

White Paper: Implementación de una Red Ethereum Simplificada en GOLang

Integrantes: Julio Barra Valenzuela, Vania Vergara Sepulveda, Marcelo Muñoz, Manuel Caceres Farias

Motivación del Proyecto

La motivación detrás del desarrollo de este proyecto es explorar y comprender a fondo la tecnología blockchain y su utilización, centrándonos específicamente en la red Ethereum. Sin embargo, esto también presenta desafíos significativos, como la posibilidad de doble gasto. Este proyecto tiene como objetivo abordar estos problemas mediante la implementación de una versión simplificada de una red Ethereum utilizando GOLang.

Problemas a atacar

Las problemáticas a las que se les busca resolver con esta implementación, son primero que nada, la seguridad y confiabilidad del sistema, puesto que una de las cosas que ofrecen los sistemas descentralizados como el Blockchain es un funcionamiento seguro a través de la participación de sus nodos y la robustez de su modelo, alcanzando así la confiabilidad debido al planteamiento transparente de este, puesto que cualquier usuario es capaz de observar y entender todo lo que sucede en la cadena.

Además, se buscó resolver la principal problemática que surge cuando se implementa una moneda o sistema financiero virtual, el doble gasto.

Enfoque

El enfoque de esta implementación es monetario, y se centra en dos aspectos fundamentales:

Manejar Transacciones Monetarias: Se implementa un sistema que permita a los usuarios realizar transacciones monetarias de manera segura y verificable.

Gestión de Usuarios: Se diseña un sistema que facilite la creación y gestión de usuarios dentro de la red, garantizando la integridad y seguridad de sus identidades.

Descripción de la arquitectura del software

Estructuración de capas lógicas:

Capa de Cliente o Capa de Presentación:

La manera en la que el cliente o usuario se conecta al sistema e interactúa con este es a través de la implementación de un menú en consola, en el cual el usuario puede iniciar con sus credenciales y realizar las distintas operaciones definidas en el sistema, como buscar una transacción o bloque, realizar una transacción, entre otras.

Capa de Servicio o Lógica de Negocios:

Para el funcionamiento del sistema y sus operaciones, se definió una lógica de transacciones y usuarios, y esta capa se encarga de manejar tal lógica que gobierna las transacciones y como estas permiten el cambio de valores entre los distintos usuarios.

Capa de Datos:

Consiste en la implementación de bloques, los que definen la red de blockchain, y en estos es en donde se almacenan las transacciones realizadas por la Capa de Lógica de Negocios, siendo estos bloques definidos por una cantidad de tiempo y una lógica de inmutabilidad. También incluye esta la persistencia de los datos a través de las distintas ejecuciones del sistema

Detalles de implementación

Descripción de implementación del nodo y sus protocolos de comunicación

La implementación del nodo la podemos encontrar p2p/node.go, en la cual se define la creación del nodo utilizando la librería libp2p, puesto que la asignación de la dirección de los nodos se vuelve una tarea trivial cuando se emplea esta.

Para el manejo de la conexión con los peers, se utilizó la librería Multiaddr, puesto que esta ofrece tratar con las direcciones y los protocolos de cualquier conexión de una manera simple de entender y leer, haciendo que el código sea más fácil de mantener y modificar a futuro.

Descripción del método de lectura/escritura implementados

La lógica utilizada para manejar la lectura y escritura de los bloques y las transacciones se encuentra definida en "api.go"

Para la lectura de los bloques, se define una blockdb que contiene los distintos bloques, los cuales dentro suyo tienen las transacciones realizadas durante su tiempo útil, estas son

accesibles llamando a la api, ya se con una consulta del bloque entero o de una transacción en específico.

#Ahora se hace en API.GO

La lógica utilizada para manejar la escritura de transacciones a un bloque, se puede encontrar en el "main.go" (definida en el caso número 2 del menú de usuario) y esta lo que hace es adjuntar a las transacciones del bloque actual la transacción consolidada por los datos del usuario y la dirección y monto que introdujo el usuario. La manera que obtiene este bloque actual, es leyendo la entidad definida en "entities.go"

Solución del doble gasto

Para la solución del doble gasto, se implementa un sistema de decisión, en este caso, se utiliza un algoritmo de consenso que busca eliminar la posibilidad de que dos transacciones utilicen el mismo token.

Descripción de el o los modelos de datos utilizados

Bloque de origen

Pruebas de funcionamiento

Para validar la implementación, se llevaron a cabo pruebas exhaustivas que incluyen transacciones simuladas, gestión de usuarios y escenarios de doble gasto. Los resultados probaron ser exitosos, demostrando la seguridad y confiabilidad de la implementación de los elementos bases que componen cualquier red Blockchain.