Informe Previo: “Proyecto Blockchain: Blockereum”

Sebastián González, David Pazán, Cristian Villavicencio

I. Introduction

A Continuación, en el presente documento, se ha de presentar el desarrollo de la red ”Blockereum”. En este documento se han de abordar diversos tópicos tales como: la motivación del proyecto, por qué se busca trabajar el problema y el enfoque escogido. Una descripción para la Arquitectura implementada, hablando de las capas que componen la red. Detalles más profundos, relacionados a la implementación. Para finalizar, se han de presentar pruebas del funcionamiento de la presente red.

II. Motivación Del Proyecto

La motivación principal detrás del desarrollo de este trabajo radica en poner a prueba los conocimientos adquiridos a lo largo del semestre, especialmente en el área de *Blockchain*, en conjunción con los aprendizajes previos obtenidos en la asignatura de estructuras de datos.

Con estos antecedentes en mente, el objetivo primordial es crear una red de *Blockchain* que aborde el problema de la centralización en las transacciones monetarias, además, como un componente crucial de lo abordado en el curso, se busca resolver el problema del doble gasto el cual puede surgir durante las transacciones entre los usuarios del servicio. Con esto en consideración, la principal orientación de la red Blockereum se enfocará en aspectos monetarios.

III. Descripción de la Arquitectura

En primer lugar hay que aclarar el funcionamiento de la Arquitectura esta efectuada mediante el modelo de capas, de entre las cuales se presentan:

- Capa de Datos: Correspondiente al manejo y almacenamiento de los registros de transacciones e información de los bloques. Se realiza mediante la programación POO y base de datos *LevevlDB*
- Capa de Infraestructura: La generación y comunicación de los nodos, permitiendo entregar un sistema descentralizado. Se configura mediante la librería *Libp2p*
- Capa de Aplicación: La aplicación de REST API que han de permitir a los usuarios realizar transacciones, consultar transacciones y consultar por bloques de la cadena.
- Capa de integración: Se presenta la lógica que utiliza el sistema para asegurar que todas las transacciones sean procesadas de manera segura y transparente.

IV. Detalles implementación

Hablando respecto a los componentes claves, para la implementación del sistema se tiene en consideración:

- Nodos: Tiene una copia completa de la cadena de bloques, es decir, almacena todos los bloques desde el bloque génesis hasta el bloque más reciente. Participa en la validación y propagación de transacciones y bloques en la red. Permite verificar y validar todas las transacciones de acuerdo con las reglas del protocolo.
- Bloques: Los bloques están caracterizados por poseer un Hash propio, una ID, una llave privada.
- Transacciones: Las transacción típicamente incluye información sobre el remitente, el destinatario y la cantidad de valor que se transfiere. Cada transacción está firmada digitalmente con la clave privada del remitente para garantizar la autenticidad.

Para hablar del funcionamiento hay que tratar dos puntos importantes. El primero es el cómo trabajan los bloques, principalmente en manera para efectuar las transacciones. Este proceso se da mediante la verificación del Hash Previo (Bloque previo establecido como nuestro parámetro) del que están consultando, con la finalidad de ir comprobando si el bloque anterior al que hace referencia el bloque existe y es válido (Figura [1]), una vez que se confirma la transacción, este nuevo bloque se ha de unir a la cadena del nodo, como se puede ver una representación en la (Figura [2])

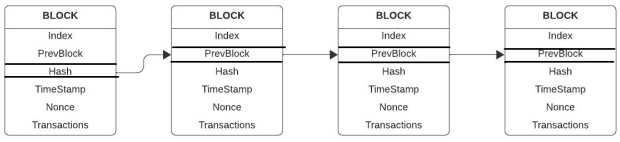

Fig. 1. Representación del funcionamiento del sistema, interacción de bloques de información

El proceso de verificación del Hash Previo en cada bloque juega un papel crucial en la seguridad y la integridad de la cadena de bloques. Al verificar el Hash Previo, cada bloque confirma la existencia y validez del bloque anterior en la cadena. Este proceso crea una conexión segura y secuencial entre los bloques, lo que garantiza que cualquier cambio en un bloque afectaría su hash, invalidando así no solo ese bloque sino también todos los bloques subsiguientes. Esta característica proporciona una inmutabilidad esencial para la integridad global de la cadena de bloques.

La unión de bloques para formar la cadena y la interacción de nodos son fundamentales para la descentralización y redundancia del sistema. Cada nodo mantiene su propia copia de la cadena, y la cohesión entre nodos se logra a través de un proceso descentralizado y consensuado, evitando así la dependencia de un solo punto de control, como se puede apreciar en (Figura [2])

- Validación
- Tiempo de vida del bloque

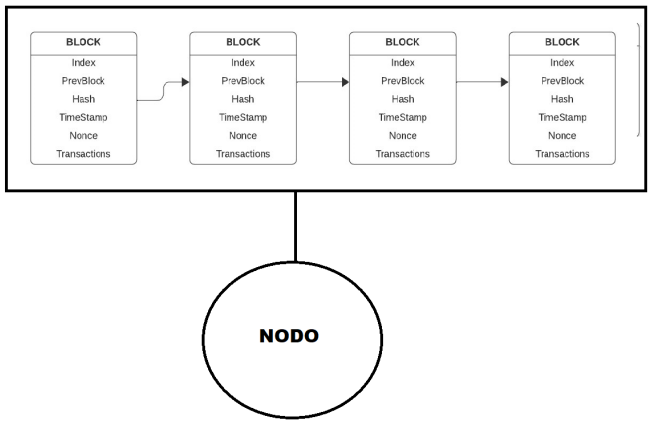

Fig. 2. Representación de la creación y almacenaje del Nodo.

La seguridad en la arquitectura se refuerza mediante el uso de hashes criptográficos para enlazar bloques. Esta técnica garantiza no solo la integridad de cada bloque individual, sino también la inmutabilidad de bloques anteriores, proporcionando una capa adicional de seguridad. Cualquier alteración en un bloque afectaría su hash y, por ende, invalidaría toda la cadena desde ese punto en adelante.

La base de datos LevelDB se ha de configurar con medidas de seguridad que restringen el acceso solo a nodos autorizados. Esto asegura la privacidad de la información almacenada y protege contra accesos no autorizados. La implementación de esta base de datos es esencial para mantener un registro seguro y confiable de las transacciones, movimientos y posesiones de los nodos de los usuarios en la red.

La implementación de LibP2P contribuye significativamente a la seguridad de la red al proporcionar descentralización en los procesos. Esto significa que la red evita posibles ataques y pérdidas que podrían ocurrir en un sistema centralizado. La descentralización también mejora la resistencia a la censura y proporciona redundancia en las conexiones entre nodos, fortaleciendo aún más la seguridad global del sistema.

A. Descripción de implementación del nodo y su protocolo de comunicación

B. Descripción de métodos de lectura/escritura implementados

C. Solución de doble gasto

D. Descripción de el o los modelos de datos utilizados

E. bloque de origen

V. Pruebas de funcionamiento

A. Bloques

B. Nodos