Smart contracts - Una aplicación de blockchain que nos lleva al futuro obligando a implementar soluciones de vanguardia

1st Laura Tobón Ospian
Universidad de Antioquia
Medellín, Colombia
cecilia.tobon@udea.edu.co

2nd Juan David arismendy
Universidad de Antioquia
Medellín, Colombia
juan.arismendy@udea.edu.co

3rd Jasmin Jaramillo
Universidad de Antioquia
Medellín, Colombia
jasmin.jaramillo@udea.edu.co

4th Ramiro Monroy Ramos
Universidad de Antioquia
Apartadó, Colombia
ramiro.monroy@udea.edu.co

Resumen

Actualmente es común hablar del tema de la corrupción en los ámbitos sociales, económicos y políticos; esta situación es altamente preocupante en todos estos aspectos, especialmente en la administración pública en los procesos de votación electoral. Con el fin de enfrentar esta problemática, proponemos investigar e implementar una herramienta tecnológica que pueda soportar y proporcionar transparencia y confiabilidad en estos procedimientos; Para esto sugerimos el uso de la tecnología Smart Contracts (Contratos inteligentes) en entornos Blockchain, con el fin de aprovechar su gran potencial de autoejecución, automatización, trazabilidad en las transacciones e inmutabilidad en la programación, para ofrecer un alto nivel de seguridad y confianza en la ejecución del contrato; adicionalmente generar un aumento en la capacidad de gestión, control y protección de la información de los votantes y del proceso electoral al suministrar información para el uso público**.**

Palabras clave*—smart contract, blockchain, transacciones, nodos.*

***Abstract* — Currently it is common to talk about the issue of corruption in the social, economic and political spheres; This situation is highly worrying in all these aspects, especially in public administration in electoral voting processes. In order to face this problem, we propose to investigate and implement a technological tool that can support and provide transparency and reliability in these procedures; For this, we suggest the use of Smart Contracts technology in Blockchain environments, in order to take advantage of its great potential for self-execution, automation, traceability in transactions and immutability in programming, to offer a high level of security and trust. in the execution of the contract; Additionally, generate an increase in the capacity for management, control and protection of voter information and the electoral process by providing information for public use.**

Keywords*—smart contract, blockchain, transactions, nodes.*

I. Introducción

Es una realidad que la humanidad después de la pandemia tuvo que cambiar y evolucionar a una mayor velocidad de como venía haciéndolo, en donde la virtualización de muchos aspectos del día a día ahora son protagonistas. Tratando de atravesar esa época de crisis vivida por la humanidad, las tecnologías, las ventas por internet y la virtualidad en general se dispararon, y empezaron a surgir oportunidades de negocios, contactos en línea, y algunas alternativas que ya existían como el trabajo en casa o las operaciones en línea las cuales tomaron aún más fuerza. Sobresale entonces con más ímpetu el Blockchain, como una plataforma ideal para la implementación de smart contacts, por ejemplo, por sus características de eficiencia, trazabilidad, seguridad, entre otras. Apalancar un negocio de este tipo de herramientas

permitiría que los clientes se sientan seguros, más aún cuando desconocen las partes que intervienen en el mismo. Estos contratos pueden celebrarse alrededor de múltiples temas, como sistemas de votación, juegos, aplicaciones móviles, servicios de mensajería y más, lo que los hace muy versátiles y revolucionarios, obligando al sistema legal a una actualización de las competencias que hasta ahora no habían dominado ampliamente. Como ingenieros de sistemas el estar actualizados y abiertos al cambio debe ser una característica primordial en el perfil de competencias blandas y duras, no solo por adquirir nuevos conocimientos sino también por ser esos profesionales conocedores que puedan dar soluciones de vanguardia a las necesidades que se presentan con los cambios actuales.

Para enfrentar este desafío, se trabajó mediante la implementación de la metodología scrum, donde el trabajo colaborativo y la distribución de tareas escalables por etapas fue vital, se desempeñaron roles diferentes por parte de los integrantes del equipo y se evaluaron los resultados parciales cada cierto tiempo dependiendo de los sprint propuestos. Esto permitió ver que tan posible sería llevar a cabo el proyecto con el equipo y las herramientas que se tenían a la mano

II. Marco teórico

Blockchain:

Si pensamos en un libro de contabilidad antiguo que es editado manualmente de manera ascendente y no puede borrarse o tacharse, sería una buena analogía con Blockchain, esto es una especie de libro inmodificable que puede eliminar los intermediarios en las transacciones de manera segura. Son bloques enlazados y cifrados aplicables a cualquier tipo de transacción, que debe tener varios usuarios (nodos) en una red, que verifiquen dichas transacciones. Las verificaciones se producen sin revelar las identidades de quien origina la transacción ni de quien la recibiría, los usuarios de la red verifican los fondos y si esto es aprobado, pasa a integrar un bloque de transacciones, que aún no se registra de manera definitiva, para ello es necesario acudir a la minería de datos. Cuando el bloque de transacciones se llena, ahí es cuando debe ser sellado con ayuda de los mineros mediante una serie de cálculos altamente complejos que al ser resueltos permiten que las transacciones y sus registros sean irreversibles añadiendo un bloque nuevo a la cadena de bloques enlazada (Blockchain). Este bloque tiene el registro de todos los datos de la transacción incluyendo el tiempo los cuales son datos públicos, permitiendo conocer todo el recorrido que ha tenido la transacción, reduciendo la inseguridad y desconfianza de los usuarios.[3]

Nodos: En general un nodo de red es un punto en el que se puede crear, recibir o transmitir un mensaje, enfocado en Blockchain un nodo, siendo una base fundamental de esta, es un ordenador que se interconecta con otros nodos equivalentes, todos operando de una forma igual y ejecutan el software que permite que la red funcione orquestando toda la información necesaria.

Plataformas Blockchain: Se pueden utilizar plataformas existentes de Blockchain como Ethereum, Hyperledger Fabric o Binance Smart Chain para desarrollar aplicaciones descentralizadas (dApps) que interactúen con el sistema operativo.

Lenguajes de programación: Para desarrollar Smart contracts y aplicaciones descentralizadas, se pueden utilizar lenguajes de programación específicos de Blockchain, como Solidity para Ethereum o Chaincode para Hyperledger Fabric. También, se pueden emplear lenguajes de programación tradicionales como C++ o Python para la integración con el sistema operativo.[2]

Entornos de desarrollo integrado (IDE): Herramientas como Remix para Solidity o Visual Studio Code con extensiones para Ethereum pueden facilitar el desarrollo y la depuración de contratos inteligentes y aplicaciones Blockchain.

Bibliotecas y SDK: Para interactuar con la red Blockchain y el sistema operativo, se pueden utilizar bibliotecas y SDK (Software Development Kits) proporcionados por las plataformas Blockchain y la comunidad de desarrolladores.

III. Metodología

La metodología Scrum es un marco ágil de gestión de proyectos que se utiliza en el ámbito de las tecnologías de la información (TI) para el desarrollo de software. Fue creado en los años 90 por Jeff Sutherland y Ken Schwaber y se ha popularizado en la industria debido a su flexibilidad y capacidad de adaptación a los cambios. (Schwaber, & Sutherland, 2017)[4].

Scrum se basa en el trabajo en equipo, la colaboración y la transparencia para lograr los objetivos del proyecto de manera eficiente y efectiva. La metodología se divide en ciclos de trabajo llamados sprints, en los que el equipo se enfoca en crear un conjunto de funcionalidades que agregan valor al producto. Durante cada sprint, se llevan a cabo reuniones diarias de seguimiento, reuniones de revisión y retrospectivas para analizar el progreso del proyecto y ajustar el trabajo en consecuencia. (Cohn, 2017)[5].

El uso de Scrum en proyectos de TI ha demostrado varios beneficios, como la mejora en la productividad y la calidad del software desarrollado, la reducción de costos y tiempos de entrega, y una mayor satisfacción del cliente al permitir una mayor colaboración y transparencia en el proceso de desarrollo.

El proceso Scrum consta de varias fases, entre ellas (Rubin, 2012)[6]:

Planificación del sprint: en esta fase se define el objetivo del sprint y se seleccionan las tareas que se van a realizar durante el sprint.

Sprint: el equipo trabaja en las tareas definidas durante un período de tiempo fijo.

Reunión diaria de Scrum: se realiza una reunión diaria para revisar el progreso y determinar los próximos pasos

Revisión del Sprint: al final de cada sprint, se presenta el trabajo completado y se obtiene retroalimentación para mejorar el producto.

Retrospectiva del Sprint: el equipo reflexiona sobre el sprint completado e identifica las mejoras que se pueden realizar para el próximo sprint.
Scrum tiene muchos beneficios en proyectos de TI, entre ellos:
Flexibilidad: Scrum es una metodología flexible y adaptable que permite realizar cambios en el proyecto durante todo el ciclo de vida del proyecto.
Mejora continua: Scrum se basa en la mejora continua y en la retroalimentación constante, lo que permite que el equipo aprenda de sus errores y mejore su desempeño.
Colaboración: Scrum fomenta la colaboración entre los miembros del equipo y el cliente, lo que ayuda a asegurar que se desarrollen productos que satisfagan las necesidades del cliente.
Mayor control: Scrum proporciona un mayor control sobre el progreso del proyecto y ayuda a identificar los problemas y riesgos de manera temprana.

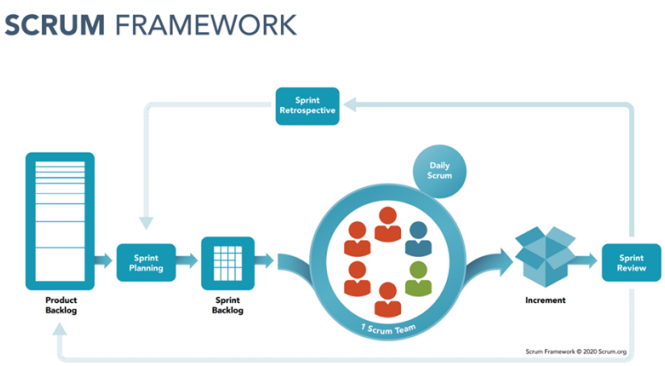

Figura 1. Metodología SCRUM [7]

IV. Implementación

Para implementar un smart contract de ejemplo se decidió hacer un sistema de votación pública donde la idea principal es que un mismo ciudadano no puede votar más de una vez, cómo es común en unas elecciones políticas.

En la implementación se seleccionó el lenguaje Solídity, que es un lenguaje de alto nivel orientado a contratos para implementar contratos inteligentes[8]. debido a su popularidad, curva de aprendizaje y comunidad de apoyo , Adicionalmente se escogió Remix (Figura 2) cómo la herramienta en la nube para codificar, compilar y desplegar el contrato.

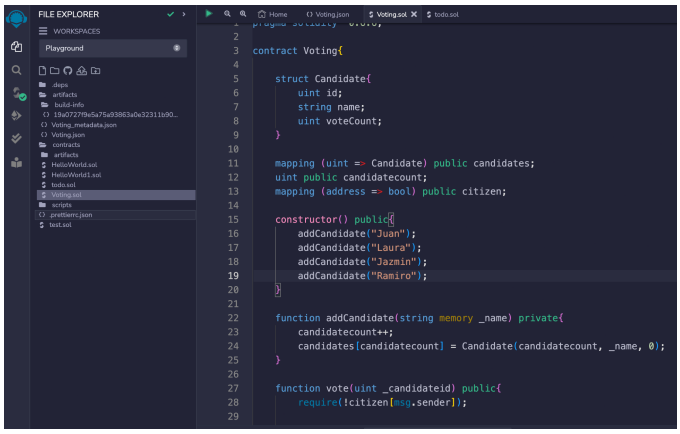

Figura 2. IDE Remix - Elaboración propia

Cómo se mencionó anteriormente, el objetivo principal del contrato desarrollado es el de generar un sistema de votación que evite el fraude, sea auditable y de fácil uso, para esto el contrato requiere inicialmente unos candidatos, que para el caso práctico se incluyeron a los participantes del proyecto,
Luego se debe compilar y desplegar en el blockchain, en esta ocasión se desplegó en el ambiente de Goerli[9], que es una red de prueba de Ethereum,

Una vez desplegado se procede a conectar la cuenta de prueba con la cual votar, ya que se requiere una dirección pública válida para poder votar, esta dirección hace las veces de identificación única de un ciudadano; Con la cuenta ya conectada, se procede a votar por el candidato usando su número de identificación, en este caso es de 1 a 4 , luego de esta acción el usuario ya queda inhabilitado para realizar otra votación pero puede seguir consultando la información de las votaciones en cada momento teniendo en cuenta que cada transacción en el blockchain tiene un costo asociado.

El código de la implementación fue desplegado en un repositorio público de github, referirse a los anexos para más detalles.

V. Resultados

- Logros:
  Durante la implementación del proyecto, se lograron los siguientes avances y éxitos:

- Aprendizaje:
  - Adquisición de conocimientos sobre la tecnología Blockchain y su aplicación en smart contracts.
  - Desarrollo de habilidades en la programación de contratos inteligentes utilizando el lenguaje Solidity.
  - Comprensión de las plataformas y herramientas relacionadas con Blockchain,

como Ethereum, Hyperledger Fabric y Binance Smart Chain.

- Smart contract desplegado:
  - Implementación exitosa de un smart contract de ejemplo.
  - Creación de un sistema de votación pública que garantice la seguridad, la auditabilidad y la prevención del fraude en elecciones políticas.
  - Elección de Solidity como el lenguaje de programación debido a su popularidad, curva de aprendizaje y soporte de la comunidad.
  - Utilización de Remix como una herramienta en la nube para codificar, compilar y desplegar el contrato.

VI. Conclusiones

- *Con la implementación de smart contracts se puede ganar tiempo en comparación con los contratos tradicionales, estos no dependen de firmas sellos o que deban ser timbrados, a su vez se disminuyen los errores humanos.[1]*

- *Por medio de Blockchain el control de un proceso lo tienen los usuarios más no los intermediarios.*

- *Esta tecnología está sujeta a una evolución conceptual y profesional por parte de quienes ejercen la abogacía, quienes deberían expandir sus conocimientos en temas como contratos online, firma electrónica, contratos inteligentes, tecnología blockchain, NFT, metaverso, tokenización y contratos en la realidad virtual.[1]*

- *Es posible que con la expansión del Blockchain, los bancos pierdan el monopolio sobre la economía.*

- *Hasta el momento no se han reportado casos en los que se haya podido hackear cuentas que utilicen Blockchain, lo que lo califica como impenetrable y altamente confiable.*

- *Los Smart contracts son una tecnología nueva e innovadora con el potencial de revolucionar la forma en que realizamos transacciones. Al automatizar la ejecución de acuerdos y eliminar la necesidad de intermediarios confiables, los Smart contracts pueden ayudar a mejorar la confianza, reducir costos, acelerar las transacciones e incluso a cambiar la forma en que se hacen negocios.*

- *Se puede determinar que la implementación de Smart contracts exige un sólido entendimiento de aspectos teóricos como la criptografía, la programación y la propia tecnología Blockchain.*

- *En áreas como gestión de concurrencia, seguridad y sistemas distribuidos, existe una relación entre la teoría requerida para los contratos inteligentes y los temas tratados en un curso de sistemas operativos de la Universidad de Antioquia, destacando la intersección e importancia de tener un conocimiento profundo en ambos campos, para afrontar con éxito los retos tecnológicos actuales*

Referencias

[1]. El papel de los smart contracts en el contexto actual https://es.linkedin.com/pulse/el-papel-de-los-smart-contracts-en-contexto-actual-doinglobal?trk=pulse-article consultada el 08 de noviembre de 2023

[2]. Lista de mejores herramientas de blockchain Lista de la 10 Mejores Herramientas de Blockchain - 101 Blockchains, consultada 23 de septiembre

[3]. ¿Qué es la tecnología blockchain? - IBM Blockchain | IBM. (s. f.). https://www.ibm.com/es-es/topics/blockchain, consultada 24 de septiembre de 2023

[4]. Schwaber, K., & Sutherland, J. (2017). The Scrum guide: The definitive guide to Scrum: The rules of the game. Scrum.org.

[5]. Cohn, M. (2017). Succeeding with agile: software development using Scrum. Pearson Education.

[6]. Rubin, K. S. (2012). Essential Scrum: A Practical Guide to the Most Popular Agile Process. Addison-Wesley Professional.

[7]. Scrum.org, “What is Scrum?” [En línea]. Disponible en: https://www.scrum.org/learning-series/what-is-scrum

[8]. *Solidity*, 07-Jan-2022. [En línea]. Disponible en: https://solidity-es.readthedocs.io/es/latest/. [Consultada: 25-Nov-2023]

[9]. “Goerli testnet,” *Goerli Testnet*. [En línea]. Disponible en: https://goerli.net/. [Visitada: 25-Nov-2023]

ANEXOS

- Repositorio: https://github.com/Juanda16/voting
- Video presentación final: https://drive.google.com/file/d/1VbgbE2kkevAgtK6232E4gJTOxFfzD916/view?usp=sharing