Prototipo blockchain para introducir integridad y acceso controlado en la recopilación de mediciones y simulaciones del proyecto LAGO

Gian Sebastián Mier Bello

Trabajo de Grado para Optar al Título de Ingeniero en Sistemas

Director

Luis Alberto Nuñez de Villavicencio Martinez

Doctor en Ciencias

Universidad Industrial de Santander

Facultad de Fisicomecanicas

Escuela de Ingeniería de Sistemas e Informática

Ingeniería en Sistemas

Bucaramanga

2025

Dedicatoria

Hago esta dedicatoria a mi madre, padre, hermano, tíos, primos y mi pareja que siempre estuvieron para mi apoyándome en mi carrera profesional y sin ellos no sería quien soy el día de hoy.

Agradecimientos

Quiero agradecer profundamente al profesor Carlos Jaime Barrios, quien siempre me apoyo incondicionalmente y a pesar de la distancia. A Alexander Martinez por su ayuda con el proyecto LAGO, A Robinson Rivas que me apoyo con concepto teóricos y técnicos de este proyecto. Y a Alejandro Torres por su apoyo y recomendaciones para la sustentación. Gracias a estas personas este proyecto pudo realizarse.

Tabla de contenido

Índice de Figuras

Lista de Abreviaturas

LAGO: Latín American Giant Observatory

DLT: Distributed Ledger Technology

TIC: *Information and Communication Technologies*

DAO: *Decentralized Autonomous Organization*

NFT: *Non-Fungible Token*

IoT: *Internet of Things*

GDPR: *General Data Protection Regulation*

KYC: *Know Your Customer*

AML: *Anti-Money Laundering*

CBDC: *Central Bank Digital Currency*

PoW: Proof of Work

PoS: Proof of Stake

TPS: Transactions Per Second

MSP: Membership Service Provider

Glosario

Blockchain: Tecnología de registro distribuido que permite almacenar datos de forma segura, inmutable y verificable sin necesidad de una autoridad central. En este trabajo, se usa para garantizar la integridad de los datos científicos.

Colaboración científica: Modelo de trabajo en red donde múltiples instituciones comparten recursos, datos y resultados, común en grandes proyectos como LAGO.

Hyperledger Fabric: Plataforma de blockchain de código abierto diseñada para entornos empresariales y redes privadas. Permite control de acceso, contratos inteligentes (chaincode) y modularidad en su arquitectura.

Integridad de datos: Propiedad que asegura que los datos no han sido alterados de manera no autorizada. Es esencial para garantizar la confiabilidad de los resultados científicos.

Trazabilidad: Capacidad de rastrear el origen, modificaciones y uso de un dato a lo largo del tiempo. Permite auditar y validar procesos en entornos colaborativos.

Resumen

Título: Prototipo blockchain para introducir integridad y acceso controlado en la recopilación de mediciones y simulaciones del proyecto LAGO*

Autor: Gian Sebastián Mier Bello**

Palabras Clave: Blockchain, Colaboración científica, Mediciones, Simulaciones, Integridad, Acceso controlado, Proyecto LAGO.

Descripción: La integridad de los datos científicos es esencial para garantizar la reproducibilidad y confiabilidad de los resultados en investigaciones colaborativas. En este trabajo, se presenta un prototipo basado en tecnología blockchain que permite registrar, verificar y acceder de forma segura a conjuntos de datos científicos generados por simulaciones y mediciones del Proyecto LAGO, una colaboración distribuida que utiliza recursos de cómputo de alto rendimiento (HPC). La solución se implementó utilizando Hyperledger Fabric, una plataforma de blockchain privada que permite control de acceso granular y trazabilidad de los registros. Esta herramienta no solo asegura la inalterabilidad de los datos, sino que también se integra con los flujos de trabajo HPC, evitando reprocesamientos innecesarios y optimizando el uso de recursos energéticos. El prototipo demuestra que es posible incorporar tecnologías descentralizadas en entornos científicos para fortalecer la integridad, seguridad y eficiencia en la gestión de datos.

* Trabajo de grado

** Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Luis Alberto Nuñez de Villavicencio Martinez

Abstract

Title: Blockchain prototype to introduce integrity and controlled access to the collection of measurements and simulations for the LAGO project*

Author(s): Gian Sebastian Mier Bello**

Key Words: Blockchain, Scientific Collaboration, Measurements, Simulations, Integrity, Controlled access, Project LAGO

Description: Scientific data integrity is essential to ensure the reproducibility and reliability of results in collaborative research. This work presents a blockchain-based prototype designed to securely register, verify, and access scientific datasets generated through simulations and measurements from the LAGO Project, a distributed collaboration that relies on high-performance computing (HPC) resources. The solution was implemented using Hyperledger Fabric, a private blockchain framework that enables fine-grained access control and record traceability. This tool not only guarantees data immutability but also integrates with HPC workflows, preventing unnecessary reprocessing and optimizing energy use. The prototype demonstrates the feasibility of integrating decentralized technologies into scientific environments to enhance data integrity, security, and efficiency in research data management.

* Final year project

** Physical-Mechanical Engineering Faculty. Systems and Informatics Engineering School. Advisor: Luis Alberto Nuñez de Villavicencio Martinez

Introducción

Con el auge de Internet y su gran proliferación, prácticamente todo se encuentra hoy en esta red, lo que expone los datos a riesgos constantes de manipulación o visualización sin autorización previa. Ante esta creciente vulnerabilidad, se hizo necesario desarrollar nuevas soluciones de seguridad; en este contexto, surgió la tecnología blockchain, concebida inicialmente como un sistema de transacciones punto a punto (P2P) para criptomonedas, pero cuyas propiedades —como la descentralización, la inmutabilidad y el cifrado— demostraron ser altamente efectivas para garantizar la integridad y la trazabilidad de los registros digitales. En este contexto, el presente trabajo propone desarrollar un prototipo basado en blockchain para introducir integridad y acceso controlado en los datos de las mediciones y simulaciones del Proyecto LAGO, empleando una red privada blockchain

El uso más conocido de blockchain es en redes como Bitcoin, que principalmente facilitan transferencias monetarias. Bitcoin permite un intercambio seguro de dinero digital sin intermediarios financieros, como afirma (Hirsh & Alman, 2020, p. 14). Sin embargo (Hirsh & Alman, 2020, p. 15) señala que el propósito subyacente de esta tecnología es proporcionar un registro inmutable que pueda probar propiedad y procedencia, características fundamentales para garantizar la confiabilidad de los datos científicos.

En la literatura, los autores (Piyoungkorn et al., 2022) hicieron una implementación de redes blockchain en un clúster HPC para la distribución y uso confiable de recursos computacionales basado en contratos electrónicos; sin embargo, en este caso se tomará en cuenta el entorno colaborativo científico del Proyecto LAGO

El propósito de este trabajo es desarrollar un prototipo blockchain que permita respaldar la integridad de los datos de mediciones y simulaciones del Proyecto LAGO, promoviendo su

conservación y reusabilidad a largo plazo. Se utilizará un enfoque que incluye el análisis del contexto del proyecto, la selección de tecnologías adecuadas, el diseño e implementación del prototipo. Se espera que el sistema resultante facilite la colaboración científica al proporcionar una base de datos que mantenga la integridad y el acceso controlado por las organizaciones 2 colaboradoras del proyecto LAGO.

1. Planteamiento y Justificación del Problema

El Proyecto LAGO es un observatorio extendido de rayos cósmicos compuesto de una red de detectores de agua Cherenkov (WCD) dispersos en diferentes sitios en diferentes longitudes y latitudes. Las mediciones recogidas de estos detectores son procesadas y analizadas en diferentes etapas. Adicionalmente, la colaboración genera datos simulados. El propósito final es permitir la curaduría y reuso de los datos fuera de la colaboración LAGO. Por lo que los conjuntos de datos deben conservarse para reprocesar si aparecen errores en el software más adelante. (LAGO, 2022)

La informática ha facilitado la realización electrónica de la ciencia(eScience), este enfoque está cambiando la forma en la que se ha hecho la ciencia hoy en día. La ciencia es cada vez más multidisciplinar, los datos son recolectados y analizados por científicos de muchas organizaciones propiciando un ambiente colaborativo (Fox & Kozyra, 2015, p. 1). En estos espacios se produce un intercambio abierto de datos, donde los investigadores trabajan en problemas complejos e integradores que requieren aportes desde múltiples perspectivas, cada campo de la investigación tiene sus propios métodos característicos y estilos científicos. Sin embargo, hay un tema que prevalece en todos los ámbitos: la primacía de los datos registrados, porque las técnicas que usan los investigadores para asegurar la integridad de sus datos son tan variadas como sus campos en sí (Kleppner, 2009)

Los apartados A.8 y A.9 de la norma (ISO/IEC, 2022) establecen la importancia de la gestión segura de activos y el control de acceso. El apartado A.8 destaca la necesidad de proteger los activos críticos, como los datos de mediciones y simulaciones, durante todo su ciclo de vida. En este contexto, la tecnología blockchain proporciona una estructura inmutable que garantiza la integridad de dichos datos, protegiéndolos contra modificaciones no autorizadas. Por otro lado, el apartado A.9 subraya la importancia de restringir el acceso únicamente a usuarios autorizados, lo

cual se logra mediante el uso de claves criptográficas y permisos específicos en la blockchain, asegurando así un acceso controlado.

Por esta razón, la implementación del prototipo blockchain en el proyecto LAGO es una propuesta adecuada para garantizar la integridad de los datos de mediciones y simulaciones en un entorno colaborativo y distribuido. Al utilizar una red blockchain privada, se asegura que los datos científicos de gran relevancia recolectados y generados por la colaboración LAGO sean inmutables, reduciendo el riesgo de manipulación o acceso no autorizado. Además, respaldará el objetivo del proyecto de permitir la curaduría a largo plazo y reuso de los datos.

2. Objetivos

2.1 Objetivo General

Desarrollar un prototipo basado en blockchain para introducir integridad y acceso controlado en la recopilación de mediciones y simulaciones del proyecto LAGO.

2.2 Objetivos Específicos

1. Diseñar un prototipo blockchain para el proyecto LAGO tras un análisis del contexto del proyecto y sus necesidades.
2. Implementar el prototipo blockchain diseñado para el proyecto LAGO, permitiendo almacenar los datos de mediciones y simulaciones de LAGO.
3. Realizar pruebas de validación sobre el prototipo blockchain para verificar que los mecanismos de integridad y control de acceso implementados funcionen correctamente.

3. Marco de Referencia

3.1. Marco teórico

3.1.1 Blockchain

Cuando se menciona blockchain, a menudo se piensa en las criptomonedas como Bitcoin, un activo digital que ha ganado notoriedad en la última década. Sin embargo, el concepto de blockchain trasciende este uso específico. Según (Hirsh & Alman, 2020, p. 14), el término fue introducido en 2008 por un desarrollador o grupo bajo el pseudónimo de Satoshi Nakamoto, quien propuso un sistema de intercambio de dinero electrónico entre pares. Esta tecnología, definida como una base de datos distribuida que almacena registros en bloques enlazados criptográficamente, garantiza la inmutabilidad de los datos almacenados (Xia et al., 2017, p. 3).

Además de su uso financiero, blockchain ofrece un mecanismo fundamental en redes descentralizadas: el algoritmo de consenso. Este mecanismo asegura que los nodos participantes acuerden la validez de los datos registrados, manteniendo la consistencia en toda la red (Xiong et al., 2022, p. 7). Esta capacidad de proporcionar integridad sin depender de intermediarios la convierte en una tecnología prometedora para entornos distribuidos como el Proyecto LAGO.

3.1.2 Smart Contracts

Los contratos inteligentes son programas que se ejecutan automáticamente en una red blockchain cuando se cumplen ciertas condiciones, y son procesados por los nodos de la red dependiendo de su mecanismo de consenso. Estos contratos eliminan la necesidad de intermediarios, ahorrando tiempo y recursos. En el contexto del proyecto LAGO, los smart contracts pueden ser utilizados para automatizar procesos como la validación y curaduría de datos, y el control de acceso de las mediciones y simulaciones. Por ejemplo, un contrato inteligente podría garantizar que un dato almacenado cumpla con los estándares establecidos antes de ser incorporado al sistema, (Kaushal et al., 2021, p. 3).

3.1.3 Integridad de los datos

La integridad de los datos es esencial en cualquier sistema de información, particularmente en proyectos científicos que requieren la preservación y confiabilidad de sus resultados. Según el estándar (ISO/IEC, 2022), la integridad implica proteger los datos contra modificaciones o alteraciones no autorizadas, asegurando que permanezcan completos y exactos durante todo su ciclo de vida. Esto se logra mediante mecanismos de control que no solo previenen cambios no deseados, sino que también permiten detectar alteraciones cuando ocurren.

En un entorno como el del Proyecto LAGO, donde los datos de mediciones y simulaciones son fundamentales para la investigación, la pérdida o manipulación de información podría comprometer no solo los resultados, sino también la reproducibilidad de los experimentos 5 científicos. Blockchain, con su estructura inmutable y sus capacidades de registro de auditoría, es una solución adecuada para garantizar que los datos permanezcan íntegros y confiables en todo momento.

3.1.4 Acceso controlado a los datos

En la gestión de datos sensibles, el acceso controlado es un principio fundamental. Según el estándar (ISO/IEC, 2022) (apartado A.9), este concepto se refiere a implementar políticas y mecanismos que permitan acceder a los datos únicamente a usuarios autorizados. Estos controles aseguran que cada usuario interactúe con los datos conforme a su rol y permisos específicos, minimizando los riesgos de acceso no autorizado.

En un proyecto colaborativo como LAGO, donde participan múltiples instituciones y científicos distribuidos geográficamente, es crucial establecer controles estrictos de acceso. Blockchain permite implementar estas restricciones de forma efectiva mediante el uso de claves criptográficas y permisos definidos, asegurando que solo los miembros autorizados puedan consultar o modificar los datos.

3.1.5 eScience y colaboración científica

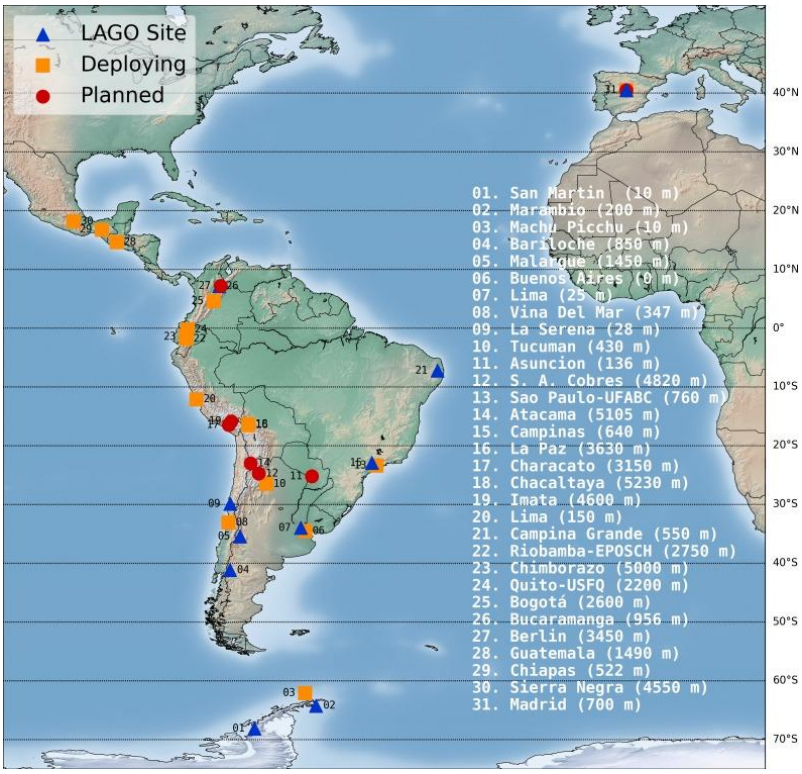
El concepto de eScience describe un enfoque colaborativo en el que se aprovecha el poder computacional distribuido para avanzar en la investigación científica. Según (Collins, 2010), este modelo permite a investigadores de diversas disciplinas trabajar juntos para resolver problemas complejos mediante el análisis de datos en entornos distribuidos. Sin embargo, (Preve, 2011) destaca que esta naturaleza distribuida también introduce desafíos significativos, como garantizar la integridad y seguridad de los datos compartidos entre múltiples organizaciones.

En el contexto de eScience, el uso de blockchain ha sido identificado como una solución innovadora para abordar estos desafíos. (Karastoyanova & Stage, "2018", p. 149) concluyen que esta tecnología permite no solo garantizar la integridad de los datos, sino también registrar información clave para la reproducibilidad de los experimentos, promoviendo la transparencia y confianza en los resultados científicos.

3.1.6 Proyecto LAGO

Figura 1

Mapa de los detectores WCD de la colaboración LAGO



El Proyecto LAGO (Latin American Giant Observatory) es un observatorio de astropartículas de alcance global que investiga fenómenos como el universo extremo, el clima espacial y la radiación atmosférica a nivel del suelo. Este proyecto involucra a más de 90 científicos de 25 instituciones en América Latina y España, quienes trabajan en un entorno altamente colaborativo y distribuido. (Donath, 2022)

El proyecto genera dos tipos principales de datos: mediciones recolectadas mediante detectores Cherenkov y datos simulados. Ambos tipos de información son esenciales para los análisis científicos y deben conservarse de forma íntegra a lo largo del tiempo. Además, es crucial

garantizar el acceso controlado a los datos para protegerlos de manipulaciones no autorizadas y, al mismo tiempo, facilitar la colaboración entre las instituciones participantes. (LAGO, 2022)

Blockchain se presenta como una solución ideal para abordar estos retos en el Proyecto LAGO, asegurando la integridad de los datos y estableciendo un sistema robusto de control de 7 acceso. Esto no solo respalda los objetivos del proyecto, sino que también contribuye a la reproducibilidad y confiabilidad de sus resultados científicos.

3.2 Estado del arte

En la revisión de literatura, se identificaron trabajos relacionados con el uso de blockchain para garantizar la integridad de los datos y promover ambientes colaborativos científicos. Sin embargo, no se encontró un enfoque que abarque simultáneamente los temas de blockchain, integridad de datos y respaldo de datos de mediciones y simulaciones en un contexto como el del Proyecto LAGO.

Un primer trabajo relevante es “Una arquitectura basada en blockchain para la confianza en la experimentación científica colaborativa” (Coelho et al., 2022). En este estudio, se implementó una red blockchain alojada en la nube con nodos distribuidos geográficamente para respaldar y generar confianza en ambientes colaborativos científicos. La solución aprovecha la inmutabilidad de los registros en blockchain para garantizar la procedencia de los datos y utiliza una aplicación descentralizada que recopila datos clave basados en un modelo de procedencia. Su caso de estudio se centró en datos genómicos del coronavirus, donde la gestión de accesos se realizó mediante certificados que garantizan la identidad de los colaboradores. Aunque este trabajo comparte nuestra visión respecto a la integridad de los datos, no aborda específicamente la necesidad de respaldar datos de mediciones y simulaciones en proyectos científicos como LAGO.

Otro trabajo relevante es “SciLedger: Procedencia de un flujo de trabajo científico y plataforma de intercambio de datos basada en blockchain” (Hoopes et al., 2022). Esta solución

implementa una blockchain para respaldar flujos de trabajo científicos, con características como el manejo de múltiples flujos y un mecanismo de invalidación. Utiliza un árbol de Merkle para verificar la procedencia de los datos y permite el acceso público al flujo de datos almacenados. Aunque se comparte la visión sobre garantizar la integridad de los datos, nuestro enfoque difiere al priorizar una red blockchain privada, donde los datos de mediciones y simulaciones del Proyecto LAGO se gestionen en un entorno controlado.

Finalmente, en "Procedencia de los datos en la observación de la tierra: Una solución basada en blockchain" (Zhang et al., 2024), se aborda la trazabilidad y el intercambio de datos relacionados con observaciones de la superficie y atmósfera terrestre. Debido al volumen de datos en petabytes, este trabajo propone un modelo de almacenamiento fuera de la cadena, utilizando la blockchain para guardar información clave mediante hashes. Además, utiliza un algoritmo de consenso basado en votación, optimizando el rendimiento y la eficiencia energética. Aunque se comparten similitudes en cuanto a garantizar la trazabilidad e integridad de los datos mediante una red privada, nuestro proyecto se enfoca específicamente en datos científicos del Proyecto LAGO, asegurando su respaldo en un entorno colaborativo que prioriza mediciones y simulaciones críticas para la investigación.

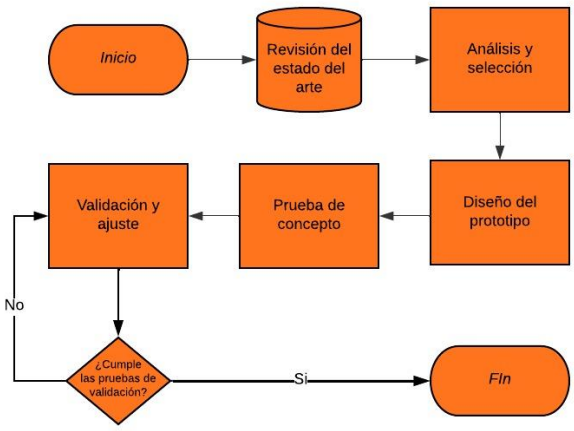
4. Metodología

4.1 Características de la metodología implementada

La metodología representada en la figura está basada en el modelo de "7 Phases of Blockchain Implementation"(CompTIA, 2021), adaptado a las necesidades específicas del proyecto. Comienza con la revisión del estado del arte, equivalente a la fase de descubrimiento, donde se investiga la tecnología y se identifican oportunidades del uso de blockchain. Luego, en la fase de análisis y selección, se evalúan y eligen las tecnologías y plataformas más adecuadas, alineándose con la etapa de evaluación de requisitos. El diseño del prototipo representa la fase de diseño de solución, seguido por la prueba de concepto, que se enfoca en la implementación del prototipo. La etapa de validación y ajuste corresponde a la implementación piloto y mejora continua, donde se refina el sistema según los resultados obtenidos. Finalmente, si la solución supera las pruebas de validación, se considera lista para una implementación completa.

Figura 2

Diagrama de flujo de la metodologia basada en "7 Phases Of Blockchain Implementation"



4.2 Revisión del estado del arte

Esta etapa consistió en explorar conceptos fundamentales sobre blockchain, integridad de datos y entornos colaborativos científicos, con el propósito de adaptarlos al proyecto. Este proceso permite identificar los mecanismos más adecuados para garantizar la integridad y acceso controlado de los datos en el contexto específico de la colaboración científica. A partir de este análisis, se establecen los fundamentos teóricos y técnicos que guían el desarrollo del sistema, asegurando su coherencia con las mejores prácticas y avances tecnológicos en la materia.

4.2.1 Investigación de fundamentos y conceptos básicos de blockchain

Blockchain es una tecnología de registro distribuido (DLT) que permite almacenar información de manera segura, inmutable y transparente. Conforme a (J. Li & Kassem, 2021), el término DLT hace referencia a un sistema de transacciones de valor ejecutado sobre una red entre pares (P2P), caracterizado por su naturaleza distribuida y la ausencia de una autoridad central que actúe como intermediaria. Blockchain, como una forma específica de DLT, surgió en 2008 como la base tecnológica de la criptomoneda Bitcoin, siendo la primera implementación exitosa de esta tecnología.

En términos simples, blockchain puede describirse como un libro de transacciones lineal y organizado en cadena, donde cada bloque está vinculado criptográficamente al anterior, formando una estructura que asegura la integridad de los datos almacenados. A diferencia de otras formas de DLT, que pueden no tener una estructura definida, blockchain establece un formato lineal y secuencial que garantiza la trazabilidad y la inmutabilidad de las transacciones registradas.

Los tipos de blockchain se clasifican según quién puede participar y cómo se accede a los registros almacenados. Existen dos criterios principales: si la red es con permisos o sin permisos

(basado en los requisitos de acceso) y si es pública o privada (basado en la visibilidad de las transacciones). Estas categorías generan cuatro combinaciones posibles:

1. **Pública y sin permisos:** abierta y transparente, cualquier persona u organización puede unirse y acceder a todas las transacciones, pasadas y presentes.
2. **Privada y sin permis**os: permite una participación amplia, pero solo las partes involucradas en las transacciones tienen acceso a los detalles.
3. **Pública y con permisos:** restringe quién puede unirse a la red, pero una vez aceptado, se obtiene acceso a todas las transacciones de la red.
4. **Privada y con permisos:** es la más restrictiva; limita tanto la membresía como el acceso a los datos, permitiendo a los participantes ver solo las transacciones en las que están involucrados (Dutta, 2020 - 2020).

En el sistema blockchain, las transacciones representan registros detallados de acciones e interacciones entre los miembros de la red, como unirse a la red, transferir activos (dinero, datos, etc.) o abandonar el sistema. Estas transacciones son almacenadas en unidades denominadas bloques. Cada bloque contiene las transacciones verificadas, un hash único y el hash del bloque anterior, asegurando así su conexión en una cadena de bloques inmutable. El hash, generado mediante una función matemática, protege la integridad de los datos y permite identificar alteraciones. Para manejar grandes volúmenes de datos, cada bloque utiliza una estructura de árbol de Merkle, que organiza eficientemente las transacciones y garantiza su autenticidad. De acuerdo con (Dutta, 2020 - 2020), estas características forman el núcleo matemático y computacional que otorga a la blockchain propiedades clave como la transparencia, la eficiencia en la verificación de transacciones y la inmutabilidad.

Los participantes o miembros del sistema blockchain son las personas u organizaciones que forman parte de la red descentralizada. Estos participantes, al compartir un interés común, como la transferencia de dinero, la integridad de una cadena de suministro o la gestión de activos, forman una red sin un ente central que ejerza control sobre las transacciones. Este modelo descentralizado garantiza la integridad e inmutabilidad de los datos, ya que cada transacción registrada se vuelve permanente y no puede ser modificada una vez validada por la mayoría de los participantes (Dutta, 2020 - 2020).

El mecanismo de consenso es fundamental en blockchain, ya que permite que la red descentralizada de nodos en un sistema blockchain asegure la validez de las transacciones y el orden de los bloques en el libro mayor distribuido. Dado que no existe una autoridad central, el consenso garantiza la integridad y la confiabilidad del sistema, permitiendo que todas las copias del libro mayor se mantengan idénticas. Este mecanismo permite a blockchain funcionar como una herramienta autosuficiente y segura para la gestión de datos en redes distribuidas (Yadav et al., 2023).

Existen varios tipos de mecanismos de consenso diseñados para satisfacer diferentes necesidades de las redes blockchain. A continuación, se presenta una breve descripción de algunos de ellos:

Proof-of-Work (PoW). Los mineros resuelven acertijos criptográficos mediante poder computacional para validar bloques y recibir recompensas. Este método es eficiente para redes públicas y abiertas, pero su alto consumo energético es una desventaja significativa (Yadav et al., 2023).

Proof-of-Stake (PoS). Los validadores son seleccionados en función de su participación económica (stake) en la red. Este método es más eficiente energéticamente que PoW y utiliza probabilidades para seleccionar al validador de cada bloque (Yadav et al., 2023).

Delegated Proof-of-Stake (DPoS). Basado en la democracia representativa, los participantes votan para elegir validadores que gestionan la creación de bloques. Este mecanismo fomenta la colaboración y distribuye las recompensas de manera más equitativa (Yadav et al., 2023).

Byzantine Fault Tolerance (BFT). Diseñado para tolerar fallas asociadas al problema de los Generales Bizantinos, este protocolo puede soportar hasta un tercio de nodos maliciosos en la red, asegurando el consenso incluso en entornos adversos (Yadav et al., 2023).

Proof-of-Authority (PoA). En este sistema, los validadores son seleccionados en función de su identidad verificable, como una reputación o documentos emitidos por el gobierno. Es eficiente y adecuado para redes con permisos (Yadav et al., 2023).

Proof-of-Elapsed Time (PoET). Utiliza un sistema de lotería aleatoria para asignar tiempos de espera a los validadores antes de crear bloques. Requiere hardware especializado como Intel SGX, y es energéticamente más eficiente que PoW (Yadav et al., 2023).

Proof of Retrievability (PoR). Asegura que los archivos almacenados en la red sean recuperables mediante un protocolo de desafío-respuesta. Es eficiente en términos de consumo energético y está diseñado para almacenamiento distribuido (Yadav et al., 2023).

Ripple Consensus Protocol/Algorithm (RCPA). Usado en la red Ripple, este protocolo permite a los nodos confiables (definidos por listas únicas de nodos) alcanzar el consenso mediante rondas de votación. Es eficiente y diseñado específicamente para transacciones financieras (Yadav et al., 2023).

IOTA (Tangle). Basado en un grafo acíclico dirigido (DAG) en lugar de bloques tradicionales. Este mecanismo es adecuado para aplicaciones del Internet de las Cosas (IoT) y micropagos debido a su alta escalabilidad y bajas tarifas (Yadav et al., 2023).

Stellar Consensus Protocol (SCP). Implementa un modelo de acuerdo bizantino federado, donde nodos confiables llamados quórums llegan a un consenso. Ofrece control descentralizado, baja latencia, confianza flexible y seguridad asintótica (Yadav et al., 2023).

Practical Byzantine Fault Tolerance (PBFT). Diseñado para redes privadas y con permisos, este mecanismo de consenso tolera actores maliciosos y fallos de red mediante un proceso intensivo de tres fases. Es altamente eficiente y utilizado en sistemas como Ripple y Stellar (Yadav et al., 2023).

Proof-of-Reputation (PoR). Utiliza la reputación de los participantes como garantía de honestidad. Las empresas actúan como validadores, y cualquier acto deshonesto conlleva consecuencias económicas y de imagen significativas (Yadav et al., 2023).

Crash Fault Tolerance (CFT). El mecanismo de consenso Crash Fault Tolerance (CFT) garantiza que un sistema distribuido pueda alcanzar consenso de manera correcta incluso si ciertos nodos fallan en la comunicación. En sistemas como CFT, este algoritmo puede tolerar fallos en hasta f nodos ($f = N-1\ 2$) requiriendo que cada nodo reciba al menos $f+1$ mensajes de verificación para completar la fase final del consenso (W. Li et al., 2023).

RAFT. El mecanismo de consenso RAFT, basado en la tolerancia a fallos por colapso (CFT), es ideal para redes blockchain privadas debido a su simplicidad y eficiencia. En estas redes, donde todos los nodos son confiables, RAFT se enfoca en manejar fallos de comunicación en lugar de nodos maliciosos. Los nodos se organizan en un líder, generalmente con mayores capacidades, y múltiples seguidores. El líder coordina las transacciones, y estas son aprobadas cuando la mayoría de los seguidores vota favorablemente. Este proceso asegura que el consenso se mantenga mientras la mayoría de los nodos funcione correctamente, incluso ante fallos de comunicación. Según (Xu et al., 2020), RAFT combina un diseño simple con un rendimiento comparable al de Paxos, haciéndolo ideal para redes privadas que requieren consensos rápidos y menos complejos.

Finalmente, los smart contracts son una innovación clave que amplía las capacidades de blockchain. Según (Dutta, 2020 - 2020), un contrato inteligente es un conjunto de reglas de negocio que se ejecuta automáticamente al cumplirse ciertas condiciones predefinidas, eliminando la necesidad de intermediarios tradicionales como abogados o servicios de depósito en garantía. Estos contratos están programados en la blockchain y utilizan la red de nodos para ejecutar acciones de forma transparente, eficiente y libre de conflictos, permitiendo la automatización y descentralización de las transacciones. Por ejemplo, un smart contract puede verificar que el saldo de una cuenta sea mayor que el monto de una transacción antes de ejecutarla, optimizando así el intercambio de valor en la red blockchain.

4.2.2 Revisión de investigaciones recientes, modelos de casos de uso y estudios de casos

Secure Reviewing and Data Sharing in Scientific Collaboration: Leveraging Blockchain and Zero Trust Architecture. (Pooja & Chandrakala, 2024) presentan una solución innovadora para mejorar la seguridad y confidencialidad en las colaboraciones científicas mediante la integración de blockchain y la arquitectura de confianza cero (Zero Trust). Proponen un sistema que utiliza smart contracts para proteger los manuscritos y comentarios en el proceso de revisión por pares, implementa un esquema ZT para evaluar y controlar el acceso a datos sensibles, y asegura la compartición de información mediante cifrado AES. Además, incorporan feedback de los autores para optimizar futuras selecciones de revisores. Su implementación técnica en Ethereum demuestra mejoras en la trazabilidad, automatización y transparencia del proceso, abordando problemas como manipulación, acceso no autorizado y falta de auditabilidad.

Este trabajo se distingue por combinar dos enfoques potentes (blockchain y Zero Trust) para abordar los problemas de seguridad en la colaboración científica, superando limitaciones de trabajos previos que solo consideraban la trazabilidad o inmutabilidad. Sin embargo, se alinea con

un enfoque más conceptual y experimental, dejando espacio para mejorar en la validación práctica y la implementación a gran escala. Aunque el modelo es prometedor, su aplicabilidad en entornos reales y su viabilidad económica requieren evaluaciones adicionales.

Blockchain-Based Secure Collaboration Platform for Sharing and Accessing Scientific Research Data. (Alniamy & Liu, 2020) propone una solución innovadora basada en blockchain y Attribute-Based Access Control (ABAC) para abordar los desafíos de seguridad y privacidad en la colaboración científica. Los autores plantean un sistema descentralizado utilizando Hyperledger Fabric, que permite a los investigadores compartir datos de manera segura, eficiente y transparente, otorgándoles control granular sobre el acceso y la gestión de sus datos.

La plataforma propuesta es una solución sólida y bien fundamentada para abordar los problemas de privacidad y seguridad en la colaboración científica. Al integrar blockchain, ABAC y técnicas criptográficas avanzadas, otorga a los investigadores control granular sobre sus datos mientras garantiza su confidencialidad y trazabilidad. No obstante, su implementación práctica podría enfrentarse a desafíos relacionados con la escalabilidad, los costos y la complejidad técnica.

Modelos de casos de uso y casos de estudio. (Dutta, 2020 - 2020) analiza cómo el blockchain puede transformar los registros de tierras, proponiendo marcos descentralizados que mejoran la transparencia en la adquisición y transferencia de propiedades. Se destacan tecnologías como Hyperledger y consensos basados en confianza para reducir fraudes, simplificar procesos complejos y garantizar la seguridad de los registros. Además, se aborda el uso de sistemas de archivo distribuidos y métodos como el e-stamp descentralizado para combatir problemas como la falsificación de documentos.

En el ámbito de la **gestión de cadenas de suministro**, (Dutta, 2020 - 2020) resalta la capacidad del blockchain para garantizar trazabilidad, prevenir pérdidas y optimizar procesos. La tecnología permite eliminar auditorías basadas en papel y mejorar la calidad del producto mediante

el monitoreo a lo largo del ciclo de vida. Ejemplos como AgriDigital y soluciones basadas en Ethereum demuestran cómo los contratos inteligentes y la integración con IoT pueden facilitar la transparencia y eficiencia en las cadenas de suministro.

En el sector salud, el blockchain aborda problemas críticos como la fragmentación de los **datos médicos** y la falta de interoperabilidad. (Dutta, 2020 - 2020) destaca el uso de registros distribuidos para compartir información de manera segura y transparente, permitiendo a los pacientes controlar sus datos. Soluciones como Medlock y frameworks ligeros mejoran la escalabilidad, la privacidad y la eficiencia en la gestión de datos médicos, mientras que el uso de consenso híbrido asegura la integridad de los registros.

En **finanzas**, el blockchain muestra un gran impacto en áreas como KYC, pagos transfronterizos, contratos inteligentes y crowdfunding. Según (Dutta, 2020 - 2020), las plataformas basadas en blockchain reducen costos, aumentan la transparencia y mejoran la eficiencia de los procesos financieros. Soluciones como sistemas descentralizados para verificación de identidad y la automatización de contratos ofrecen beneficios significativos para bancos y otras instituciones financieras.

En el ámbito de la **escasez digital**, el blockchain permite la gestión de derechos digitales en arte y medios, asegurando propiedad y trazabilidad. (Dutta, 2020 - 2020) destaca aplicaciones como Artchain y sistemas que utilizan contratos inteligentes para registrar y proteger activos digitales. Además, propone mecanismos que fomentan relaciones más directas entre artistas y consumidores, eliminando intermediarios y mejorando la distribución de beneficios.

Finalmente, en el área de **seguridad alimentaria,** (Dutta, 2020 - 2020) resalta la capacidad del blockchain para garantizar la trazabilidad de productos desde su origen hasta el consumidor. Esto facilita la detección rápida de contaminantes en casos de enfermedades transmitidas por

alimentos. La tecnología también se integra con IoT para mejorar la toma de decisiones y la gestión de datos, asegurando transparencia y confianza en la cadena de suministro alimentaria.

4.3 Análisis y selección.

Se llevó a cabo un estudio detallado de los objetivos y prioridades del proyecto LAGO, con énfasis en la curaduría y reutilización de datos a largo plazo, la implementación de políticas **FAIR** y la preservación de datos para su reprocesamiento y validación. Se analizaron los flujos de datos y los procesos de medición y simulación, identificando los distintos niveles de datos generados, su almacenamiento y las herramientas utilizadas para su gestión.

En esta fase también se evaluaron diferentes tecnologías blockchain con el fin de seleccionar la más adecuada para garantizar la **integridad, trazabilidad** y **acceso controlado** de los datos científicos. Se compararon varias soluciones con permisos, como **Corda, Ethereum Quorum, Ripple** y **Hyperledger Fabric**, considerando aspectos como privacidad, consenso, escalabilidad e interoperabilidad. Tras un análisis exhaustivo, se determinó que Hyperledger Fabric era la opción más adecuada para el proyecto, debido a su capacidad para gestionar **datos científicos** en un entorno seguro, su modularidad, la posibilidad de establecer canales privados y su compatibilidad con los principios **FAIR**, facilitando la transparencia y colaboración dentro de la comunidad científica.

4.3.1 Identificación de los objetivos y prioridades del proyecto LAGO

Tras un análisis del plan de gestión de datos del proyecto LAGO (Collaboration, 2022) se identificó lo siguiente:

Curaduría y reutilización de datos a largo plazo. El objetivo principal del proyecto LAGO es habilitar la curaduría y reutilización a largo plazo de los datos generados, tanto dentro como fuera de la colaboración LAGO, mediante el desarrollo de un Observatorio Virtual. Esto

incluye no solo los datos obtenidos a través de las mediciones de los detectores Cherenkov de agua (WCD), sino también los datos simulados generados por los investigadores.

Implementación de políticas FAIR. Como parte de las iniciativas promovidas por la Comisión Europea y en colaboración con el proyecto EOSC-Synergy (H2020), LAGO se compromete a implementar las políticas **FAIR**. Asegurando que sus datos sean localizables **(Findable)** mediante metadatos adecuados e identificadores persistentes, accesibles **(Accessible)** tras un periodo de embargo a través de plataformas abiertas, interoperables **(Interoperable)** gracias al uso de vocabularios y estándares reconocidos, y reutilizables **(Reusable)** bajo licencias abiertas que permiten su uso no comercial, garantizando al mismo tiempo su calidad y trazabilidad.

Esto garantiza que los datos generados y almacenados dentro del proyecto cumplan con estándares internacionales de accesibilidad y transparencia, promoviendo su uso más allá del ámbito académico.

Preservación de datos para reprocesamiento y validación. LAGO asegura la preservación de todos los conjuntos de datos generados, incluyendo datos brutos (L0), datos procesados (L1, L2, L3) y simulaciones (S0, S1, S2), con el fin de permitir su reprocesamiento en caso de errores de software o nuevas necesidades de investigación. Esta práctica permite no solo garantizar la integridad de los datos, sino también facilitar su validación y adaptabilidad a futuras metodologías de análisis. Este enfoque resulta esencial para la longevidad y el impacto del proyecto, ya que permite la comparación y actualización continua de los resultados, asegurando su utilidad para la comunidad científica internacional.

4.3.2 Análisis de flujos de datos y procesos de mediciones y simulaciones del proyecto LAGO

De acuerdo con el plan de gestión de datos del proyecto LAGO (Collaboration, 2022) el flujo de datos se organiza en diferentes capas, desde datos en bruto (L0) hasta datos simulados y analizados (S2). A continuación, se analizan las características principales de cada capa,

destacando las fuentes, tamaños, estándares de metadatos y herramientas utilizadas. Actualmente, existe una transición a un nuevo estándar de metadatos para el almacenamiento en un repositorio distribuido; sin embargo, solo capas como (S0) están disponibles, mientras que las otras aún se encuentran en servidores locales.

Datos en bruto (L0). Los datos en bruto (L0) son generados directamente por los detectores de agua Cherenkov (WCD) y constituyen el primer nivel de información capturada. Este nivel contiene las señales originales registradas por los fotomultiplicadores, que pueden alcanzar un tamaño de 150 GB al mes por detector. Estos datos son esenciales para el análisis inicial y la calibración, ya que representan el estado más puro de la información obtenida en los experimentos. En términos de estándares, actualmente emplean JSON para validación con GEANT4/ROOT y contienen metadatos básicos en XML/DublinCore según las especificaciones de ACQUA.

Datos preliminares (L1). Los datos preliminares (L1) son el resultado del procesamiento inicial realizado principalmente por investigadores. Este nivel procesa los datos en bruto en el framework ANNA, el cual corresponde al primer nivel de procesamiento, reduciendo su volumen a un rango de entre 70 y 120 GB al mes por detector. Esta etapa representa un paso intermedio crítico para estructurar la información de manera más manejable.

Datos de calidad para astrofísica y público (L2 y L3). Los datos de calidad (L2 y L3) son versiones refinadas y calibradas de los datos preliminares, correspondiendo al segundo y tercer nivel de procesamiento, respectivamente, del framework ANNA. El nivel L2 está enfocado en proporcionar datos listos para análisis astrofísicos, mientras que el nivel L3 adapta estos datos para fines de divulgación pública. Ambos niveles reducen significativamente el volumen de datos a entre 10 y 40 GB al mes por detector. Estas capas son esenciales para garantizar la calidad y utilidad de los datos tanto en investigación como en educación.

Datos simulados (S0). El nivel S0 abarca los datos simulados generados por herramientas computacionales como ARTI, CORSIKA y onedataSim. Estos datos replican las interacciones de partículas cósmicas en los detectores, proporcionando un marco teórico para validar y comparar las mediciones experimentales. Los tamaños de estos datos varían según la cantidad de simulaciones realizadas, alcanzando un máximo estimado de 120 GB al mes por usuario. Inicialmente, emplean metadatos en XML/DublinCore, pero se planea una transición a JSON-LD/DCAT-AP. Esta capa es clave para comprender fenómenos astrofísicos complejos y mejorar los modelos experimentales.

Datos analizados (S1 y S2). El nivel S0 abarca los datos simulados generados por herramientas computacionales como ARTI, CORSIKA y onedataSim. Estos datos replican las interacciones de partículas cósmicas en los detectores, proporcionando un marco teórico para validar y comparar las mediciones experimentales. Los tamaños de estos datos varían según la cantidad de simulaciones realizadas, alcanzando un máximo estimado de 120 GB al mes por usuario. Esta capa es clave para comprender fenómenos astrofísicos complejos y mejorar los modelos experimentales.

La organización del flujo de datos en el proyecto LAGO en diferentes capas (L0 a L3 para datos medidos y S0 a S2 para datos simulados) refleja un enfoque estructurado y progresivo para garantizar la calidad, accesibilidad y utilidad de la información. Desde los datos en bruto, que representan el estado más puro de las mediciones, hasta los datos simulados y analizados, cada nivel aporta un valor incremental al conocimiento científico y a la divulgación pública. Este enfoque no solo optimiza el manejo de grandes volúmenes de datos, sino que también asegura que los resultados sean validados y reutilizables en diversos contextos, promoviendo la colaboración científica interdisciplinaria.

En este contexto, la implementación de tecnologías como blockchain podría ser un pilar clave para cumplir con los objetivos del proyecto LAGO, especialmente en cuanto a la adopción de estándares como FAIR. Blockchain permitiría garantizar la integridad, trazabilidad y autenticidad de los datos en cada capa del flujo de información. Mediante la creación de un registro distribuido, sería posible asociar cada conjunto de datos con un identificador único (PID), asegurando que los metadatos cumplan con los estándares internacionales (como JSON-LD/DCAT-AP) y sean accesibles de manera segura para la comunidad científica.

Además, blockchain facilitaría el reprocesamiento de datos, ya que almacenaría un historial inmutable de cada etapa de procesamiento y validación. Esto garantizaría que los datos originales y las versiones procesadas puedan rastrearse, verificarse y utilizarse nuevamente en investigaciones futuras o para la validación de nuevos modelos experimentales. Asimismo, la transparencia y descentralización de blockchain permitirían que investigadores de diferentes regiones pudieran colaborar de manera más eficiente y confiable, alineándose con los objetivos a largo plazo del Observatorio Virtual de LAGO y su impacto global.

4.3.3 Evaluación y selección de la tecnologia blockchain más adecuada

Para cumplir con el requisito del proyecto LAGO de gestionar de forma privada su plataforma de datos científicos, se decidió limitar las opciones tecnológicas a soluciones de tipo blockchain con permisos. Este enfoque asegura un control exclusivo por parte de los miembros autorizados de la colaboración LAGO, permitiendo una administración segura y trazable de los datos. Las siguientes alternativas destacan:

Corda. Corda es una plataforma de registro distribuido creada por R3, diseñada específicamente para la industria de servicios financieros. A diferencia de otros sistemas blockchain que emplean un método de difusión global, Corda distribuye el registro de manera confidencial, lo que significa que solo las partes involucradas en una transacción tienen acceso a

los detalles de esta. Este enfoque añade una capa extra de privacidad, donde solo las contrapartes de la transacción pueden identificar a los participantes. Corda también emplea un servicio de validación de transacciones basado en un "Notario Simple", el cual permite una mayor eficiencia y escalabilidad al no requerir que todos los nodos validen cada transacción, lo que mejora el rendimiento respecto a los sistemas blockchain tradicionales (Raj & Deka, 2018).

El sistema está diseñado para ser fácil de integrar en las arquitecturas de tecnología empresarial existentes, permitiendo la adición de nuevos nodos sin cambios significativos en la red. Aunque Corda está basado en blockchain, es más una solución de software empresarial que un sistema blockchain totalmente descentralizado. Además, el uso de un modelo UTXO (Unspent Transaction Output) y la validación de las transacciones a través de un servicio de notario garantizan la integridad y la trazabilidad de los activos, a la vez que optimizan el rendimiento y la escalabilidad del sistema. No obstante, la dependencia de un servicio de notario único podría generar puntos de fallo, por lo que en implementaciones más grandes se recomienda usar un clúster de notarios para aumentar la resiliencia y la capacidad de transacciones.

Ethereum Quorum. Ethereum Quorum es una versión empresarial de Ethereum diseñada para redes con permisos que requieren transacciones rápidas, seguras y privadas entre participantes conocidos, como bancos de inversión. A diferencia de Ethereum público, Quorum soporta tanto transacciones públicas como privadas. Las transacciones privadas son gestionadas por el servicio de privacidad Çonstellation", que envía la carga útil de la transacción cifrada únicamente a los participantes involucrados, mientras que el resto de los nodos solo tiene acceso a un hash cifrado para referencia futura. Esto garantiza la privacidad y la posibilidad de validar la existencia de transacciones sin necesidad de confiar en las contrapartes (Raj & Deka, 2018).

Una característica distintiva de Quorum es el uso de Pruebas de Conocimiento Cero (ZKPs), que permiten validar saldos y transacciones sin revelar información confidencial, evitando

el doble gasto. Además, emplea "salting dinámico"para reforzar la seguridad de los datos frente a ataques cibernéticos, añadiendo una capa de defensa que dificulta el análisis de patrones por parte de los atacantes. Las instrucciones de pago públicas muestran el remitente y el destinatario, pero ocultan los montos de transferencia mediante valores hash. Los balances privados están protegidos en contratos privados y son accesibles únicamente por el propietario de la cuenta.

En términos de consenso, Quorum utiliza el modelo Raft, que difiere del sistema de prueba de trabajo (PoW) de Ethereum público. Raft implementa un líder por clúster que gestiona las entradas de registro, garantizando transacciones rápidas y alta tolerancia a fallos. A diferencia de Ethereum público, en el que cualquier nodo puede minar un bloque, en Quorum solo el líder de Raft puede crear bloques, lo que optimiza la eficiencia y proporciona inmutabilidad y finalización de transacciones. En futuras versiones, se espera la inclusión de mecanismos de consenso tolerantes a fallos bizantinos, mejorando aún más la resiliencia del sistema.

Hyperledger Fabric. Hyperledger Fabric es un marco de trabajo de blockchain empresarial de código abierto, alojado por la Fundación Linux, diseñado para ofrecer una arquitectura modular que permite integrar componentes como consenso, servicios de membresía y análisis de negocios de manera plug-and-play en infraestructuras corporativas existentes. Utiliza tecnología de contenedores estándar para alojar contratos inteligentes, conocidos como chaincode, que contienen la lógica de negocio y de aplicación del sistema. Inicialmente, Hyperledger Fabric fue contribuido por Digital Asset Group e IBM. (Raj & Deka, 2018) Hyperledger Fabric permite la creación de canales privados entre participantes específicos, garantizando la privacidad al compartir información solo con las partes involucradas. Cada canal mantiene un libro mayor independiente, limitando la visibilidad de las transacciones únicamente a los miembros del canal, ya sea en redes de gran escala o en canales bilaterales más específicos.

A diferencia de las blockchains públicas, Hyperledger Fabric implementa un enfoque integral al consenso que abarca desde la propuesta y el respaldo hasta el ordenamiento, validación y compromiso de transacciones. Este sistema incluye políticas de respaldo explícitas y verificaciones de versiones del libro mayor, garantizando integridad y evitando problemas como el doble gasto. El consenso está diseñado para ser modular y puede integrarse con sistemas distribuidos como Raft. Los nodos en Hyperledger Fabric asumen roles específicos: como, los cuales gestionan el ordenamiento de transacciones en bloques y su distribución a los nodos del canal. Endorsers, los cuales aprueban transacciones basándose en las políticas definidas.

Committers, que son los que validan y actualizan el libro mayor con las transacciones procesadas.

Ripple. Ripple es un sistema de liquidación bruta en tiempo real, red de intercambio de divisas y plataforma de remesas lanzado en 2012. Diseñada para pagos instantáneos y transacciones de crédito, su objetivo es ofrecer transacciones financieras globales rápidas, seguras y de bajo costo. A diferencia de criptomonedas como Bitcoin, Ripple se enfoca en instituciones financieras, posicionándose como un posible sucesor de SWIFT, el sistema global de pagos bancarios. Utiliza un protocolo de código abierto, un libro mayor de consenso y una criptomoneda nativa llamada XRP. (Raj & Deka, 2018)

La red de Ripple permite realizar pagos transfronterizos en segundos, eliminando los largos tiempos de espera del sistema tradicional, mientras ofrece visibilidad total y liquidaciones a bajo costo. Su modelo se basa en relaciones de crédito entre nodos, permitiendo que las transacciones rippleen.a través de rutas preestablecidas sin intervención humana. Esto la diferencia de las blockchains tradicionales, que agrupan transacciones en bloques.

Ripple es una red modular compuesta por gateways (entidades como bancos que autentican nuevos nodos), market makers (carteras que facilitan el intercambio de divisas) y usuarios.

Además, permite manejar múltiples tipos de crédito, como monedas fiduciarias, criptomonedas y valores personalizados, optimizando el uso de capital y liberando recursos que actualmente están bloqueados en cuentas prefinanciadas.

El proceso de consenso iterativo de Ripple asegura que las transacciones sean validadas de manera rápida y segura. Sin embargo, enfrenta retos como el riesgo de calidad de crédito durante las transacciones y la dependencia de nodos clave en regiones emergentes. Estas limitaciones pueden mitigarse mediante configuraciones adecuadas y mayor control en las carteras.

Tabla 1

Comparación de plataformas blockchain

Características	Hyperledger Fabric	Ripple	Corda	Quorum
Tipo de red	Con permisos, diseñada para redes empresariales.	Híbrida: liquidación, cambio de divisas y remesas.	Con permisos, enfocada en redes financieras empresariales.	Con permisos basada en Ethereum, con control granular.
Privacidad	Canales privados para transacciones confidenciales.	Red de confianza basada en relaciones de crédito.	Transacciones visibles solo para partes involucradas.	Transacciones privadas mediante nodos y Tessera.
Consenso	Modular: Kafka, Raft, políticas de endoso.	Descentralizado, nodos validadores, sin minería.	Validación por notaries, sin consenso global.	Algoritmos alternativos (Raft, IBFT), rápidos y eficientes.
Smart contracts	Soporte modular con chaincode.	No aplica.	JVM (Java, Kotlin), alineados a normativas legales	Compatible con Solidity y la EVM.
Interoperabilidad	Modular, integración con sistemas empresariales.	Compatible con sistemas financieros (SWIFT).	Compatible con sistemas heredados y otras plataformas.	Compatible con herramientas Ethereum como MetaMask.
Escalabilidad	Configurable para múltiples canales y participantes.	Limitada por gateways en algunas regiones.	Optimizado para transacciones bilaterales.	Rendimiento superior gracias a consenso eficiente.
Enfoque principal	Redes empresariales en varios sectores.	Instituciones financieras, pagos internacionales.	Contratos legales, banca y comercio.	Empresas que requieren privacidad y alto rendimiento.
Manejo de datos	Modular con servicios de membresía y analítica.	Basado en XRP o caminos de crédito.	Garantiza privacidad selectiva en transacciones.	Tessera para cifrar datos privados.
Automatización	Smart contracts para reglas de negocio.	Limitada a liquidaciones y remesas.	Flujo automatizado basado en contratos legales.	Compatible con herramientas de Ethereum.
Rendimiento	Alta disponibilidad con servicios redundantes.	Rápido para pagos internacionales, retrasos regionales posibles.	Alta capacidad para transacciones simultáneas.	Eficiente para grandes volúmenes gracias al consenso.
Aplicaciones principales	Sectores empresariales generales.	Finanzas, remesas y pagos globales.	Instituciones financieras y sectores legales.	Finanzas, logística e industrias con Ethereum.

Hyperledger Fabric es la mejor alternativa para el proyecto LAGO debido a su enfoque en redes con permisos, lo que garantiza que solo los participantes autorizados accedan a los datos, preservando la privacidad y la seguridad de la información científica. Su naturaleza de código abierto facilita la personalización y el desarrollo colaborativo, permitiendo adaptar la plataforma a las necesidades específicas del proyecto sin costos adicionales de licencias. Además, su capacidad para configurar canales privados entre participantes asegura que los datos sensibles puedan manejarse de manera confidencial, mientras que sus políticas de consenso y automatización mediante smart contracts mejoran la eficiencia operativa al eliminar redundancias. En términos de costos, Hyperledger Fabric resulta competitivo ya que elimina la necesidad de minería intensiva, lo que reduce significativamente los gastos operativos relacionados con energía y hardware. También permite personalizar la red para ajustarse a los requisitos específicos del proyecto, optimizando recursos y asegurando una inversión sostenible a largo plazo.

4.4 Diseño del prototipo

En esta fase, se estableció la arquitectura y los mecanismos de acceso del prototipo blockchain del proyecto LAGO. Se definió un sistema de control basado en Hyperledger Fabric, utilizando certificados X.509 y políticas de permisos. La infraestructura incluyó nodos **pares, servicio de ordenamiento** y **chaincode**, asegurando un flujo seguro de transacciones. Además, se diseñó un esquema de almacenamiento de datos con identificadores únicos y funciones hash para garantizar la integridad y trazabilidad. Finalmente, los datos fueron procesados y estructurados en JSON, optimizando su almacenamiento en la blockchain y facilitando su verificación y acceso.

Establecimiento de mecanismos de acceso controlado al prototipo

El acceso controlado al prototipo blockchain del proyecto LAGO se implementará utilizando el sistema de Proveedores de Servicios de Membresía (MSP, por sus siglas en inglés) de Hyperledger Fabric (Hyperledger, 2020a). Este servicio asegura que solo actores autenticados y confiables puedan interactuar con la red, cumpliendo con los requisitos de privacidad, seguridad y escalabilidad del proyecto. A continuación, se describen los mecanismos clave:

Identidades digitales basadas en certificados X.509. Cada participante de la red blockchain tendrá una identidad digital única que será representada por un certificado X.509. Estos certificados actúan como una "tarjeta de identificación"que incluye información específica sobre el participante, como su nombre, organización, unidad organizativa y rol dentro de la red. Estos certificados son emitidos por una Autoridad Certificadora (CA) confiable, que asegura la validez y autenticidad de las identidades. Este enfoque garantiza que solo los actores autenticados puedan interactuar con la red, evitando accesos no autorizados y estableciendo un estándar de confianza entre los participantes.

Principales como uniones de identidad y permisos. En Hyperledger Fabric, un principal es la combinación de la identidad digital de un participante con atributos específicos que definen sus permisos en la red. Estos principales pueden incluir propiedades como la organización a la que pertenece el actor, su rol dentro de la red (por ejemplo, administrador, lector o validador) y otros atributos relevantes. Este sistema permite configurar reglas avanzadas de acceso, ajustando los permisos según el contexto y las necesidades del proyecto LAGO, proporcionando flexibilidad y control granular sobre los recursos de la red.

Control de acceso basado en políticas. El Proveedor de Servicios de Membresía (MSP) permite definir políticas de control de acceso que determinan qué acciones pueden realizar los participantes dentro de la red blockchain. Estas políticas establecen quién puede leer o escribir datos, ejecutar contratos inteligentes, o realizar cambios administrativos en la red. Por ejemplo, las políticas pueden restringir la capacidad de un usuario para interactuar con determinados canales de datos o limitar las operaciones de escritura solo a ciertos roles. Este enfoque garantiza que las acciones dentro de la red sean consistentes con los permisos asignados a cada participante.

Autoridad Certificadora (CA) para la emisión y gestión de Certificados. La CA se encargará de emitir los certificados X.509 para nuevos usuarios, renovar los certificados existentes cuando sea necesario y revocar aquellos que hayan sido comprometidos o ya no sean válidos. Este proceso asegura que solo los participantes autorizados puedan interactuar con la red y que las identidades comprometidas sean desactivadas de inmediato.

Basado en este mecanismo MSP, se plantean las siguientes identidades.

Espectador. Esta identidad hace referencia a cualquier usuario que tenga acceso a los datos de la colaboración LAGO sin permisos de escritura al estado del blockchain, pero sí de lectura a la red blockchain.

Colaborador. Esta identidad hace referencia a los miembros de la colaboración LAGO; pueden modificar los datos de las mediciones y simulaciones.

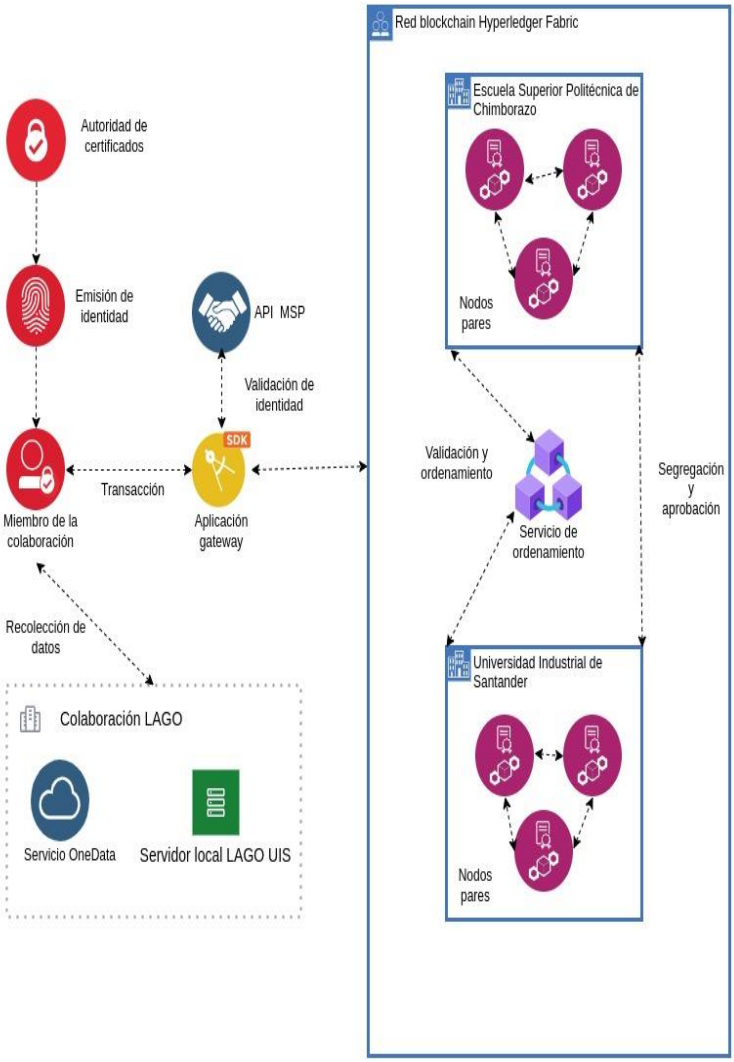
Administrador. Es el responsable de mantener los nodos pares de la blockchain y gestionar el ciclo de vida del chaincode en los nodos pares.

El uso de MSP en Hyperledger Fabric proporciona un alto nivel de control sobre los mecanismos de acceso. Además, la capacidad de definir permisos personalizados y gestionar identidades digitales de manera eficiente reduce significativamente los costos operativos en comparación con otras soluciones de autenticación. Estos mecanismos se adaptan perfectamente a las necesidades del proyecto LAGO, garantizando acceso controlado en el prototipo blockchain.

4.4.1 Modelado de la arquitectura del prototipo blockchain

Figura 3

Arquitectura del prototipo blockchain



El prototipo blockchain mostrado en la imagen utiliza Hyperledger Fabric como su plataforma base. Se tuvieron en cuenta dos organizaciones integrantes de la Colaboración LAGO: **Universidad Industrial de Santander y Escuela Superior Politécnica de Chimborazo**, dado que ambas contaban con los datos necesarios para la prueba.

Para soportar el protocolo Raft basado en Crash Fault Tolerance (CFT) se escogió un número impar de nodos por organización. A continuación, se detalla la descripción de los componentes y el flujo de transacciones.

Autoridad de certificados. Es responsable de emitir las identidades digitales basadas en MSP (Membership Service Provider) para los participantes de la red. Estas identidades están encapsuladas en certificados X.509, necesarios para interactuar con la red.

Chaincode. El chaincode define la lógica de negocio y las reglas de la red blockchain, gestionando pares clave-valor que representan el estado inicial y las transacciones del sistema. A diferencia de los contratos inteligentes tradicionales, el chaincode opera en redes privadas autorizadas dentro de los nodos y no en máquinas virtuales descentralizadas, bajo políticas de endoso específicas que garantizan seguridad y control.

Aplicación gateway (SDK). Es el puente entre los usuarios (miembros de la colaboración) y la red blockchain. Facilita el envío de propuestas de transacciones y la validación de identidades mediante la API del MSP. Las transacciones hacen referencia a las funciones del Chaincode que se ejecutan en los nodos.

Servicio de ordenamiento. Se encarga de recibir las transacciones propuestas por los usuarios, asignarles un orden cronológico y distribuir los bloques resultantes a los nodos pares en ambas organizaciones.

Nodos pares. Cada organización tiene tres nodos pares que validan las transacciones según las políticas de respaldo (endorsement policies) y almacenan la copia del libro mayor (ledger) actualizado. De esta forma se asegura una descentralización de los datos registrados.

Colaboración LAGO. Representa la integración de los sistemas de recolección de datos, como Servicio OneDataHUB y el servidor local LAGO UIS, el cual recibe los datos de las

mediciones (L0), con la red blockchain. Los datos de las mediciones y transacciones por los miembros de la colaboración se procesan y almacenan mediante transacciones.

El flujo de transacciones comienza con la emisión de identidades digitales a los miembros de la colaboración. Una Autoridad de Certificados (CA), parte esencial del sistema, genera certificados X.509 que contienen las credenciales necesarias para autenticar a cada usuario. Estas credenciales permiten que los participantes interactúen de forma segura con la red blockchain, garantizando que solo actores autorizados puedan iniciar transacciones o consultar información.

Un miembro de la colaboración, utilizando su identidad digital, envía una solicitud de transacción a través de la aplicación gateway. Esta aplicación actúa como intermediario entre el usuario y la red blockchain. Antes de procesar la solicitud, la aplicación valida la identidad del usuario utilizando el sistema MSP (Membership Service Provider), asegurándose de que el actor cuenta con los permisos necesarios. Si la validación es exitosa, se genera una propuesta de transacción firmada digitalmente, que luego es enviada a los nodos pares de las organizaciones participantes.

La propuesta de transacción llega a los nodos pares (peers) de las organizaciones, tanto de la Universidad Industrial de Santander como de la Escuela Superior Politécnica de Chimborazo. Los nodos ejecutan la propuesta de forma simulada, sin realizar cambios permanentes en el libro mayor (ledger).

Durante esta simulación, se generan dos conjuntos:

- Read Set: Representa los datos leídos del estado actual.
- Write Set: Define los cambios propuestos al estado

Si la transacción cumple con las políticas de respaldo (endorsement policies), cada nodo que aprueba la transacción devuelve una respuesta firmada al usuario a través de la aplicación

gateway. Estas políticas aseguran que se requiera el respaldo de nodos específicos antes de proceder.

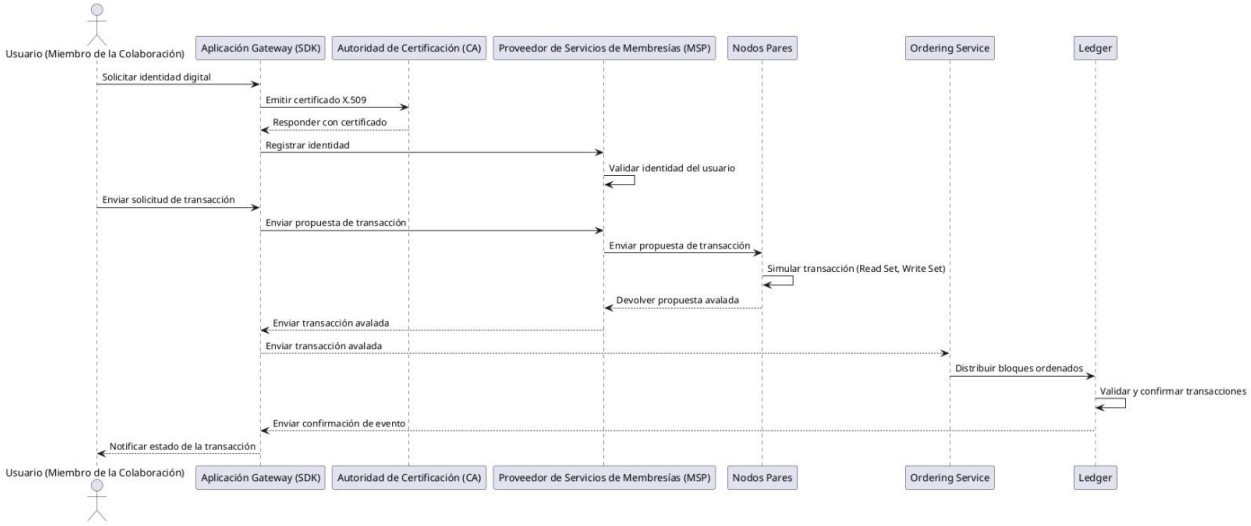
Una vez obtenidas las respuestas necesarias de los nodos que respaldaron la transacción, la propuesta es enviada al **servicio de ordenamiento.** Este componente organiza las transacciones recibidas, asignándoles un orden cronológico y agrupándolas en bloques. Estos bloques representan un conjunto de transacciones listas para ser procesadas por los nodos pares.

Una vez generados los bloques por el servicio de ordenamiento son distribuidos a todos los nodos pares en la red, en los cuales se validan las transacciones, comprobando que las transacciones cumplen con las políticas de respaldo y que no existen conflictos en los datos leídos y escritos (por ejemplo, intentos de doble gasto). Luego, las transacciones válidas son aplicadas al libro mayor (ledger) registro de todas las transacciones, actualizando de forma permanente el estado del mundo (world state) el cual es el estado actual de los datos. Las transacciones inválidas son registradas como rechazadas, y sus cambios propuestos no son aplicados.

Finalmente, los nodos pares generan un evento de confirmación para notificar al usuario sobre el estado de su transacción. Este evento informa si la transacción fue validada y comprometida exitosamente o si fue rechazada debido a conflictos o incumplimiento de las políticas de la red. La aplicación gateway recibe esta notificación y la muestra al usuario, completando así el flujo de transacción. (Hyperledger, 2020b)

Figura 4

Diagrama de secuencia del flujo de transacciones



4.4.2 Diseño de esquemas para el almacenamiento de datos y metadatos clave

Figura 5

Esquema de datos de mediciones y simulaciones

Scientific Data	
id	String
type	String
generationDate	String
metadata	{ hash: Hash location: String }
rawData	{ hash: Hash location: String }
inputData	{ hash: Hash location: String }
inputMetadata	{ hash: Hash location: String }
outputData	{ hash: Hash location: String }
outputMetadata	{ hash: Hash location: String }
siteName	String
collaboratorName	String
orcId	String
accessUrl	String

El esquema de datos presentado corresponde a la estructura de información genérica que se registrará en la red blockchain de Hyperledger Fabric para almacenar simulaciones y mediciones científicas del proyecto LAGO. A continuación, la descripción de cada campo:

Id. Identificador único del registro, esencial para localizar y referenciar cada transacción en la blockchain; este corresponde a la convención de nombres existentes en el DMP (Collaboration, 2022).

Type. Especifica el tipo de dato (por ejemplo, simulación o medición) y su nivel (L0, L1, S0, S1, etc).

Metadata. Hash de los metadatos asociados al dato crudo y la localización del archivo.

Raw data. Hash de los datos crudos obtenidos en las mediciones o simulaciones y la localización del archivo.

Input data. Hash de los datos de entrada utilizados en las simulaciones y la localización del archivo.

Input metadata. Hash de los metadatos asociados a los datos de entrada y la localización del archivo.

Output data. Hash de los resultados generados tras el procesamiento de los datos crudos y la localización del archivo.

Output metadata. Hash de los metadatos vinculados a los resultados y la localización del archivo.

Site name. Nombre del sitio donde se realizó la medición o simulación.

Collaborator name. Nombre del colaborador responsable o asociado a los datos.

ORC id. Identificador ORCID del colaborador, proporcionando un vínculo único y verificable.

Access url. URL de acceso a los datos o información relacionada almacenada externamente.

Figura 6

Ejemplo de dato científico

```
{
    "Id": "SO_bga_10_77402_QGSII_flat_defaults_DAT000703",
    "type": "SO",
    "generationDate": "2021-04-16T10:51:59.484880Z",
    "metadata": {
        "hash": "0d306e78140af8100682db4a6a9493152a492b5845e663df8625320a372cf246",
        "location": "/SO_bga_10_77402_QGSII_flat_defaults/.metadata/.DAT000703.bz2.jsonld.20220701T110100.291889Z"
    },
    "rawData": {
        "hash": "d02d21a138f4ca5eb2cf23873c45e0cd2b6bada836155616206ce3bbb77e3026",
        "location": "/SO_bga_10_77402_QGSII_flat_defaults/DAT000703.bz2"
    },
    "inputData": {
        "hash": "47e8ceb1d194c19708cf7953560ef4301b752e5f3eb7c5962afa5b473553fc46",
        "location": "/SO_bga_10_77402_QGSII_flat_defaults/DAT000703-0703-00000000024.input"
    },
    "inputMetadata": {
        "hash": "30be9443fdcf95c935ae9401bf34d3e4fe947d33b693e0ae8807c7e9a2e3b934",
        "location": "/SO_bga_10_77402_QGSII_flat_defaults/.metadata/.DAT000703-0703-00000000024.input.jsonld.20220701T110055.716745Z"
    },
    "outputData": {
        "hash": "c5c1ac3c1d2dd730d72836354024a7cbdbe05c80ca605b150884f98bb0b550d0",
        "location": "/SO_bga_10_77402_QGSII_flat_defaults/DAT000703-0703-00000000024.1st.bz2"
    },
    "outputMetadata": {
        "hash": "b4540c82a72d2413a9ee7366ba6a3a5034d9dad230c0ef8e8d8f018b1e2934bc",
        "location": "/SO_bga_10_77402_QGSII_flat_defaults/.metadata/.DAT000703-0703-00000000024.1st.bz2.jsonld.20220701T110057.934361Z"
    },
    "siteName": "bga",
    "orcid": "https://orcid.org/0000-0001-6497-753X",
    "accessUrl": [
        "http://hdl.handle.net/21.12145/gi9Hc9c",
        "https://datahub.egi.eu/share/37b0a0d99cbae5791db93ababbef5b90chccda"
    ]
}
```


4.4.3 Curación y tratamiento de los datos de mediciones y simulaciones

El proceso inició con la extracción de un subconjunto de datos provenientes de dos fuentes principales. Por un lado, las mediciones almacenadas en un servidor local del proyecto LAGO fueron seleccionadas y descargadas. Por otro lado, las simulaciones se obtuvieron desde el repositorio centralizado de One Data Hub, una plataforma destinada al almacenamiento y acceso de datos científicos. Estas dos fuentes proporcionaron la información base para ser procesada.

Los datos extraídos se procesaron utilizando scripts de Python, diseñados para convertir las mediciones y simulaciones en una estructura estándar y ligera de tipo JSON. Durante este procesamiento, se analizaron los archivos de entrada, salida y sus respectivos metadatos. Para garantizar la integridad de los datos, se generaron identificadores únicos mediante la función de hash SHA-256, aplicados a cada uno de los archivos y su contenido al formato original. Este

enfoque permite verificar que los datos no hayan sido alterados durante o después del procesamiento.

El resultado del procesamiento consistió en archivos JSON que siguen una estructura predefinida. Cada JSON incluye un identificador único (Id), el tipo de dato (type), y la fecha de generación (generationDate). También se registran los hashes de los archivos procesados y sus metadatos, como metadata, rawData, inputData, entre otros. Adicionalmente, se incluyen datos contextuales como el nombre del sitio de medición (siteName), el identificador ORCID del colaborador (orcid), y la URL de acceso al repositorio (accessUrl). Este diseño asegura que los datos sean fáciles de interpretar y verificar.

La estructura JSON creada fue pensada para ser compatible con una red blockchain. Esto garantiza que tanto las mediciones como las simulaciones puedan almacenarse de forma descentralizada, ofreciendo trazabilidad, transparencia y seguridad en el manejo de la información. De este modo, los usuarios interesados pueden acceder a los datos procesados y verificar su autenticidad mediante los hashes SHA-256 generados durante el procesamiento.

Finalmente, los datos procesados y los scripts utilizados para su transformación fueron almacenados en un repositorio de GitHub. Este repositorio, accesible en[1], contiene los JSON generados, así como el código utilizado para transformar los datos. Este almacenamiento centralizado facilita el acceso a los datos procesados y permite que otros colaboradores repliquen el proceso o lo adapten a nuevas necesidades.

[1] Link del repositorio de scripts y datos procesados https://github.com/GSMier/LAGO-data

4.5 Prueba de concepto

En esta etapa, se implementó el prototipo blockchain en **Hyperledger Fabric** utilizando una infraestructura basada en **Docker y Node.js**. Se configuraron identidades digitales mediante certificados X.509 para garantizar el acceso controlado en la red. Se estableció un canal de comunicación entre organizaciones y se desplegó el chaincode 'ScientificDataCollectContract', que permite gestionar registros científicos con control de acceso basado en roles.

La red blockchain se configuró con nodos pares y ordenadores, asegurando la integridad de las transacciones. Se desarrolló una aplicación **gateway** en **Node.js** para interactuar con la red y verificar la integridad de los datos mediante cálculos de hash. Finalmente, se almacenaron datos de prueba en la blockchain y se monitorizaron las transacciones mediante **Hyperledger Explorer**, validando la funcionalidad del sistema con más de 400 registros científicos procesados exitosamente.

4.5.1 Implementación del prototipo basado en la tecnología seleccionada

Para la implementación del prototipo en Hyperledger Fabric se dispone de una máquina x86_64 GNU/Linux Debian 6.1.128-1 con 7.51 GB de RAM y 8 núcleos, junto a las herramientas: **Docker y Docker Compose**[2] esenciales para la ejecución de los diferentes servicios de la red como contenedores con las **imágenes** de las autoridades de certificados (fabric-ca), los pares (peer), las bases de datos noSQL CouchDB (couchdb) y ordenadores (). Los **binarios**[3] de Hyperledger Fabric como (fabric-ca-client, configtxgen, peer, , etc.), que mediante la ejecución de **shell scripts** permitirán la creación y configuración de la red junto al chaincode. Adicionalmente se usará

[2] Docker y Docker Compose son herramientas para gestionar los contenedores: https://www.docker.com/

[3] La informacion de los binarios e imagenes de Docker se encuentra en: https://hyperledger-fabric.readthedocs.io/en/latest/install.html

Node.js para la creación del chaincode inmerso en los pares y el **Fabric SDK4**[4] de Node.js para la creación de la aplicación que se va a conectar a la red.

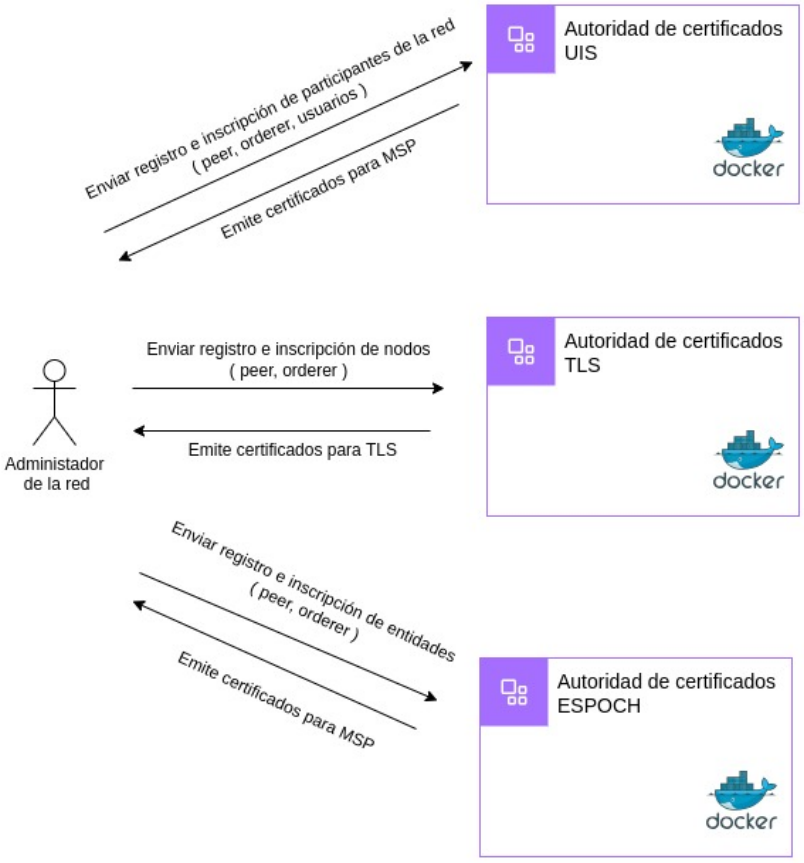
Generación de identidades. Para garantizar la identidad y autenticación de las entidades participantes en la red, se generan certificados digitales mediante Hyperledger Fabric CA (Certificate Authority). Cada organización en la red obtiene su propia autoridad certificadora para emitir credenciales a sus usuarios y nodos.

En esta etapa, se configuran las variables de entorno de las entidades certificadoras (CA) y nodos de la red, necesarias para desplegar los servicios en contenedores y generar los certificados X.509 que validarán la identidad de cada participante en la blockchain. A su vez, se garantiza una comunicación TLS, por lo cual se destina un CA para la comunicación TLS, el cual solo será para los nodos pares y ordenadores.

[4] https://hyperledger.github.io/fabric-sdk-node/

Figura 7

Generación de identidades



Una vez configurados los CA, se levantan los servicios (CA) mediante un compose de Docker[5]. Con las autoridades de certificados en ejecución, se utiliza un shell script[6] que genera los certificados de autoridad, claves privadas y certificados firmados (cacerts, keystore, signcerts, etc.) a las entidades (peer, , admin y usuarios) de las organizaciones UIS y ESPOCH, que serán usados

[5] Archivo Compose de los CA:

https://github.com/GSMier/ProyectoLago/blob/main/channel/certificates/docker-compose.yaml

[6] Scrip para generar los certificados:

https://github.com/GSMier/ProyectoLago/blob/main/channel/certificates/create-certificates.sh

para MSP y la comunicación TLS. Los certificados generados[7] en esta fase son esenciales para la autenticación y autorización dentro de la red blockchain.

Creación del canal. En Hyperledger Fabric, la comunicación entre organizaciones se realiza a través de canales. Se configura el archivo configtx.yaml[8] , donde se definen las organizaciones participantes con la configuración MSP, el mecanismo de consenso dentro del servicio de ordenamiento y sus respectivas políticas de gobernanza en la red, como quién puede escribir, leer o aprobar transacciones, basado en el rol de la identidad MSP, ejemplo de configuración en la figura 8. A partir de esta configuración, se genera el bloque génesis (genesis.block) con el siguiente script[9], que será el punto de inicio de la red blockchain.

La red blockchain en Hyperledger Fabric se compone de nodos pares (que almacenan la información en bases de datos en memoria **LevelDB** o en noSQL **CouchDB** y validan transacciones) y nodos ordenadores (que ordenan y distribuyen transacciones). Se configuran los archivos de cada nodo de las organizaciones con sus respectivos certificados e identidades digitales. Luego, con el Docker Compose[10] se levantan los nodos en contenedores de Docker.

[7] Certificados generados y configuracion MSP:

https://github.com/GSMier/ProyectoLago/tree/main/channel/crypto-config

[8] Archivo de configuracion de la red:

https://github.com/GSMier/ProyectoLago/blob/main/channel/configtx.yaml

[9] Script para la creación del bosque génesis:

https://github.com/GSMier/ProyectoLago/blob/main/artifacts/lagochannel.block

[10] Archivo compose para el levantamiento de los nodos de la red:

https://github.com/GSMier/ProyectoLago/blob/main/docker-compose.yaml

Figura 8

Políticas de la organización UIS

```
- &UIS
    Name: UISMSP
    ID: UISMSP
    MSPDir: crypto-config/peerOrganizations/uis/msp
    Policies:
        Readers:
            Type: Signature
            Rule: "OR('UISMSP.admin', 'UISMSP.peer', 'UISMSP.client')"
        Writers:
            Type: Signature
            Rule: "OR('UISMSP.admin', 'UISMSP.client')"
        Admins:
            Type: Signature
            Rule: "OR('UISMSP.admin')"
        Endorsement:
            Type: Signature
            Rule: "OR('UISMSP.peer')"
```

En esta etapa, los nodos pares y ordenadores inician sus operaciones y quedan listos para interactuar en la red. Después, a partir del bloque génesis, se crea el canal[11] y cada nodo se une a él para permitir la comunicación y ejecución de transacciones de manera segura.

Despliegue del chaincode. Una parte importante de la red blockchain son las transacciones, que en este caso se realizan a través del chaincode, el equivalente a los contratos inteligentes en Hyperledger Fabric. Para este prototipo, se desarrolla un chaincode en Node TypeScript con fabric-contract-api. Se implementa un control de acceso basado en atributos, permitiendo que solo usuarios con el rol de **Colaborador** (collaborator) puedan modificar los registros.

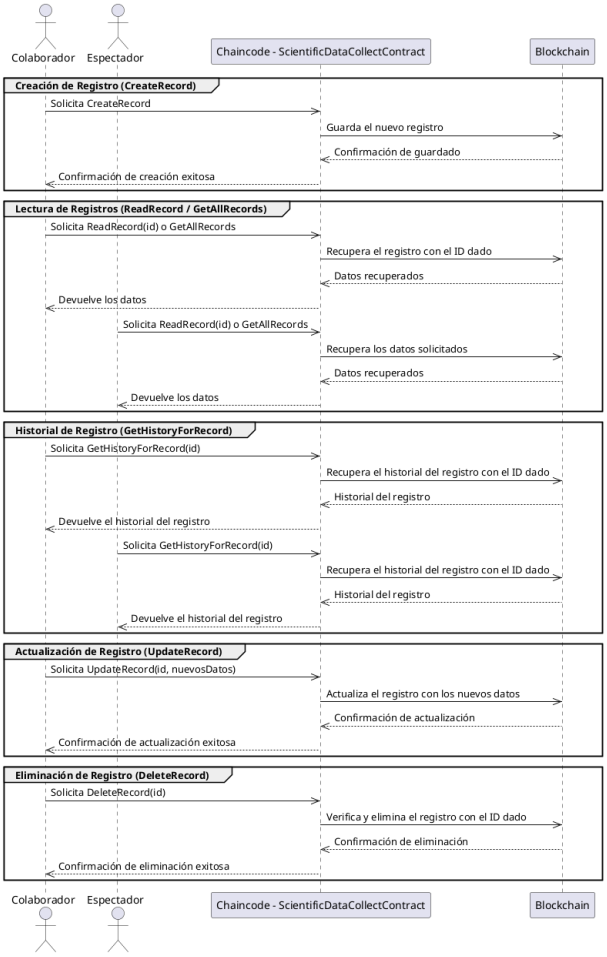
[11] Script para la creación y unión de los nodos del canal

https://github.com/GSMier/ProyectoLago/blob/main/scripts/create-channel.sh

El chaincode ScientificDataCollectContract[12] está diseñado para gestionar registros de datos científicos en una red Hyperledger Fabric, asegurando su almacenamiento, consulta, modificación y eliminación bajo condiciones de control de acceso. Consta de las siguientes transacciones y operaciones:

Figura 9

Diagrama de secuencia del chaincode ScientificDataCollection



[12] El chaincode para los datos cientificos:

https://github.com/GSMier/ProyectoLago/blob/main/chaincode/chaincode-typescript/src/scientificDataCollect.ts

Creación de registros. La transacción CreateRecord permite almacenar un nuevo registro en la blockchain a los usuarios con rol **Colaborador.** Se validan los datos de entrada para asegurar que contengan los siguientes campos:

- Id
- Tipo de dato científico
- Fecha de generación
- Metadatos y datos crudos
- Nombre del sitio de origen

Lectura e historial de registros. El chaincode permite la recuperación de información con métodos de lectura:

- ReadRecord(id): Obtiene un registro individual por su ID.
- GetAllRecords(): Recupera todos los registros almacenados en la blockchain.
- GetHistoryForRecord(id): Obtiene el historial de modificaciones de un registro, permitiendo rastrear cambios a lo largo del tiempo.

Actualización de registros. La transacción UpdateRecord permite modificar un registro existente a los usuarios con rol **Colaborador.**

Eliminación de registros. La transacción DeleteRecord permite a los usuarios con rol **Colaborador** eliminar un registro si éste existe.

Una vez desarrollado el chaincode, se procede a su instalación y despliegue en la red donde se ejecuta todo el ciclo de vida del chaincode[13]. Primero, se empaca (package) el chaincode y se instala (install) en cada nodo par. Luego, cada organización debe aprobar (approve) el chaincode según la política de consenso establecida que, en este caso, cualquier nodo par(peer) puede aprobar.

[13] Script para ejecutar el ciclo de vida del chaincode ScientificDataCollectContract: https://github.com/GSMier/ProyectoLago/blob/main/scripts/deploy-chaincode.sh

Después de la aprobación, el chaincode comienza a ejecutarse (commit) en la red blockchain como contenedores. A partir de este momento, las transacciones pueden ser procesadas y registradas en la blockchain siguiendo la lógica definida en el chaincode.

Aplicación gateway. Para permitir la interacción de los usuarios con la blockchain, se desarrolla una aplicación gateway que actuará como intermediaria entre los clientes y la red Fabric. Esta aplicación se conecta con los nodos pares, permitiendo enviar transacciones y realizar consultas sobre los datos almacenados en la blockchain.

En este caso se desarrolla una aplicación en Node con TypeScript usando el SDK fabric-gateway. El proceso de conexión a un nodo par en Hyperledger Fabric mediante el Fabric Gateway SDK en Node.js y TypeScript comienza con la configuración de parámetros como el canal, el chaincode, el host y las rutas de los certificados criptográficos MSP y TLS generados por las autoridades de certificación (CA). Luego, se establece una conexión gRPC segura cargando el certificado TLS del nodo par para cifrar la comunicación. A continuación, se autentica la identidad del usuario leyendo su certificado X.509 y se genera un firmante (signer) usando su clave privada para firmar transacciones. Con estos elementos, se crea una conexión con el Fabric Gateway, donde se pueden invocar transacciones del chaincode ‘ScientificDataCollectContract‘ u otros chaincodes[14].

[14] Funcion para la conexión por medio del fabric-gateway al nodo par (peer)

https://github.com/GSMier/ProyectoLago/blob/main/app-gateway/application-gateway-typescript/src/connect.ts

Antes de ejecutar cualquier otro chaincode, se verifica con el chaincode qscc, un contrato especial que permite consultar información del ledger (blockchain), como bloques y transacciones. Se utiliza para detectar modificaciones en los datos almacenados en la blockchain[15].

Algorithm 2 Verificar la integridad del bloque

```
1: procedure INTEGRITYVERIFICATION(contentHash, dataHashString, block, i)
2:     if contentHash ≠ dataHashString then
3:         blockDataList ←Obtener la lista de transacciones del bloque.
4:         if blockDataList is not empty then
5:             for each bl in blockDataList do
6:                 if bl is not empty then
7:                     signature ← Extraer la firma de la transacción.
8:                     envPayload ← Obtener el payload de la transacción.
9:                     headerPayload ← Extraer el encabezado de la transacción.
10:                    txId ←Obtener el identificador de la transacción.
11:                    creatorBytes ←Extraer la identidad del creador de la transacción.
12:                    verifier ← Preparar la verificación de la firma con el algoritmo SHA-256.
13:                    Validar la firma del creador con los datos de la transacción.
14:                    isValid ← Se firma envPayload con creatorBytes y se compara con signature
15:                    if isValid = false then
16:                        Lanzar error indicando que el bloque i ha sido manipulado en la transacción con identificador txId en el bloque i.
17:                    end if
18:                else
19:                    Imprimir No hay datos de transacción en el bloque
20:                end if
21:            end for
22:        end if
23:    end if
24: end procedure
```

Para finalizar se verifica la integridad de un bloque en la blockchain comparando los datos almacenados con los calculados en tiempo real. Primero, se compara el hash del contenido

[15] Función para la verificación de la integridad de los datos dentro del legder: https://github.com/GSMier/ProyectoLago/blob/main/app-gateway/application-gateway-typescript/src/commands/qscc.ts

(contentHash) con el hash registrado en el encabezado del bloque (dataHashString). Si no coinciden, se considera que el bloque podría haber sido manipulado. Luego, se extraen y verifican las transacciones dentro del bloque.

Para cada transacción en el bloque, si no está vacía, se extraen la firma, el payload y el encabezado, junto con el identificador (txId) y la identidad del creador. Se prepara un verificador de firma usando SHA-256 y se valida si la firma del creador es auténtica. Si la validación falla, se lanza un error indicando que la transacción con el txId en el bloque i ha sido manipulada. Si no hay datos de transacción, se imprime un mensaje indicando su ausencia.

Adicionalmente, para garantizar que los datos almacenados en la blockchain no han sido modificados con respecto a los archivos originales, la aplicación tiene una función[16] para revisar los registros del ledger y compararlos con los hashes de los archivos fuente.

[16] Funcion para comparar hash de los datos fuente con los registrados en el legder https://github.com/GSMier/ProyectoLago/blob/main/app-gateway/application-gateway-typescript/src/commands/verify.ts

Algorithm 3 Verificar la integridad de los registros científicos

```
1: procedure VERIFYSCIENTIFICRECORDS(folderPath, network, channelName, chaincodeName)
2:     network ← Obtener la red de la blockchain.
3:     contract ← Obtener el contrato de los datos científicos
4:     records ← Obtener todos los registros con contract
5:     for each record in records do
6:         isRawDataVerified ← Verificar el hash de record.rawData con los datos crudos.
7:         isMetadataVerified ← Verificar el hash de record.metadata con los metadatos
8:         isInputDataVerified ← Verificar el hash de record.inputData con los datos de entrada.
9:         isInputMetadataVerified ← Verificar el hash de record.inputMetadata con los metadatos de entrada.
10:        isOutputDataVerified ← Verificar el hash de record.outputData con los datos de salida.
11:        isOutputMetadataVerified ← Verificar el hash de record.outputMetadata con los metadatos de salida
12:        if isRawDataVerified = true and isMetadataVerified = true and isInputDataVerified = true and isInputMetadataVerified = true
    and isOutputDataVerified = true and isOutputMetadataVerified = true then
13:            Imprimir Registro record.Id verificado con éxito.
14:        else
15:            Imprimir Error en la verificación del registro record.Id.
16:        end if
17:    end for
18: end procedure
```

Esta función verifica la integridad de los registros científicos almacenados en la blockchain comparando sus hashes con los datos originales. Primero, obtiene la red de la blockchain y accede al contrato que gestiona los datos científicos. Luego, recupera todos los registros almacenados en la red.

Para cada registro, se calculan y comparan los hashes de sus componentes clave: datos crudos, metadatos, datos y metadatos de entrada, y datos y metadatos de salida. Si todos los hashes coinciden con los valores almacenados en la blockchain, se imprime un mensaje confirmando la verificación exitosa del registro. Si alguno de los valores no coincide, se reporta un error indicando que el registro ha sido modificado o no es íntegro.

Y para mejorar la auditabilidad sobre posibles vulnerabilidades, la aplicación también puede recuperar una transacción específica del ledger y visualizar su contenido, lo que permite

verificar los parámetros de la transacción en múltiples nodos pares[17], con ayuda del contrato se extraen los datos crudos de la transacción y con la librería fabric-protos se deserializa.

Algorithm 4 Obtener y decodificar una transacción por el id de la transacción

```
1: procedure GETTRANSACTIONBYTXID(channelName, transactionId)
2:     network ← Obtener la red de la blockchain.
3:     contract ← Obtener el contrato de sistema.
4:     transaction ← Obtener la transacción con contract
5:     decodedData Decodificar los datos de la transacción.
6:     Almacenar decodedData en un archivo JSON
7: end procedure
```

Esta función permite obtener y decodificar una transacción específica en la blockchain utilizando su identificador único (transactionId). Primero, obtiene la red de la blockchain y accede al contrato del sistema que permite consultar transacciones. Luego, valida el ID de transacción proporcionado para asegurarse de que es válido.

Una vez validado, se recuperan y decodifican los datos de la transacción para obtener su estructura interna y detalles. Finalmente, los datos decodificados se almacenan en un archivo JSON, permitiendo su análisis o auditoría posterior.

Con el motivo de facilitar la ejecución de la aplicación, se contenedoriza y se exporta al repositorio Dockerhub[18].

[17] Función para extraer y deserializar transacción por su id https://github.com/GSMier/ProyectoLago/blob/main/app-gateway/application-gateway-typescript/src/commands/getByTxId.ts

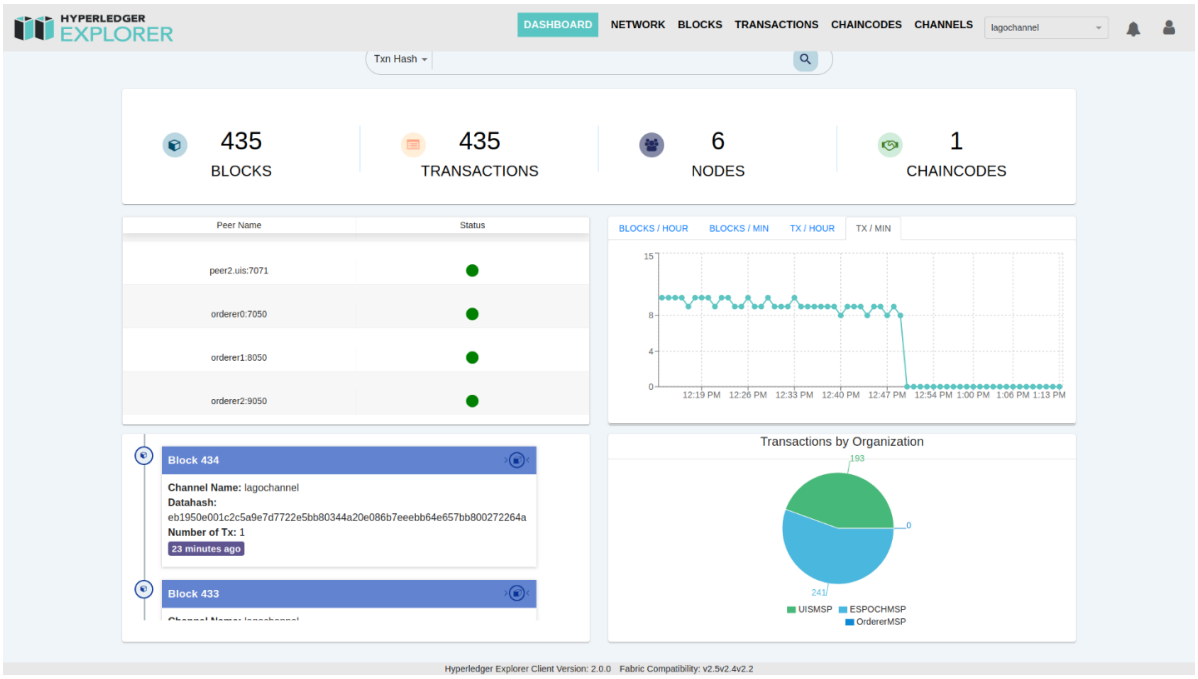
[18] Repositorio Docker hub de la aplicación gateway https://hub.docker.com/repository/docker/sebstiian/lagochain-app/general

4.5.2 Almacenamiento de datos de prueba (mediciones y simulaciones)

El proceso de almacenamiento de datos en la blockchain se realiza mediante un script en Bash[19] que automatiza la carga de archivos JSON generados por mediciones y simulaciones que se obtuvieron en la Curación y tratamiento de los datos de mediciones y simulaciones. Con las credenciales y parámetros de conexión configurados, se ejecuta el comando para guardar registros de la aplicación gateway, el cual llama al chaincode; este itera los archivos JSON, que contienen los datos de simulaciones y mediciones del proyecto LAGO.

Figura 10

Panel Hyperledger Explorer

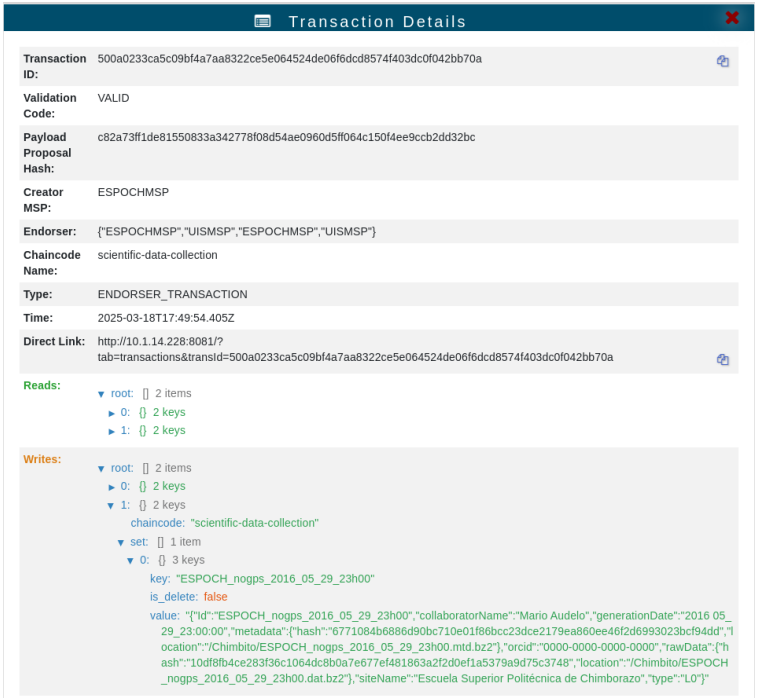


[19] Script bash para guardar los datos en la red blockchain: https://github.com/GSMier/ProyectoLago/blob/main/scripts/collect-data.sh

Con la ayuda de Hyperledger Explorer, un servicio web desarrollado por Hyperledger Fabric[20], se pueden visualizar datos de la red como: bloques, nodos, transacciones, chaincode y canales. Como se puede apreciar en la imagen 10, se efectuaron 434 transacciones, de las cuales 430 fueron invocadas por el chaincode "ScientificDataCollection" para almacenar los datos, que fueron distribuidos en bloques diferentes, dado que cada bloque acepta transacciones en intervalos de 2 segundos. Con un promedio de aproximadamente 8 transacciones/min, se registraron los datos científicos de las 2 organizaciones.

Figura 11

Detalles de una transacción en Hyperledger Explorer



[20] Repositorio de Hyperledger Explorer:

https://github.com/hyperledger-labs/blockchain-explorer

En todas las transacciones, se presentan los siguientes detalles: el ID de la transacción, si es válida, el hash de la propuesta del payload, el ID de la organización creadora, los que firmaron la propuesta y parámetros de creación.

4.6 Validación y ajuste

Se plantearon las pruebas de trazabilidad, integridad y control de acceso en la red blockchain. Se realizaron pruebas simulando manipulaciones en el ledger y en los datos externos, verificando la detección de alteraciones mediante herramientas de integridad y la aplicación **gateway.** Se evaluaron los mecanismos de autenticación y permisos, asegurando que solo usuarios autorizados pudieran operar en la red y bloqueando accesos no permitidos.

Finalmente, se realizaron ajustes para corregir inconsistencias detectadas en los nodos pares, implementando un mecanismo de reinicio para alinear su estado con la red y evitar corrupción de datos. Con estas validaciones y optimizaciones, se garantizó el correcto funcionamiento y seguridad del sistema.

4.6.1 Pruebas de trazabilidad e integridad de los datos recopilados.

Para validar la trazabilidad e integridad de los datos en la red, se plantean dos escenarios en los que se simula la manipulación de la información.

Manipulación del ledger. La blockchain es el componente donde se almacenan todas las transacciones registradas en la red. En **Hyperledger Fabric**, su estructura es más compleja, ya que cada transacción incluye firmas digitales generadas con las claves privadas de sus creadores. Además, los nodos pares que validan y aprueban estas transacciones también añaden sus firmas a los bloques.

En **Hyperledger Fabric**, la integridad y trazabilidad de los datos en la blockchain se validan mediante verificaciones en los bloques y transacciones. A nivel de bloques, se asegura la continuidad de la cadena comprobando que el hash del bloque anterior coincida con el almacenado

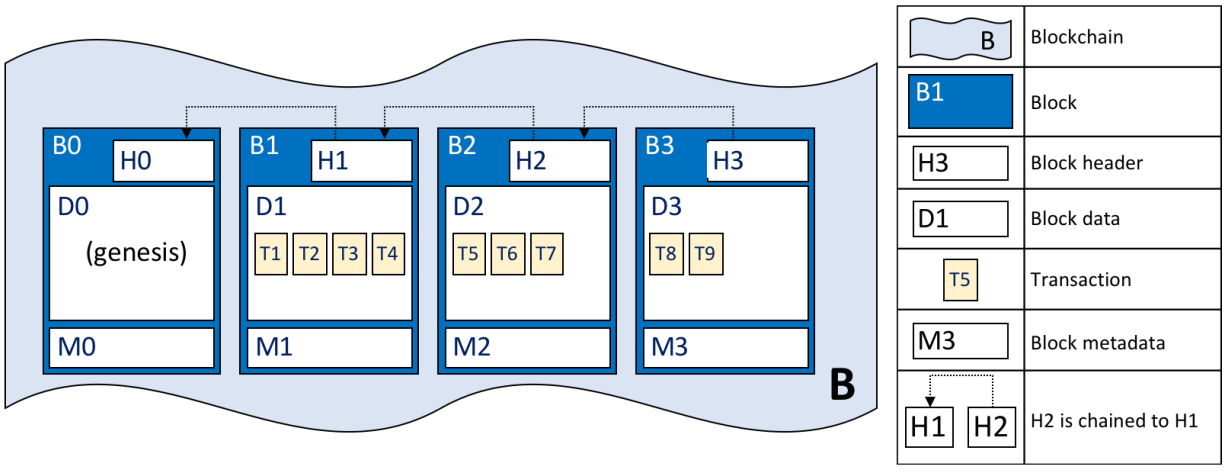
en el bloque actual, evitando alteraciones en el historial. También se verifica que el hash del bloque actual corresponda a su contenido, garantizando que no haya sido modificado.

Además, se validan las firmas de metadatos y configuraciones para confirmar que los bloques han sido emitidos por entidades autorizadas y que la configuración de la red es coherente y legítima.

En cuanto a las transacciones, se realizan verificaciones sobre las firmas digitales en los encabezados y aprobaciones, asegurando que las identidades involucradas pertenezcan a entidades autorizadas dentro del Membership Service Provider (MSP). Se valida que las firmas de los aprobadores sean legítimas y correspondan con el contenido aprobado en la transacción, garantizando que el consenso se haya seguido correctamente.

Figura 12

Estrutura blockchain de ledger – Hyperledger Fabric Docs main documentation. (2025)



En la estructura, cada cabecera contiene el número del bloque, el hash del bloque y el hash del bloque anterior, exceptuando el bloque génesis.

Para evaluar la integridad de los datos, se modifican directamente los registros de mediciones y simulaciones dentro del archivo que almacena la blockchain y se verifica cualquier alteración utilizando la herramienta **Blockchain Verifie**r[21], desarrollada por Hyperledger Fabric.

Manipulación del dato fuera del ledger. Los archivos que contienen las mediciones y simulaciones están vinculados a los registros almacenados en la blockchain. Para garantizar su integridad, se verifica que estos archivos no hayan sido alterados utilizando la aplicación **Gateway.** La funcionalidad descrita en el algoritmo 3 permite comprobar que la información externa sigue correspondiendo con los registros originales en la blockchain.

Además de la verificación de la integridad, la aplicación **Gateway** también se empleará para analizar la trazabilidad de los datos, asegurando que se pueda reconstruir el historial de cada registro, desde su creación hasta sus modificaciones y accesos a lo largo del tiempo.

Validacion de los mecanismos de acceso controlado

4.6.2 Validacion de los mecanismos de acceso controlado.

La validación del mecanismo de acceso controlado se centra en la verificación de identidad, autenticación y control de permisos. Para ello, se realizan pruebas con credenciales válidas, inválidas y revocadas, garantizando que solo los usuarios autenticados puedan acceder y operar en la red. Asimismo, se evalúa el correcto funcionamiento del **Membership Service Provider (MSP)**, asegurando que los usuarios registrados posean credenciales adecuadas y que los certificados revocados sean efectivamente rechazados.

[21] Repositorio de Blockchain Verifier: https://github.com/hyperledger-labs/blockchain-verifier

En cuanto a los permisos y roles, se verifica el acceso de distintos tipos de usuarios **(Colaboradores y Espectadores)** para garantizar que solo aquellos con los privilegios adecuados puedan realizar operaciones como la creación, consulta, modificación o eliminación de registros en el ledger. También se simulan intentos de acceso con credenciales revocadas y se comprueba que sean bloqueados correctamente, garantizando que el sistema mantenga un control de acceso robusto y seguro.

4.6.3 Ajustes finales de la implementación

Para realizar los ajustes finales en la implementación, se desarrolló un mecanismo que permite resetear el estado de la blockchain en un nodo par y alinearlo con el estado de los demás nodos mediante el uso de un script[22]. Esta solución se implementó tras detectar que, durante las pruebas, el nodo par quedaba en un estado corrupto, impidiendo su correcto funcionamiento.

Adicionalmente, se identificó que ciertas modificaciones realizadas en el ledger no eran reconocidas por la red blockchain y, en consecuencia, eran ignoradas, lo que generaba inconsistencias en la validación de transacciones. Para mitigar este problema, a nivel de aplicación se decidió bloquear las transacciones cuando el nodo par consultado se encuentra en estado corrupto, como se muestra en el procedimiento para la verificación de la integridad de los bloques 2. Sin embargo, este comportamiento es inesperado y no debería ocurrir en un sistema con garantías de integridad.

Figura 13

Transacción bloqueada debido a manipulación de la blockchain

```
> app-gateway-lago@1.0.0 start
> node dist/app.js update /var/data/UPDATE-1.json

channelName:       lagochannel
chaincodeName:     scientific-data-collection
mspId:             UISMSP
cryptoPath:        /test-network/organizations/peerOrganizations/org1.example.com
keyDirectoryPath:  /etc/data/uis/users/prueba-user1/msp/keystore/
certDirectoryPath: /etc/data/uis/users/prueba-user1/msp/signcerts/
tlsCertPath:       /etc/data/uis/tlsca/tlsca.uis-cert.pem
peerEndpoint:      peer0.uis:7051
peerHostAlias:     peer0.uis
*** Successfully submitted transaction of record update with  ID: S0_bga_10_77402_QGSII_flat_defaults_DAT190014}
npm notice
```

[22] Script para resetear un nodo: https://github.com/GSMier/ProyectoLago/blob/main/scripts/repair-ledger.sh

5. Resultados y discusión

5.1 Pruebas de manipulación del ledger.

Para las pruebas de validación de integridad, se manipuló directamente el archivo de la blockchain, lo que representa un escenario extremo en el que se puede demostrar la solidez del mecanismo de integridad de los datos. Sin embargo, este tipo de alteración es poco probable en un entorno real. Para este experimento, se seleccionaron trece datos de manera aleatoria entre todas las transacciones.

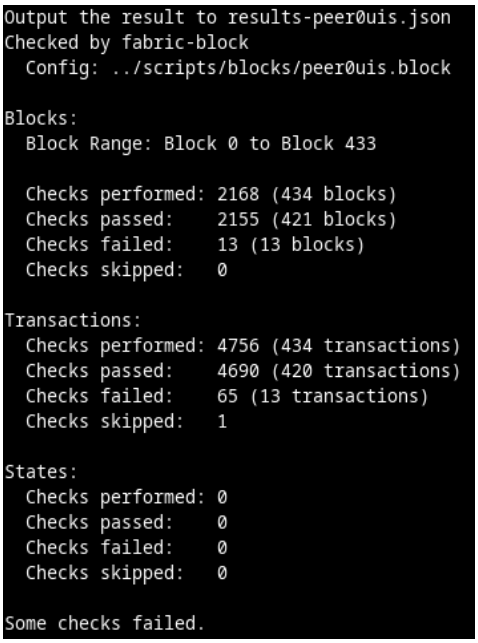
A continuación, se presentan los resultados obtenidos con la herramienta **Blockchain Verifier**, que a través de muestras de la blockchain de los nodos se hicieron las validaciones utilizando el script[23], arrojando los siguientes resultados[24] en formato JSON de las pruebas sobre bloques y transacciones.

[23] Link del script para validar la blockchain de los nodos: https://github.com/GSMier/ProyectoLago/blob/main/scripts/verify-integrity.sh

[24] Resultados de **Blockchain Verifier** https://github.com/GSMier/ProyectoLago/tree/main/resultados

Figura 14

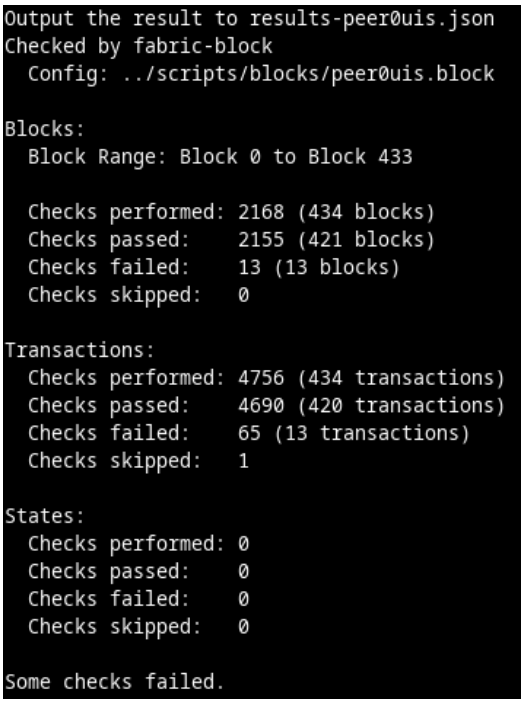
Resumen de validación sobre el bloque del nodo peer0 organización UIS



Como se muestra en la imagen 15, se realizó la validación sobre los 434 bloques generados a partir de la recolección de datos científicos, de los cuales 421 pasaron exitosamente la verificación, mientras que 13 fueron detectados como manipulados.

Figura 15

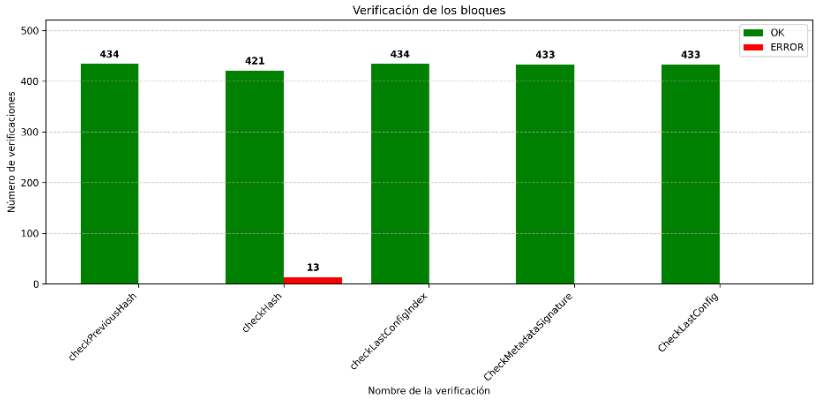
Resumen de validación sobre el bloque del nodo peer1 organización UIS



En contraste, al ejecutar la verificación sobre un nodo par que no ha sido manipulado, se observa que todas las pruebas realizadas por **Blockchain Verifier** son exitosas, lo que confirma la integridad de los datos almacenados en la blockchain.

Figura 16

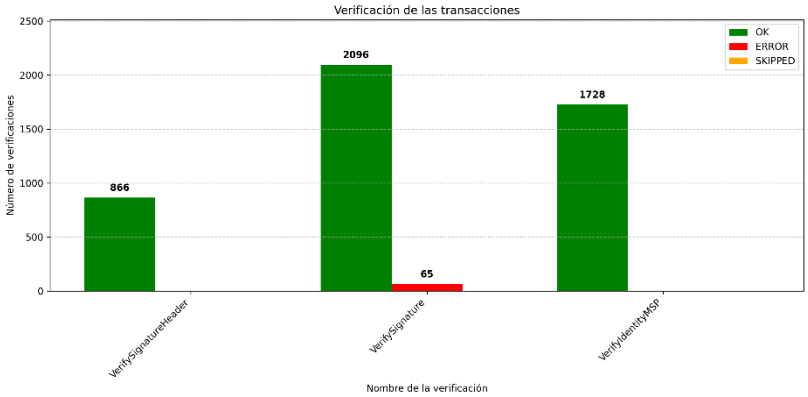
Grafica de verificación de los bloques



Al analizar en detalle las verificaciones realizadas por **Blockchain Verifier,** se evidencia que cuando se manipulan los datos, el hash del contenido del bloque en el momento de su creación deja de coincidir con el hash calculado a partir de los datos actuales. Del mismo modo, los valores generados durante la creación del bloque no se recalculan, como es el caso del hash del bloque almacenado en su cabecera, lo que impide la detección de la alteración.

Figura 17

Grafica de verificación de las transacciones



Adicionalmente, en **Hyperledger Fabric,** cada transacción es firmada con la clave privada del **Membership Service Provider (MSP)**. Las cargas útiles **(payloads)** son firmadas por las identidades MSP, generando firmas digitales **(signatures),** las cuales dependen directamente de los datos contenidos en la carga útil.

En este caso, 65 firmas no pasaron la validación. Esto se debe a que cada transacción puede contener una o más firmas, dependiendo de la cantidad de firmas aprobatorias necesarias para su validación. En el caso del chaincode **ScientificDataCollection,** se requieren cuatro firmas para aprobar una transacción. Es importante considerar que existen otros chaincodes en ejecución, como el correspondiente al ciclo de vida para la implementación de un nuevo **chaincode.**

5.2 Pruebas de manipulación del dato fuera del ledger.

Para evaluar la manipulación de archivos externos a la blockchain, se seleccionaron los archivos asociados a un dato almacenado en la blockchain. A continuación, mediante la funcionalidad 3, se verificó la integridad de estos archivos.

Figura 18

Verificación del registro S0_bga_10_77402_QGSII_flat-defaults_DAT030402

```
> app-gateway-lago@1.0.0 start
> node dist/app.js verifyOne /var/data S0_bga_10_77402_QGSII_flat_defaults_DAT030402

channelName:       lagochannel
chaincodeName:     scientific-data-collection
mspId:             UISMSP
cryptoPath:        /test-network/organizations/peerOrganizations/org1.example.com
keyDirectoryPath:  /etc/data/uis/users/viewer-uis/msp/keystore/
certDirectoryPath: /etc/data/uis/users/viewer-uis/msp/signcerts/
tlsCertPath:       /etc/data/uis/tlsca/tlsca.uis-cert.pem
peerEndpoint:      peer0.uis:7051
peerHostAlias:     peer0.uis
Record S0_bga_10_77402_QGSII_flat_defaults_DAT030402 is verified.
```

Posteriormente, se modificaron algunos de los archivos[25] con el objetivo de detectar la manipulación.

[25] Datos de prueba manipulados: https://github.com/GSMier/ProyectoLago/tree/main/tampered-data

Figura 19

Verificación del registro S0_bga_10_77402_QGSII_flat_defaults_DAT030402

```
> app-gateway-lago@1.0.0 start
> node dist/app.js verifyOne /var/data S0_bga_10_77402_QGSII_flat_defaults_DAT030402

channelName:       lagochannel
chaincodeName:     scientific-data-collection
mspId:             UISMSP
cryptoPath:        /test-network/organizations/peerOrganizations/org1.example.com
keyDirectoryPath:  /etc/data/uis/users/viewer-uis/msp/keystore/
certDirectoryPath: /etc/data/uis/users/viewer-uis/msp/signcerts/
tlsCertPath:       /etc/data/uis/tlsca/tlsca.uis-cert.pem
peerEndpoint:      peer0.uis:7051
peerHostAlias:     peer0.uis
Record S0_bga_10_77402_QGSII_flat_defaults_DAT030402 verification failed.
```

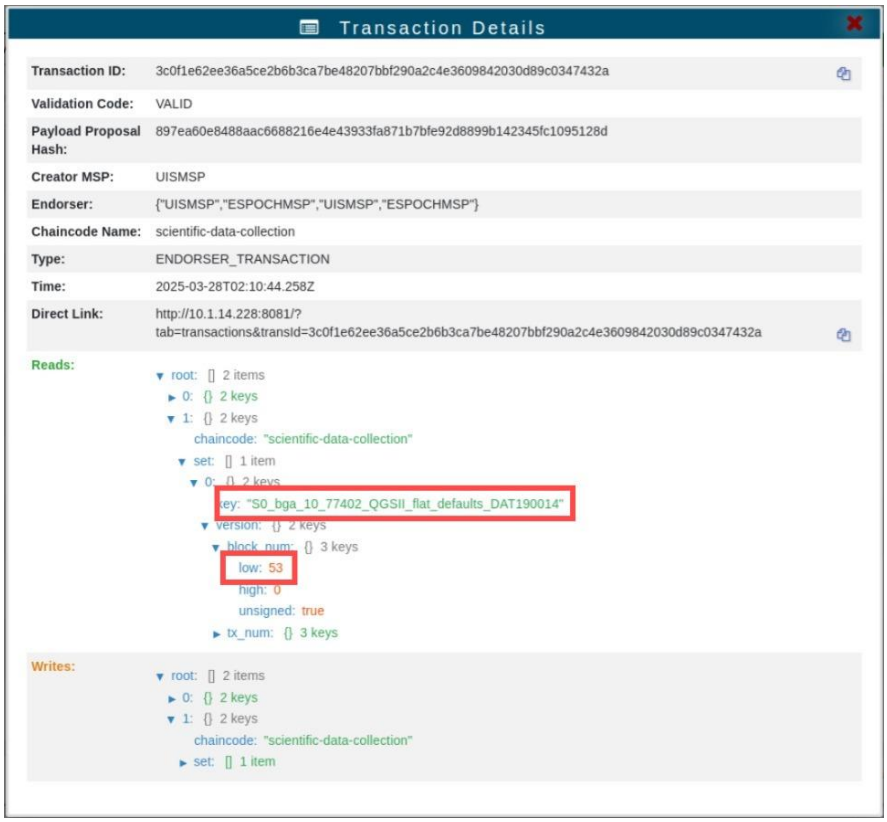
De esta manera, cuando ocurre una alteración en los archivos sin que exista una actualización correspondiente en el registro de la blockchain, la aplicación Gateway es capaz de detectar dicha manipulación.

5.3 Trazabilidad de un registro

A través de la aplicación **gateway**, es posible rastrear el historial de modificaciones de un registro. Para este experimento, se realizó una actualización del registro con el propósito de evaluar su trazabilidad. Utilizando el chaincode **ScientificDataCollect,** se recuperó el historial completo de transacciones asociadas al registro.

Figura 20

Transacción de actualización del registro S0_bga_10_77402QSII_flat_defaults_DAT190014



En la figura 20 se muestra la actualización del registro S0_bga_10_77402QSII_ f lat_de f aults_DAT190014. Al consultar el historial de un registro, se obtienen todas las transacciones que lo han modificado, lo que permite visualizar su estado en cada momento del tiempo. Por ejemplo,

en la transacción 1[26] se puede ver el estado previo del registro junto con su marca temporal, mientras que en la transacción 2[27] se observa su estado más reciente. Esto permite trazar claramente la evolución de cada registro a lo largo del tiempo.

5.4 Pruebas de acceso controlado

Para evaluar el mecanismo de acceso controlado, se crearon dos nuevos usuarios utilizando la autoridad de certificados de la organización UIS. Se generaron sus certificados y se configuró el **MSP**, asignando a uno el rol de **Colaborador** y al otro el rol de **Espectador** mediante el siguiente script[28].

Cuando el usuario con rol de **Espectador** intenta modificar un registro en la blockchain, el chaincode genera un error porque valida que en los certificados esté presente el atributo colaborador, por lo que se bloquea cualquier transacción realizada.

. Por otro lado, al utilizar las credenciales del usuario con rol de **Colaborador**, la transacción se envía exitosamente.

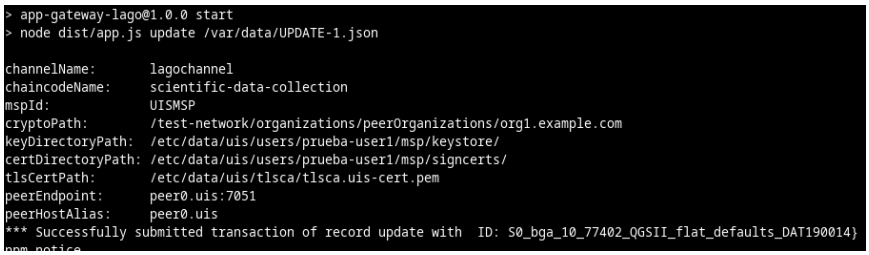
[26] Imagen de la transacción 1: https://github.com/GSMier/ProyectoLago/blob/main/tx1.png

[27] Imagen de la transacción 2: https://github.com/GSMier/ProyectoLago/blob/main/tx2.png

[28] Link del script para crear los usuarios: https://github.com/GSMier/ProyectoLago/blob/main/scripts/add-user.sh

Figura 21

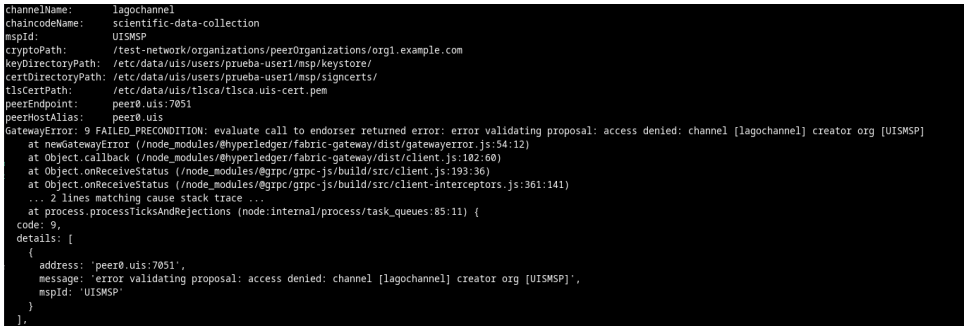
Mensaje de confirmación de transacción exitosa con el rol de colaborador



Posteriormente, las credenciales del usuario **Colaborador** fueron revocadas mediante la actualización de la **Certificate Revocation List (CRL)** utilizando el script[29]. De este modo, al intentar efectuar una transacción, el servicio MSP deniega la conexión.

Figura 22

Mensaje de error al tratar de conectarse con un certificado revocado



Como última prueba, se generaron certificados utilizando autoridades de certificación no válidas con el siguiente script[30]. Esto se debe a que solo se consideran válidas aquellas con las

[29] Link del script para revocar certificados:

https://github.com/GSMier/ProyectoLago/blob/main/scripts/revoke-credentials.sh

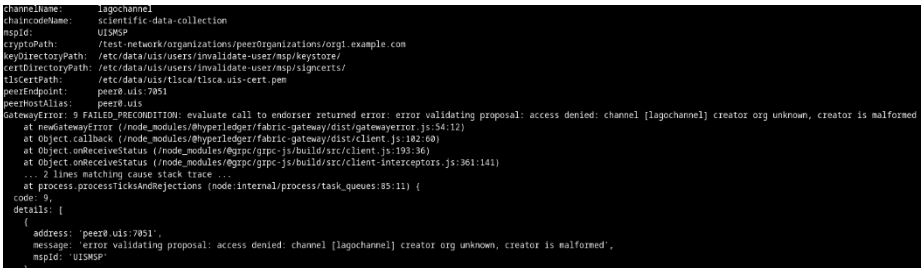
[30] Link del script para crear credenciales invalidas:

https://github.com/GSMier/ProyectoLago/blob/main/scripts/generate-invalid-credentials.sh

cuales se configuró el servicio **MSP** de cada organización en el momento de la creación de la red. Por lo tanto, estos certificados no cuentan con un emisor autorizado.

Figura 23

Mensaje de error al tratar de conectarse con un certificado inválido



6. Conclusiones

El desarrollo del **prototipo blockchain para introducir integridad y acceso controlado en la recopilación de mediciones y simulaciones del Proyecto LAGO** ha permitido evaluar la viabilidad de esta tecnología en entornos de colaboración científica. En este contexto, la ciencia colaborativa enfrenta desafíos relacionados con la seguridad, la transparencia y la preservación de datos, lo que ha impulsado el interés en soluciones basadas en **blockchain** para fortalecer la integridad y el control de acceso. Por lo que se puede concluir a partir del prototipo:

Integridad garantizada. La implementación de **Hyperledger Fabric** demostró ser una solución eficaz para preservar la integridad de los datos científicos en el **Proyecto LAGO**, no solo por su bajo costo computacional, sino por su capacidad de adaptarse al contexto de los datos científicos, facilitando la creación de registros inmutables y descentralizados, asegurando que los datos sean verificables. Se verificó que, a la más mínima manipulación en los registros almacenados dentro de la blockchain, es detectada de inmediato gracias a los mecanismos criptográficos inherentes a la blockchain.

Durante el desarrollo e implementación, se identificaron algunos desafíos en la gestión de los nodos pares, específicamente en la detección de modificaciones no sincronizadas. Para solventar esto, se implementó un mecanismo de reinicialización de nodos que garantiza la consistencia del sistema.

Esto resalta la necesidad de continuar con mejoras en la escalabilidad y mantenimiento del sistema para garantizar una operatividad continua y sin interrupciones en un entorno de producción.

Trazabilidad de datos para su reutilización. Uno de los principales objetivos del **Proyecto LAGO** es garantizar la preservación y reutilización de datos científicos, desde mediciones en bruto hasta simulaciones avanzadas. Durante el desarrollo del prototipo blockchain,

se observó que la integración de **Hyperledger Fabric** permite registrar cada transformación que experimentan los datos en las distintas capas del flujo de información (L0 a L3 y S0 a S2), asegurando una trazabilidad completa.

Este enfoque facilita la validación y comparación de datos a lo largo del tiempo, lo que es clave en experimentos de larga duración. Además, la posibilidad de asociar metadatos estructurados y verificables a cada registro mejora la interoperabilidad y accesibilidad de la información, alineándose con los principios **FAIR** (Findable, Accessible, Interoperable, and Reusable).

Otro hallazgo relevante fue la capacidad de la blockchain para respaldar procesos de reprocesamiento y validación de datos. El almacenamiento de versiones verificables de cada conjunto de datos en un registro inmutable permite que los investigadores puedan revisar y actualizar sus análisis sin riesgo de pérdida o manipulación indebida. Esto es especialmente útil en escenarios donde se requiere ajustar modelos o corregir errores en simulaciones.

Además, el análisis del flujo de datos del **Proyecto LAG**O evidenció que actualmente solo algunas capas de información están disponibles en repositorios distribuidos, mientras que otras siguen almacenadas en servidores locales. La implementación de blockchain ayudaría a mitigar esta fragmentación, proporcionando un marco confiable para la integración progresiva de datos en un entorno descentralizado, con acceso controlado y registro inmutable de modificaciones.

Acceso controlado en los datos científicos. El uso de **Hyperledger Fabric** como infraestructura para la gestión del acceso controlado ha demostrado ser altamente efectivo en entornos de ciencia colaborativa. A través del **Membership Service Provider (MSP)**, se logró una administración rigurosa de autenticación y autorización, asegurando que cada usuario acceda únicamente a los recursos permitidos según su rol. La correcta implementación de permisos

impidió modificaciones no autorizadas en la blockchain, fortaleciendo el acceso controlado de los datos científicos.

El desarrollo del **prototipo blockchain** en el **Proyecto LAGO** ha demostrado su viabilidad para garantizar **integridad, trazabilidad y acceso controlado** en la recopilación de mediciones y simulaciones del **Proyecto LAGO.** La implementación de Hyperledger Fabric permitió registrar y validar cada transformación de datos, alineándose con los principios **FAIR** y asegurando que solo usuarios autorizados accedan a la información.

7. Trabajo futuro

Si bien los resultados obtenidos son prometedores, aún existen desafíos en términos de **escalabilidad** y **usabilidad** del sistema en un entorno de producción. En futuras etapas, el trabajo se enfocará en:

- Optimizar la escalabilidad del sistema, permitiendo el manejo eficiente de grandes volúmenes de datos generados por el Proyecto LAGO sin comprometer el rendimiento de la red.
- Mejorar la interoperabilidad con plataformas de almacenamiento distribuido, asegurando que todas las capas de datos puedan integrarse de manera transparente en la blockchain.
- Desarrollar herramientas de gestión y visualización accesibles para la comunidad científica, facilitando la adopción del sistema y promoviendo su uso dentro de la colaboración LAGO.
- Evaluar el desempeño en un entorno real, analizando su aplicación en escenarios de colaboración internacional para garantizar su viabilidad y sostenibilidad a largo plazo.

Con estas mejoras, se espera consolidar un sistema escalable y funcional, que no solo fortalezca la **integridad y acceso controlado** en el manejo de datos científicos, sino que también fomente una colaboración más abierta, confiable y eficiente dentro del ecosistema del **Proyecto LAGO.**

Aunque para este tipo de casos la tecnología Blockchain resulta ser de mucha utilidad, en el mundo real tiene muy pocas aplicaciones, lo que desincentiva a seguir investigando en esta tecnología. Sin embargo, apoyarse en otras tecnologías demandadas como **Inteligencia Artificial**, podría llegar a beneficiar ambas en sus puntos débiles, en temas de trazabilidad, interoperabilidad y escalabilidad.

Referencias

Alniamy, A. M., & Liu, H. (2020). *Blockchain-based secure collaboration platform for sharing and accessing scientific research data*. In *2020 3rd International Conference on Hot Information-Centric Networking (HotICN)* (pp. 34–40). IEEE. https://doi.org/10.1109/HotICN50757.2020.9355117

Coelho, R., Braga, R., David, J. M. N., Stroele, V., Campos, F., & Dantas, M. (2022). A Blockchain-Based Architecture for Trust in Collaborative Scientific Experimentation. Journal of grid computing, 20(4), 35-35.

Collaboration, L. (2022). The LAGO Data Management Plan (DMP) document 1.1 [Accessed: 2025-1-10].

Collins, J. P. (2010). Sailing on an Ocean of 0s and 1s The Fourth Paradigm Data-Intensive Scientific Discovery Tony Hey, Stewart Tansley, and Kristin Tolle, Eds. Microsoft Research, Redmond, WA, 2009. 286 pp. Paper, 46.ISBN9780982544204.PDFathtt p : //research.microso ft.com/en−us/collaboration/ f ourthparadigm. Science, 327(5972), 1455-1456.

CompTIA. (2021). 7 phases of Blockchain Implementation: Comptia. https: //www.mspinsights.com/doc/phases-of-blockchain-implementation-0001

Donath, M. (2022). About - the Latin American giant observatory [Accessed: 2024-10-29].

Dutta, S. K. (2020 - 2020). The definitive guide to blockchain for accounting and business : understanding the revolutionary technology (1st ed.). Emerald Publishing Limited.

Fox, P., & Kozyra, J. (2015). eScience and Informatics for international science programs. Progress in earth and planetary science, 2(1), 1-.

Hirsh, S., & Alman, S. W. (2020). Blockchain. ALA Neal-Schuman.

Hoopes, R., Hardy, H., Long, M., & Dagher, G. G. (2022). SciLedger: A Blockchain-based Scientific Workflow Provenance and Data Sharing Platform. 2022 IEEE 8th International Conference on Collaboration and Internet Computing (CIC), 125-134.

Hyperledger. (2020a). Identity — Hyperledger Fabric Docs main documentation [Accessed: 2025-1-15].

Hyperledger. (2020b). Transaction Flow — Hyperledger Fabric Docs main documentation [Accessed: 2025-1-22].

ISO/IEC. (2022). ISO/IEC 27001:2022 - Information security, cybersecurity and privacy protection — Information security management systems — Requirements [ISO/IEC 27001:2022 specifies the requirements for establishing, implementing, maintaining, and continually improving an information security management system (ISMS).]. International Organization for Standardization.

Karastoyanova, D., & Stage, L. ("2018"). Towards Collaborative and Reproducible Scientific Experiments on Blockchain. Advanced Information Systems Engineering Workshops, 144-149.

Kaushal, R. K., Kumar, N., Panda, S. N., & Kukreja, V. (2021). Immutable Smart Contracts on Blockchain Technology: Its Benefits and Barriers. 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 1-5.

Kleppner, D. (2009). Ensuring the integrity, accessibility, and stewardship of research data in the digital age (1st ed.). National Academies Press.

LAGO. (2022). The LAGO Data Management Plan (DMP) document 1.1 [Accessed: 2024-11-9].

Li, J., & Kassem, M. (2021). Applications of distributed ledger technology (DLT) and Blockchain-enabled smart contracts in construction. Automation in construction, 132, 103955-.

Li, W., Meese, C., Nejad, M., & Guo, H. (2023). P-CFT: A Privacy-preserving and Crash Fault Tolerant Consensus Algorithm for Permissioned Blockchains. ArXiv.org.

Piyoungkorn, K., Chaisawat, S., & Vorakulpipat, C. (2022). Trusted Electronic Contract for Enabling Peer-to-Peer HPC Resource Sharing. Applied sciences, 12(10), 5153-.

Pooja, S., & Chandrakala, C. B. (2024). Secure Reviewing and Data Sharing in Scientific Collaboration: Leveraging Blockchain and Zero Trust Architecture. IEEE access, 12, 92386-92399.

Preve, N. P. (2011). Grid computing : towards a global interconnected infrastructure (1st ed. 2011.). Springer-Verlag.

Raj, P., & Deka, G. C. (2018). Blockchain technology : platforms, tools and use cases. Academic Press, an imprint of Elsevier.

Xia, Q., Sifah, E. B., Asamoah, K. O., Gao, J., Du, X., & Guizani, M. (2017). MeDShare: Trust-Less Medical Data Sharing Among Cloud Service Providers via Blockchain. IEEE Access, 5, 14757-14767. https://doi.org/10.1109/ACCESS.2017.2730843

Xiong, H., Chen, M., Wu, C., Zhao, Y., & Yi, W. (2022). Research on Progress of Blockchain Consensus Algorithm: A Review on Recent Progress of Blockchain Consensus Algorithms. Future internet, 14(2), 47-.

Xu, H., Zhang, L., Liu, Y., & Cao, B. (2020). RAFT Based Wireless Blockchain Networks in the Presence of Malicious Jamming. IEEE Wireless Communications Letters, 9(6), 817-821. https://doi.org/10.1109/LWC.2020.2971469

Yadav, A. S., Singh, N., & Kushwaha, D. S. (2023). Evolution of Blockchain and consensus mechanisms its real-world applications. Multimedia tools and applications, 82(22), 34363-34408.

Zhang, F., Wang, Z., Guo, R., & Qu, G. (2024). Earth Observation Data Provenance: A Blockchain-Based Solution. IEEE transactions on industrial informatics, 20(7), 9548-9556.