y beneficios relacionados con el desarrollo, facilitando la toma de decisiones informadas por parte de los interesados. En este contexto, se realizará un análisis detallado que incluirá la estimación de los costos iniciales, los costos operativos anuales, los beneficios previstos y el cálculo del retorno de la inversión (ROI).

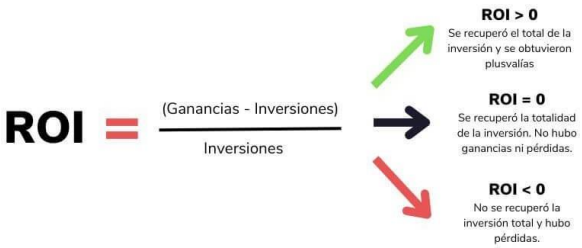
El presente análisis económico se enfoca en los costos asociados al desarrollo de un software de almacenamiento de resultados electorales utilizando tecnología Blockchain. Este análisis considera las tarifas horarias de los diferentes roles involucrados, el total de horas trabajadas, el cálculo de horas con un factor de riesgo del 30% y el valor por rol. Además, se incluye el cálculo del valor total con y sin IVA, la proyección de los costos operativos anuales y un análisis de retorno de inversión (ROI) para evaluar la rentabilidad del proyecto a largo plazo.

5.1. HISTORIA DEL RETORNO SOBRE LA INVERSIÓN (ROI)

El concepto de retorno sobre la inversión (ROI) fue desarrollado por Donaldson Brown a principios del siglo XX mientras trabajaba en DuPont. Originalmente ingeniero eléctrico, Brown se unió a DuPont como representante de ventas y posteriormente fue promovido a analista administrativo. En 1914, como asistente del tesorero, Brown creó una fórmula para medir el rendimiento empresarial que combinaba las ganancias, el capital de trabajo y las inversiones en plantas y propiedades en una sola medida, que denominó "Retorno sobre la Inversión".

La fórmula de ROI se ha convertido en una herramienta esencial en el análisis financiero, permitiendo evaluar la eficiencia de diferentes inversiones al comparar el retorno obtenido con el costo inicial. La fórmula básica del ROI es:

Ilustración 27 - Formula ROI



Este concepto se ha adaptado y utilizado en diversos contextos, incluyendo el análisis de proyectos empresariales y la toma de decisiones estratégicas.

5.2. METODOLOGÍA DE ANÁLISIS ECONÓMICO

En la metodología empleada para este análisis económico, se utilizan diversas técnicas de evaluación financiera y gestión de proyectos. Se han identificado y desglosado tanto los costos iniciales del desarrollo del software como los costos operativos anuales necesarios para su mantenimiento y soporte. Además, se han estimado los beneficios económicos derivados de la implementación del sistema. Por último, se ha calculado el ROI para evaluar la rentabilidad del proyecto a largo plazo.

5.3.COSTOS INICIALES DE DESARROLLO

5.3.1. Desglose de Costos por Rol

Tabla 2 - Desglose Actividades

										M1			M2							M3					M2	
Actividad	Horas por Ingeniero	Días	Arquitecto	Analista	Desarrollador	Frontend	Ing DevOps	Tester	Ingenieros	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14
Gestión del Proyecto	1890	210	1	1				1	3	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Análisis y Diseño Técnico	180	20	1		1				2		x	x											
Desarrollo y Pruebas Unitarias	630	70			1	1			2				x	x	x	x	x	x	x				
Pruebas Funcionales	90	10						1	1											x	x		
Apoyo en la instalación de versiones en el laboratorio	135	15	1		1		1		3													x	
Generación de artefactos desplegables	45	5			1				1													x	
Diseño y documentación de pruebas sugeridas	90	10		1	1				2													x	
Evidencias de pruebas	45	5						1	1													x	
Documento funcional de la aplicación	90	10		1	1				2														x
Total	3195	355																					

Tabla 3 - Desglose Costos ROI

Roles	Tarifas	Total Horas	Total Horas * role * riesgo	Valor por roles
Arquitecto	$ 36,00	765	994,5	$ 35.802,00
Analista Funcional	$ 22,00	720	936,0	$ 20.592,00
Desarrollador Backend	$ 22,00	585	760,5	$ 16.731,00
Desarrollador Frontend	$ 21,00	315	409,5	$ 8.599,50
Ing DevOps	$ 36,00	45	58,5	$ 2.106,00
Tester	$ 22,00	765	994,5	$ 21.879,00

% Riesgo	30%
Total Horas	4154
Valor Total (sin IVA)	$ 105.709,50
Valor Total (con IVA)	$ 125.794,31

Este costo cubre todas las fases del desarrollo del software, desde la planificación inicial hasta la finalización del producto. Incluye también pruebas de funcionalidad y seguridad, asegurando que el software cumple con todos los requisitos técnicos y legales.

5.3.2. Proyección de Costos Operativos Anuales

Una vez desarrollado el software, se deben considerar los costos operativos anuales relacionados con el mantenimiento y soporte del sistema. Estos costos aseguran que el software permanezca actualizado, seguro y eficiente en su operación diaria.

Tabla 4 - Costos Operativos Anuales

Actividad	Horas Anuales	Tarifa (USD)	Costo Anual (USD)
Mantenimiento y Soporte	600	$36.00	$21,600.00
Actualizaciones de Software	400	$22.00	$8,800.00
Mejoras de Seguridad	300	$22.00	$6,600.00
Auditorías de Seguridad	200	$22.00	$4,400.00
Soporte Técnico	200	$22.00	$4,400.00
Total Anual	**1700**		**$45,800.00**

5.3.3. Descripción del ROI

El retorno de inversión (ROI) es una métrica crucial para evaluar la rentabilidad de un proyecto. Se calcula comparando los beneficios económicos generados por el proyecto con los costos totales de inversión.

La implementación de tecnología Blockchain para el almacenamiento de resultados electorales ofrece múltiples beneficios que van más allá de los económicos directos. Entre estos beneficios se incluyen la reducción de fraudes, la mejora de la seguridad, y el aumento de la transparencia y eficiencia del proceso electoral.

5.3.4. Cálculo del ROI

- **Costo Inicial Total (sin IVA):** $105,709.50
- **Costo Operativo Anual:** $45,800.00
- **Beneficios Anuales Estimados:**
  - Reducción de costos de auditorías tradicionales: $50,000.00
  - Disminución de fraudes y errores en el conteo de votos: $100,000.00
  - Ahorro en tiempo y recursos de verificación de resultados: $30,000.00

Total, Beneficios Anuales: $180,000.00

- **Para un período de cinco años:**
  - Inversión Total a 5 Años:

$$\$105{,}709.50 + (5 * \$45{,}800.00) = \$334{,}709.50$$

  - Beneficios Totales a 5 Años:

$$5 * \$180{,}000.00 = \$900{,}000.00$$

  - ROI

$$ROI = \frac{(Beneficios\ Totales\ -\ Inversión\ Total)}{Inversión\ Total}$$

$$ROI = \frac{(900{,}000.00\ -\ \$334{,}709.50)}{\$334{,}709.50} = 1{,}69\ \rightarrow\ 169\%$$

5.3.5. Parámetros

Tabla 5 - Tabla de Beneficios Anuales Estimados

Descripción	Valor (USD)
Costo Inicial Total (sin IVA)	$105,709.50
Costo Operativo Anual	$45,800.00
Beneficios Anuales Estimados	$180,000.00

5.3.6. Beneficios Anuales Estimados

Tabla 6 - Tabla de Beneficios Anuales Estimados

Concepto	Monto (USD)
Reducción de costos de auditorías tradicionales	$50,000.00
Disminución de fraudes y errores en el conteo de votos	$100,000.00
Ahorro en tiempo y recursos de verificación de resultados	$30,000.00
Total Beneficios Anuales	**$180,000.00**

5.3.7. Cálculo del ROI a 5 Años

Tabla 7 - Cálculo del ROI a 5 Años

Descripción	Valor (USD)
Inversión Total a 5 Años	$334,709.50
Beneficios Totales a 5 Años	$900,000.00
ROI	169%

5.3.8. Detalle del Cálculo

Tabla 8 - Detalle del Cálculo

Concepto	Valor (USD)
Costo Inicial Total (sin IVA)	$105,709.50
(Costo Operativo Anual * 5 Años)	$229,000.00
Inversión Total a 5 Años	**$334,709.50**
Beneficios Anuales Estimados	$180,000.00
(Beneficios Anuales * 5 Años)	$900,000.00
Beneficios Totales a 5 Años	**$900,000.00**
ROI = (Beneficios Totales - Inversión Total) / Inversión Total	1.69 o 169%

6. Conclusiones y trabajo futuro

Se logró evidenciar que gestionar el almacenamiento de los de la información a través de la tecnología Blockchain, se tuvo una mejora considerable en cuestión de seguridad, y descentralización.

Siendo una de las herramientas más utilizadas dentro de su contexto, Bockchain brinda la flexibilidad de poder implementarse con diversos lenguajes de programación, destacando su conexión con C++, Java, Python y JavaScript.

El proyecto se implementó con la finalidad de almacenar los resultados del proceso electoral a través de Blockchain, en el cual al finalizar el proceso de registro se encripta utilizando el algoritmo SHA-256 que es uno de los más seguros del momento.

Se desarrolló el proyecto para garantizar que la información ingresada en los procesos electorales sea segura y escalable en la cual se validó el funcionamiento del software por medio de diferentes pruebas que demuestran la seguridad de la creación de los bloques. En los cuales, el primer bloque creado generó su hash que valida la información registrada. En un segundo registro, se evidencia que sea el hash original y el nuevo hash que contiene la información del registro y del anterior hash.

El análisis económico del proyecto muestra que el costo total estimado para el desarrollo del software de almacenamiento de resultados electorales con tecnología Blockchain, considerando un riesgo del 30% y los impuestos correspondientes, asciende a

$125,794.31. Los costos operativos anuales se proyectan en $45,800.00, asegurando que el sistema continúe funcionando de manera óptima.

El ROI proyectado del 169% en un período de cinco años sugiere que la inversión es altamente rentable, recuperando más del costo inicial en beneficios económicos y operativos. Además de los beneficios financieros, el proyecto contribuirá a mejorar la seguridad, transparencia y eficiencia del proceso electoral, fortaleciendo la confianza pública en los sistemas democráticos

Referencias bibliográficas

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.

Nofer, M., Gomber, P., Hinz, O., & Schiereck, D. (2017). Blockchain. Business & Information Systems Engineering, 59(3), 183-187.

Swan, M. (2015). Blockchain: Blueprint for a New Economy. O'Reilly Media, Inc.

Yli-Huumo, J., Ko, D., Choi, S., Park, S., & Smolander, K. (2016). Where is current research on Blockchain technology? —A systematic review. PloS one, 11(10), e0163477.

Buterin, V. (2014). A next-generation smart contract and decentralized application platform. Ethereum White Paper.

Dannen, C. (2017). Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners. Apress.

Hjalmarsson, F., Hreiðarsson, G. K., Hamdaqa, M., & Hjálmtýsson, G. (2018).

Blockchain-based e-voting system. In 2018 IEEE 11th International Conference on Cloud Computing (CLOUD) (pp. 983-986).

Katz, J., & Lindell, Y. (2007). Introduction to Modern Cryptography. CRC press.

Glaser, F. (2017). Pervasive Decentralisation of Digital Infrastructures: A Framework for Blockchain-enabled System and Use Case Analysis. In Proceedings of the 50th Hawaii International Conference on System Sciences.

Beck, R., Avital, M., Rossi, M., & Thatcher, J. B. (2017). Blockchain Technology in Business and Information Systems Research. Business & Information Systems Engineering, 59(6), 381-384.

Liu, Yi; Wang Qi. (2017). An E-Voting Protocol Based on Blockchain. Southern University of Science and Technology, Shenzhen, China.

J. W. Lamare. (1981). “Eva Etzioni-Halevy. Political Manipulation and Administrative Power: A Comparative Study”, The ANNALS of the American Academy of Political and Social Science 453.1

Business Wire. (1999). Arizona Democratic Party Selects Votation.com to Hold World's First Legally-Binding Public Election Over the Internet.

ACE Project. (2024). Página web del proyecto ACE. Disponible en: http:// aceproject.org/

United States Election Assistance Commission. (2005). Voluntary Voting System Guidelines. Volume I, version 1.0

Tribunal Superior Electoral. (2005). ¿O que faz a urna funcionar? Página web del tribunal superior de Brasil. Disponible en: http://www. tse.jus.br/o-tse/escola-judiciaria-

eleitoral/publicacoes/ revistas-da-eje/artigos/revista-eletronica-eje-n.-5-ano-5/ digressoes-sobre-as-doacoes-de-campanha-oriundas-de-pessoas-juridicas.

India Today. (2008). New counting method for Assembly polls. 2008. Disponible en: https://www.indiatoday.in/elections-2008/story/ new-counting-method-for-assembly-polls- 34541-2008-12-04.

e-Estonia. (2024). Página web de e-Estonia. Disponible en: https://e-estonia. com/

J. Barrati-Esteve, B. Goldsmith y J. Turner. (2012). "International ex- perience with e- voting", Norwegian E-Vote Project. International Foundation for Electoral Systems.

Document disponibil online la adresa http://www. ifes. org/Content/Publications/News-in-Brief/2012/June/ % 7E/media/B7FB434187E943C18F4D4992A4EF75DA. pdf.

Catalini, C., & Gans, J. S. (2016). Some Simple Economics of the Blockchain. NBER Working Paper No. 22952. NBER

Horváth, A., & Zimányi, K. (2022). Exploratory Analysis of Blockchain Platforms in Supply Chain Management. Economies, 10(9), 206. DOI: 10.3390/economies10090206

PwC. (2020). Blockchain technologies could boost the global economy US$1.76 trillion by 2030. PwC

VATrader. All about Return On Investment (ROI) - Formula, History, Pros and Cons!

VATrader

Consejo de Europa. (1981). Convenio para la Protección de las Personas con respecto al Tratamiento Automatizado de Datos de Carácter Personal (Convenio 108).

https://www.coe.int/en/web/conventions/full-list/-/conventions/treaty/108 Decreto 2241 de 1986, Código Electoral Colombiano. https://www.suin-

juriscol.gov.co/viewDocument.asp?ruta=Decretos/30018231

Decreto 1377 de 2013, por el cual se reglamenta parcialmente la Ley 1581 de 2012. https://www.suin-juriscol.gov.co/viewDocument.asp?id=1136422

Directiva (UE) 2016/1148 del Parlamento Europeo y del Consejo, sobre medidas para garantizar un alto nivel común de seguridad de las redes y sistemas de información en la Unión. https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=CELEX%3A32016L1148

Ley 12/2018, de 24 de mayo, de seguridad de las redes y sistemas de información, España. https://www.boe.es/buscar/doc.php?id=BOE-A-2018-6612

Ley 1581 de 2012, Ley de Protección de Datos Personales, Colombia. https://www.suin-juriscol.gov.co/viewDocument.asp?ruta=Leyes/1606220

Ley 527 de 1999, Ley de Comercio Electrónico, Colombia. https://www.suin-juriscol.gov.co/viewDocument.asp?id=1566549

Ley Federal de la Innovación y Tecnología Blockchain, Suiza, 2020. https://www.admin.ch/opc/es/classified-compilation/20201144/index.html

Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y

garantía de los derechos digitales, España. https://www.boe.es/buscar/doc.php?id=BOE-A-2018-16673

Ley Orgánica 5/1985, de 19 de junio, del Régimen Electoral General, España. https://www.boe.es/buscar/doc.php?id=BOE-A-1985-11672

Real Decreto 3/2010, de 8 de enero, por el que se regula el Esquema Nacional de Seguridad en el ámbito de la Administración Electrónica, España. https://www.boe.es/buscar/doc.php?id=BOE-A-2010-1330

Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo, relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre

circulación de estos datos (Reglamento General de Protección de Datos). https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=CELEX%3A32016R0679

UNCITRAL. (1996). Ley Modelo de la UNCITRAL sobre Comercio Electrónico.

BBC News Mundo. (2024). Venezuela: desafíos electorales y la lucha contra el fraude. Razón Pública. Recuperado de https://razonpublica.com/venezuela-desafios-electorales-la-lucha-fraude/

DW. (2024). DW Verifica: ¿Qué tan grande es el riesgo de fraude electoral en Estados Unidos? Deutsche Welle. https://www.dw.com/es/dw-verifica-qu%C3%A9-tan-grande-es-el-riesgo-de-fraude-electoral-en-estados-unidos/a-69598694

France24. (2024). Oposición en Sudáfrica denunció fraude electoral y dijo que emprenderá acciones legales. France 24.https://www.france24.com/es/video/20240602-oposici%C3%B3n-en-sud%C3%A1frica-denunci%C3%B3-fraude-electoral-y-dijo-que-emprender%C3%A1-acciones-legales

CNN Español. (2024). Difícil que se cometa alguna forma de fraude, dice Insulza previo a participar como observador electoral en México. CNN. Recuperado de https://cnnespanol.cnn.com/video/dificil-que-se-cometa-alguna-forma-de-fraude-dice-insulza-previo-a-participar-como-observador-electoral-en-mexico/

EFE Verifica. (2024). Los votos atribuidos a Junts fuera de Cataluña no son un fraude. EFE. https://verifica.efe.com/junts-resultados-madrid-elecciones-europeas-error/

Anexo A.

- **Marco Legal en España**

En España, la normativa electoral, la protección de datos personales y la legislación sobre nuevas tecnologías son cruciales para evaluar la viabilidad de implementar Blockchain en el ámbito electoral.

- *Ley Orgánica del Régimen Electoral General (LOREG)*

La LOREG, aprobada en 1985 y modificada en varias ocasiones, establece las normas fundamentales para el desarrollo de elecciones en España. Esta ley garantiza la transparencia, la equidad y la seguridad del proceso electoral. La introducción de Blockchain en el sistema electoral español requeriría una revisión y posible modificación de la LOREG para asegurar que las nuevas tecnologías se alineen con estos principios básicos (Ley Orgánica 5/1985).

- *Reglamento General de Protección de Datos (RGPD)*

El RGPD, que se aplica en toda la Unión Europea, establece estrictos requisitos para el tratamiento de datos personales. Cualquier sistema electoral basado en Blockchain debe garantizar que los datos personales de los votantes se traten de manera que cumpla con el RGPD, incluyendo principios de minimización de datos, integridad y confidencialidad (Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo, 2016).

- *Ley Orgánica de Protección de Datos Personales y Garantía de los Derechos Digitales (LOPDGDD)*

La LOPDGDD, aprobada en 2018, adapta el RGPD al marco legal español y establece garantías adicionales para los derechos digitales. Esta ley refuerza la necesidad de obtener el consentimiento explícito de los ciudadanos para el tratamiento de sus datos personales y asegura que los datos sean utilizados únicamente para los fines específicos para los que fueron recogidos (Ley Orgánica 3/2018).

- *Ley de Seguridad de las Redes y Sistemas de Información*

Esta ley, que transpone la Directiva NIS de la UE, establece medidas para alcanzar un elevado nivel de seguridad de las redes y sistemas de información en la Unión Europea. Para los sistemas electorales basados en Blockchain, es esencial asegurar la integridad y disponibilidad de los datos, así como la protección contra ciberataques (Ley 12/2018).

- *Esquema Nacional de Seguridad (ENS)*

El ENS es un conjunto de principios y requisitos para garantizar la seguridad de los sistemas de información en el sector público español. Cualquier sistema electoral que utilice Blockchain debe cumplir con los estándares del ENS, asegurando que las medidas de seguridad sean adecuadas para proteger la información electoral (Real Decreto 3/2010).

- **Marco Legal en Colombia**

En Colombia, la normativa electoral y la legislación sobre protección de datos personales y tecnologías de la información son igualmente importantes para considerar la implementación de Blockchain en los procesos electorales.

- *Código Electoral Colombiano*

El Código Electoral establece las normas básicas para el desarrollo de elecciones en Colombia, garantizando la transparencia y la equidad del proceso electoral. La implementación de Blockchain en el sistema electoral colombiano requeriría ajustes en la normativa vigente para integrar esta tecnología de manera efectiva (Decreto 2241 de 1986).

- *Ley Estatutaria de Protección de Datos (LEPD)*

La LEPD, también conocida como Ley 1581 de 2012, regula el tratamiento de datos personales en Colombia. Cualquier sistema electoral basado en Blockchain debe cumplir con los principios de protección de datos establecidos en esta ley, asegurando que los datos de los votantes se traten de manera segura y confidencial (Ley 1581 de 2012).

- *Decreto 1377 de 2013*

Este decreto reglamenta la LEPD y establece las obligaciones de los responsables del tratamiento de datos personales. Para un sistema electoral basado en Blockchain, es crucial obtener el consentimiento de los ciudadanos y garantizar que los datos se utilicen únicamente para los fines autorizados (Decreto 1377 de 2013).

- *Ley 527 de 1999*

La Ley de Comercio Electrónico regula el uso de tecnologías de la información y las comunicaciones en Colombia. Esta ley establece el marco legal para el uso de firmas digitales y la validez de los documentos electrónicos, lo cual es relevante para la implementación de Blockchain en el ámbito electoral (Ley 527 de 1999).

- **Marco Legal Internacional**

A nivel internacional, diversas normativas y estándares buscan regular el uso de Blockchain y otras tecnologías emergentes en procesos electorales y de gestión de datos.

- *Directiva NIS de la Unión Europea*

La Directiva NIS establece medidas para lograr un alto nivel común de seguridad de las redes y sistemas de información en la UE. Esta directiva es relevante para los países europeos que implementan Blockchain en sus sistemas electorales, asegurando la protección contra ciberataques y la integridad de los datos (Directiva (UE) 2016/1148 del Parlamento Europeo y del Consejo, 2016).

- *Convenio 108 del Consejo de Europa*

Este convenio es el primer instrumento legal vinculante a nivel internacional en el ámbito de la protección de datos personales. Los países que implementan Blockchain deben garantizar que el tratamiento de datos personales cumpla con los principios establecidos en el Convenio 108 (Consejo de Europa, 1981).

- *Ley Modelo de UNCITRAL sobre Comercio Electrónico*

La Ley Modelo de la Comisión de las Naciones Unidas para el Derecho Mercantil Internacional (UNCITRAL) proporciona un marco para la adopción de legislaciones sobre comercio

electrónico. Esta ley modelo es relevante para la validación de firmas digitales y documentos electrónicos en sistemas electorales basados en Blockchain (UNCITRAL, 1996).

- *Regulación del Blockchain en Suiza:*

Suiza ha sido pionera en la adopción de Blockchain, incluyendo su uso en votaciones electrónicas. La regulación suiza proporciona un ejemplo de cómo integrar Blockchain en procesos electorales de manera legal y segura, asegurando la transparencia y la confianza pública (Ley Federal de la Innovación y Tecnología Blockchain, Suiza, 2020).