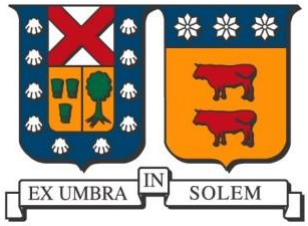
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



"ARQUITECTURA HÍBRIDA BLOCKCHAIN - BASE DE DATOS PARA TRAZABILIDAD EN COMPRAS PÚBLICAS CHILENAS"

IGOR ANDREE VEGA PUEBLA

TESINA PARA OPTAR AL GRADO
DE MAGÍSTER EN TECNOLOGÍAS DE
LA INFORMACIÓN

Profesor Guía: MAURICIO FIGUEROA COLARTE

CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☐ Memoria o trabajo de título ☒ Tesis de Postgrado

Título del trabajo: Arquitectura Híbrida Blockchain-Base de Datos para Trazabilidad en Compras Públicas Chilenas

Nombre del candidato(a): Igor Andree Vega Puebla

Carrera / Grado: Magíster en Tecnologías de la Información.

Campus: San Joaquín **Departamento:** Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, **Mauricio Figueroa**, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por **(marcar una opción):**

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

_____

_____

_____

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 30-10-2026 **Firma:** _____

Estudiante o Candidato(a):

Fecha: 30-10-2026 **Firma:** _____

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Arquitectura Híbrida Blockchain-Base de Datos para Trazabilidad en Compras Públicas Chilenas

Igor Andreé Vega Puebla

ivegap@usm.cl

Resumen: La trazabilidad de compras públicas exige verificar la integridad histórica de hitos críticos. Esta investigación diseñó, implementó y evaluó ChainGov, arquitectura híbrida blockchain-base de datos propia, concebida específicamente para este estudio, que mantuvo Microsoft SQL Server como sistema transaccional e incorporó una red Hyperledger Fabric permisionada como capa de verificabilidad, con hashes SHA-256, Transactional Outbox y Kafka para anclar evidencia de eventos seleccionados. El objetivo de negocio esperado fue incrementar la confianza en el proceso, dimensión no evaluada empíricamente. La validación comparó dos condiciones en el mismo escenario funcional: T0, línea base relacional sin esa capa, y T1, la misma plataforma operando con ChainGov, midiendo tiempo de auditoría, rendimiento transaccional y completitud verificable. En T1, 500 casos auditados concordaron con el *ground truth* y se verificaron 3500 de 3500 eventos. La comparación temporal T0/T1 no pudo determinarse bajo equivalencia funcional: T0 carecía de capacidad histórica independiente comparable. Bajo carga, la degradación máxima de *throughput* en T1 alcanzó 80,66%, superior al límite preespecificado de 15%, y la hipótesis no quedó respaldada. Los resultados mostraron la viabilidad técnica de incorporar verificabilidad distribuida, pero evidenciaron un costo de rendimiento y un rezago de anclaje que requieren optimización antes de un uso productivo.

Palabras Clave: • Applied computing → E-government; • Security and privacy → Distributed systems security; • Information systems → Relational database model; Information integration; • General and reference → Performance

1 Introducción

1.1 Contexto y motivación

La transformación digital de las compras públicas ha incrementado la disponibilidad de información y la capacidad de gestionar electrónicamente las distintas etapas de los procesos de contratación. En Chile, este ámbito posee una escala significativa: durante 2025, las entidades públicas transaron US$21.953 millones mediante la plataforma administrada por la Dirección de Compras y Contratación Pública, en aproximadamente 1,8 millones de órdenes de compra [1]. En este contexto, la trazabilidad de los hitos asociados a una contratación constituye un elemento relevante para la auditoría y la revisión posterior de los procesos.

Los sistemas de gestión transaccional se apoyan habitualmente en bases de datos relacionales, adecuadas para almacenar, consultar y actualizar información operacional. Sin embargo, la verificación retrospectiva de la integridad de los registros depende de los mecanismos de control, auditoría y privilegios administrados por la propia infraestructura. Esto dificulta que actores institucionales distintos puedan comprobar, mediante evidencia criptográfica independiente, que determinados hitos históricos permanecieron inalterados.

Las tecnologías de registro distribuido ofrecen mecanismos complementarios para abordar este problema. En particular, Hyperledger Fabric permite implementar redes blockchain permisionadas en las que las identidades, políticas de respaldo y mecanismos de acceso son administrados explícitamente [2]. No obstante, incorporar una blockchain a un sistema transaccional introduce costos adicionales de integración, procesamiento y coordinación. Por ello, una arquitectura híbrida debe evaluarse no solo por la verificabilidad que incorpora, sino también por su impacto sobre el rendimiento del sistema [3] [4].

1.2 Problema de investigación

El problema abordado en esta investigación correspondió a la limitada capacidad de una arquitectura basada únicamente en una base de datos relacional para proporcionar a actores institucionales una verificación criptográfica distribuida e independiente de la integridad histórica de hitos críticos de un proceso de compras públicas.

Esta limitación no implica que SQL Server sea inseguro ni que los administradores modifiquen indebidamente los registros. El problema se situó específicamente en el modelo de verificabilidad: cuando el estado operacional y los mecanismos para comprobar su historia dependen de la misma infraestructura administrativa, la evidencia disponible no posee las mismas propiedades que un registro distribuido entre organizaciones con identidades y políticas de validación diferenciadas.

A partir de este problema se formuló la siguiente pregunta de investigación:

¿En qué medida una arquitectura híbrida blockchain-base de datos basada en Hyperledger Fabric puede mejorar el tiempo de auditoría de trazabilidad de procesos de compras públicas respecto de una arquitectura basada únicamente en SQL Server, manteniendo una degradación del rendimiento transaccional dentro de un umbral aceptable?

1.3 Solución propuesta

Para abordar el problema se diseñó e implementó ChainGov, una arquitectura híbrida que mantuvo SQL Server como sistema de persistencia transaccional e incorporó Hyperledger Fabric como capa distribuida de verificabilidad. La propuesta no buscó sustituir la base de datos relacional, sino complementar sus capacidades mediante evidencia criptográfica asociada a hitos seleccionados.

ChainGov generó representaciones canónicas de los eventos críticos y calculó hashes SHA-256 que posteriormente fueron anclados en una red Hyperledger Fabric. La integración entre la persistencia operacional y el procesamiento posterior se implementó mediante el patrón Transactional Outbox y un flujo asíncrono que utilizó Apache Kafka. La arquitectura incluyó, además, mecanismos de idempotencia, separación de información según su nivel de privacidad y una interfaz de verificación que permitió contrastar el estado almacenado con la evidencia registrada.

Este enfoque se relacionó con propuestas de sistemas híbridos y bases de datos verificables que combinan propiedades de sistemas de gestión tradicionales con mecanismos criptográficos o registros distribuidos [4] [5]. En contratación pública, la literatura también ha estudiado blockchain como mecanismo para reforzar transparencia, trazabilidad y auditabilidad, aunque sus beneficios dependen del diseño institucional y técnico adoptado [6] [7].

1.4 Objetivos

El objetivo general fue diseñar, implementar y validar una arquitectura híbrida blockchain-base de datos, denominada ChainGov, que incorporara verificabilidad criptográfica a la trazabilidad de hitos críticos de

procesos de compras públicas, manteniendo los datos transaccionales en SQL Server y utilizando Hyperledger Fabric como capa distribuida de verificabilidad.

Para alcanzar este propósito se establecieron cuatro objetivos específicos:

1. Caracterizar las limitaciones y alternativas para la trazabilidad y verificabilidad de procesos de compras públicas soportados por sistemas relacionales.
2. Diseñar e implementar ChainGov mediante una partición explícita de responsabilidades entre SQL Server y Hyperledger Fabric, incorporando hashes SHA-256, Transactional Outbox y middleware de sincronización.
3. Evaluar experimentalmente la arquitectura frente a una línea base SQL mediante tiempo de auditoría, rendimiento transaccional y completitud verificable.
4. Analizar los resultados obtenidos respecto de la hipótesis planteada, identificando limitaciones y condiciones necesarias para la replicabilidad del estudio.

1.5 Hipótesis y estrategia de validación

La hipótesis de investigación H1 estableció que una arquitectura híbrida blockchain-base de datos (ChainGov) mejoraría en un 40% el tiempo de auditoría de trazabilidad respecto de la línea base en SQL Server, sin degradar el rendimiento transaccional (TPS) en más de un 15%. Se rechaza si cualquiera de las dos condiciones no se cumple simultáneamente.

La validación utilizó una única variable independiente correspondiente a la condición arquitectónica experimental, con dos niveles: T0, línea base basada en SQL Server, y T1, arquitectura ChainGov. Se evaluaron tres variables dependientes: VD1, tiempo de auditoría de trazabilidad; VD2, rendimiento transaccional bajo carga sostenida; y VD3, completitud verificable de los eventos procesados.

Para VD1 se preparó un conjunto experimental de 500 casos con un ground truth independiente del mecanismo auditor. Para VD2 se realizaron pruebas con 10, 50, 100, 250 y 500 usuarios concurrentes, utilizando tres repeticiones oficiales por configuración. El criterio C1 correspondió a una mejora del tiempo de auditoría igual o superior al 40%, mientras que C2 exigió que la degradación del rendimiento no superara el 15% en las cargas preespecificadas. VD3 se utilizó como medida complementaria de la capacidad de recuperar y verificar la evidencia histórica, sin incorporarle retrospectivamente un umbral de aceptación no definido en la hipótesis original.

La comparación temporal asociada a C1 presentó posteriormente una limitación de equivalencia funcional entre T0 y T1, que se desarrolla en los capítulos metodológico y de resultados. Esta limitación se mantuvo explícita para evitar sustituir la tarea de verificación histórica por una consulta relacional más simple y, con ello, preservar la validez de la comparación experimental.

1.6 Organización del informe

El resto del informe se organizó de la siguiente manera. El Capítulo 2 presenta el marco teórico y el estado del arte sobre trazabilidad, verificabilidad, blockchain permisionada, sistemas híbridos y aplicaciones en contratación pública. El Capítulo 3 describe el diseño e implementación de ChainGov, incluyendo la distribución de responsabilidades entre SQL Server y Hyperledger Fabric, el mecanismo de sincronización y las funciones de verificación. El Capítulo 4 expone el diseño experimental, las variables, instrumentos, procedimientos y criterios utilizados para la validación. El Capítulo 5 presenta y discute los resultados de auditoría, completitud verificable y rendimiento. Finalmente, el Capítulo 6 desarrolla las conclusiones, las limitaciones del estudio, sus contribuciones y las líneas de trabajo futuro.

2 Marco teórico y estado del arte

2.1 Trazabilidad, integridad y verificabilidad

La trazabilidad corresponde a la capacidad de reconstruir la secuencia de eventos y estados asociados a un proceso. En un sistema de información, esta propiedad depende de que los eventos relevantes puedan identificarse, relacionarse y recuperarse posteriormente. Sin embargo, disponer de un historial no implica por sí mismo que dicho historial permita comprobar criptográficamente que la información permaneció inalterada.

En esta investigación se distinguieron tres conceptos relacionados. La trazabilidad se entendió como la posibilidad de reconstruir los hitos de un proceso; la integridad, como la propiedad de que la información no haya sido modificada de manera no detectada; y la verificabilidad, como la posibilidad de comprobar dicha integridad mediante evidencia independiente del estado operacional consultado.

Las funciones hash criptográficas constituyen un mecanismo básico para esta última propiedad. Una función como SHA-256 genera una representación de longitud fija a partir de una entrada y permite detectar modificaciones cuando el valor recalculado difiere del previamente registrado [8]. No obstante, almacenar el hash junto con los datos originales dentro del mismo dominio administrativo limita el grado de independencia de la evidencia. Por ello, ChainGov utilizó una capa adicional de registro distribuido para conservar evidencia criptográfica asociada a hitos seleccionados.

2.2 Blockchain y Hyperledger Fabric

Una blockchain mantiene un registro de transacciones cuya evolución es validada de acuerdo con reglas compartidas entre los participantes. En redes permisionadas, la participación está restringida a identidades conocidas y las reglas de validación pueden adaptarse al contexto organizacional. Hyperledger Fabric fue diseñado bajo este enfoque y proporciona mecanismos de identidad, políticas de endorsement, ejecución de chaincode y separación lógica de información [2].

Fabric resulta pertinente para escenarios institucionales en los que los participantes no son anónimos. Su Membership Service Provider (MSP) permite asociar identidades con organizaciones de la red, mientras que las políticas de endorsement establecen qué participantes deben respaldar una transacción antes de que pueda considerarse válida [2]. Los canales permiten mantener registros separados entre subconjuntos de organizaciones y las Private Data Collections posibilitan restringir la distribución de determinados datos, manteniendo evidencia de su existencia en el ledger [9].

El servicio de ordenamiento establece un orden común para las transacciones antes de su incorporación al ledger. Fabric admite implementaciones basadas en Raft, un protocolo tolerante a fallas por caída (Crash Fault Tolerant, CFT) [10]. Esta característica no debe confundirse con tolerancia bizantina: el uso de Raft permite mantener disponibilidad ante determinadas fallas de nodos, pero no proporciona por sí mismo protección frente a participantes arbitrariamente maliciosos.

Estas propiedades hicieron de Fabric una alternativa adecuada para el MVP de ChainGov, donde las organizaciones representadas fueron conocidas y la finalidad del ledger fue proporcionar evidencia distribuida de integridad, no sustituir el procesamiento transaccional de SQL Server.

2.3 Sistemas híbridos blockchain / base de datos

Las bases de datos tradicionales y las blockchains presentan propiedades distintas. Las primeras priorizan procesamiento transaccional, consulta y actualización eficiente del estado; las segundas incorporan mecanismos de replicación, validación y evidencia compartida que introducen costos adicionales. La literatura sobre

integración blockchain-base de datos ha estudiado distintas formas de combinar ambos modelos para evitar que todas las operaciones deban ejecutarse directamente sobre una blockchain [3] [4].

Una arquitectura híbrida puede mantener fuera de la cadena (off-chain) la información operacional y registrar en el ledger únicamente aquella evidencia que requiere verificabilidad distribuida. Este enfoque permite conservar capacidades propias de un sistema de gestión de bases de datos y, simultáneamente, disponer de pruebas criptográficas vinculadas a determinados estados o eventos [4].

La decisión introduce, sin embargo, un problema de consistencia entre ambos dominios. Si una operación se confirma en la base de datos, pero la evidencia correspondiente no alcanza el ledger, el sistema debe detectar y resolver esa diferencia. Por esta razón, el diseño de una arquitectura híbrida no se limita a seleccionar qué datos almacenar en cada tecnología: requiere definir mecanismos explícitos de sincronización, reintento, idempotencia y verificación.

2.4 Alternativas y justificación de la arquitectura

Para el problema estudiado podían considerarse distintas estrategias. Una solución exclusivamente relacional conserva simplicidad operacional y alto rendimiento, pero la evidencia histórica permanece administrada dentro del mismo dominio de persistencia. En el extremo opuesto, trasladar el procesamiento principal a una blockchain incrementaría la participación del registro distribuido, pero introduciría costos y restricciones innecesarios para operaciones que una base de datos relacional ya resuelve eficientemente.

Una tercera alternativa corresponde a las bases de datos verificables, que incorporan estructuras criptográficas para permitir comprobar propiedades del historial. Veritas, por ejemplo, estudió la incorporación de verificabilidad a sistemas de almacenamiento sin exigir que toda la gestión de datos adoptara el modelo convencional de una blockchain [5]. Estos enfoques demuestran que verificabilidad y procesamiento transaccional no necesitan resolverse necesariamente mediante una única tecnología.

En la misma línea, SQL Server 2022 incorporó tablas ledger. En ellas, las filas modificadas por cada transacción y las propias transacciones se resumen mediante hashes SHA-256 organizados en árboles de Merkle, y los resúmenes criptográficos resultantes de la base de datos (digests) pueden almacenarse en repositorios inmutables y utilizarse posteriormente para detectar alteraciones [11] [12]. Según su documentación, esta alternativa está orientada a verificar la integridad de datos administrados centralmente sin introducir el consenso de una red blockchain [12]. ChainGov, en cambio, buscó distribuir la evidencia entre organizaciones con identidades MSP diferenciadas y validación conjunta mediante la política de endorsement. La comparación experimental con tablas ledger no formó parte del alcance del estudio.

ChainGov adoptó una arquitectura híbrida porque el problema de investigación no requería reemplazar SQL Server. La base relacional mantuvo el estado operacional y Hyperledger Fabric se utilizó para distribuir evidencia criptográfica de hitos críticos entre organizaciones simuladas. De este modo, el diseño separó persistencia transaccional de verificabilidad distribuida, haciendo posible evaluar experimentalmente el costo asociado a incorporar esta segunda propiedad.

2.5 Integración y Transactional Outbox

Uno de los riesgos de una arquitectura orientada a eventos aparece cuando una operación debe modificar la base de datos y, además, publicar un mensaje en otro sistema. Si ambas acciones se realizan de manera independiente, una falla intermedia puede dejar el estado persistido sin su evento correspondiente o producir inconsistencias entre componentes.

El patrón Transactional Outbox aborda este problema almacenando el evento dentro de la misma transacción local que modifica los datos de negocio. Posteriormente, un proceso independiente recupera los eventos

pendientes y los publica hacia el sistema de mensajería [13]. Así se evita depender de una transacción distribuida entre la base de datos y el broker.

El patrón no implica semántica exactly-once de extremo a extremo. Los reintentos pueden producir entregas repetidas, por lo que los consumidores deben diseñarse para tolerar duplicados mediante mecanismos de idempotencia [13]. Esta distinción fue relevante para ChainGov: la arquitectura se diseñó bajo una semántica de entrega al menos una vez, complementada con controles para obtener un efecto lógico idempotente sobre el registro de evidencia.

2.6 Rendimiento y costo de verificabilidad

La incorporación de mecanismos distribuidos de validación tiene un costo que depende de la configuración, el número de participantes, las políticas utilizadas, el patrón de acceso y la carga aplicada. Por ello, las cifras de rendimiento obtenidas para una plataforma blockchain no deben interpretarse como propiedades constantes de la tecnología [3].

Trabajos experimentales han mostrado además que la relación entre bases de datos y blockchains debe evaluarse considerando las características concretas de la carga y de las garantías requeridas [14]. En una arquitectura híbrida, incluso cuando la escritura final en blockchain ocurre asincrónicamente, pueden existir costos sincrónicos asociados a canonicalización, cálculo de hashes, persistencia del evento y coordinación con los componentes de integración.

En consecuencia, evaluar ChainGov exigió medir no solo si la evidencia podía recuperarse y verificarse, sino también cuánto rendimiento transaccional se perdía al habilitar la intervención híbrida. Esta relación entre verificabilidad y costo operacional constituyó uno de los ejes de la validación experimental.

2.7 Sistemas híbridos y verificables relacionados

La literatura presenta diversas propuestas que aproximan las propiedades de bases de datos y blockchain desde perspectivas diferentes. BlockchainDB exploró una capa de base de datos compartida sobre blockchains con el objetivo de facilitar el acceso a almacenamiento descentralizado mediante abstracciones familiares para aplicaciones [15]. Veritas abordó la construcción de almacenamiento verificable, enfatizando que las propiedades de integridad pueden incorporarse sin adoptar necesariamente todos los mecanismos de una blockchain convencional [5].

Otros trabajos continuaron esta línea de convergencia. GriDB estudió una arquitectura orientada a combinar propiedades de bases de datos con un entorno distribuido basado en blockchain [16]. RollStore, por su parte, investigó mecanismos para mejorar la eficiencia de almacenamiento y procesamiento en sistemas que combinan garantías verificables con gestión de datos [17].

Estos trabajos muestran que no existe una única forma de integrar ambas familias tecnológicas. Algunas propuestas modifican el motor de almacenamiento, otras construyen una base de datos sobre infraestructura blockchain y otras separan el estado operacional de la evidencia verificable. ChainGov se posicionó en este último grupo: mantuvo un motor relacional convencional y añadió una capa de verificabilidad distribuida mediante integración asíncrona, sin modificar internamente SQL Server.

2.8 Blockchain en contratación y administración públicas

La aplicación de blockchain a contratación pública ha sido estudiada principalmente por su potencial para reforzar transparencia, trazabilidad y auditabilidad. Govindan et al. analizaron el uso de blockchain en procesos de contratación pública considerando los efectos que las características de esta tecnología pueden tener sobre

el funcionamiento de dichos procesos [6]. Otros trabajos han examinado su utilización para gestión documental y preservación de evidencia en contextos administrativos [7].

La literatura también ha señalado que la incorporación de blockchain en contratación pública no constituye únicamente una decisión tecnológica. Su adopción interactúa con procesos organizacionales, reglas institucionales, gobernanza y requisitos de implementación [18]. En administración pública, estudios sobre iniciativas de infraestructura digital, como las desarrolladas en Estonia, muestran asimismo que el valor de tecnologías de registro distribuido depende de su inserción dentro de estructuras institucionales y técnicas más amplias [19].

Por ello, ChainGov no se planteó como demostración de superioridad general de blockchain sobre una base de datos relacional ni como propuesta de cumplimiento jurídico para el sistema chileno de compras públicas. Su alcance fue técnico y experimental: estudiar una arquitectura que incorporara evidencia criptográfica distribuida a determinados hitos y cuantificar las consecuencias observables de esa decisión.

2.9 Brecha de investigación y posicionamiento

Los trabajos revisados proporcionan evidencia sobre sistemas híbridos, bases de datos verificables y aplicaciones de blockchain a procesos organizacionales [4] [5] [14]. Paralelamente, la investigación en contratación pública ha identificado oportunidades y restricciones asociadas al uso de blockchain [6] [18].

Sin embargo, para el problema abordado en esta investigación resultaba necesario analizar conjuntamente tres dimensiones: mantener un sistema relacional como fuente operacional, incorporar una capa permisionada de evidencia distribuida y evaluar experimentalmente tanto la verificabilidad obtenida como su costo sobre el rendimiento transaccional.

ChainGov se situó en esta intersección. Su contribución no consistió en proponer una nueva blockchain ni un nuevo motor de base de datos, sino en diseñar, implementar y someter a validación una arquitectura híbrida reproducible para un escenario controlado inspirado en compras públicas chilenas. Este posicionamiento permitió estudiar de manera explícita el compromiso entre verificabilidad y rendimiento.

2.10 Síntesis

El marco revisado permitió establecer cuatro decisiones fundamentales para el diseño. Primero, trazabilidad e integridad no fueron consideradas equivalentes a verificabilidad criptográfica independiente. Segundo, Hyperledger Fabric proporcionó mecanismos adecuados para representar participantes institucionales conocidos y distribuir evidencia entre ellos. Tercero, la conservación de SQL Server como sistema operacional evitó trasladar innecesariamente toda la carga transaccional a blockchain. Cuarto, esta separación exigió mecanismos explícitos de integración y una evaluación experimental del costo introducido.

Sobre estas bases se diseñó ChainGov. El capítulo siguiente describe su arquitectura, la distribución de responsabilidades entre componentes y los mecanismos implementados para generar, transportar, registrar y posteriormente verificar la evidencia asociada a los hitos críticos.

3 Diseño e implementación de ChainGov

3.1 Principios de diseño

ChainGov se diseñó como una arquitectura híbrida orientada a incorporar verificabilidad criptográfica sin sustituir las capacidades transaccionales de la base de datos relacional. Para ello se estableció una separación

explícita entre el estado operacional del proceso y la evidencia utilizada para verificar posteriormente la integridad de sus hitos.

El diseño siguió cuatro principios. Primero, SQL Server permaneció como fuente del estado transaccional, evitando trasladar a blockchain operaciones que no requerían consenso distribuido. Segundo, se registró en Hyperledger Fabric evidencia criptográfica de eventos seleccionados, en lugar de replicar indiscriminadamente toda la información de negocio. Tercero, la integración se realizó de manera asíncrona mediante Transactional Outbox, permitiendo desacoplar la transacción de negocio del anclaje posterior. Cuarto, los componentes se diseñaron considerando reintentos e idempotencia, debido a que la entrega de eventos no se modeló como exactly-once de extremo a extremo.

Esta separación permitió estudiar la verificabilidad como una capacidad adicional y medir experimentalmente el costo introducido por la condición híbrida.

3.2 Arquitectura general

La arquitectura implementada estuvo compuesta por SQL Server 2022, un worker desarrollado en .NET 8, Apache Kafka, un conector para Fabric desarrollado en Node.js, una red Hyperledger Fabric 2.5, chaincode implementado en Go y una interfaz de verificación.

El flujo principal comenzó en SQL Server. Una operación de negocio generó, dentro de la misma transacción local, el registro correspondiente en la tabla de salida transaccional. El Outbox Worker recuperó posteriormente los eventos pendientes y los publicó en Apache Kafka. El Fabric Connector consumió estos mensajes y gestionó su envío hacia Hyperledger Fabric. De esta manera, la confirmación de la operación relacional no quedó condicionada a completar sincrónicamente todo el proceso de consenso y registro distribuido.

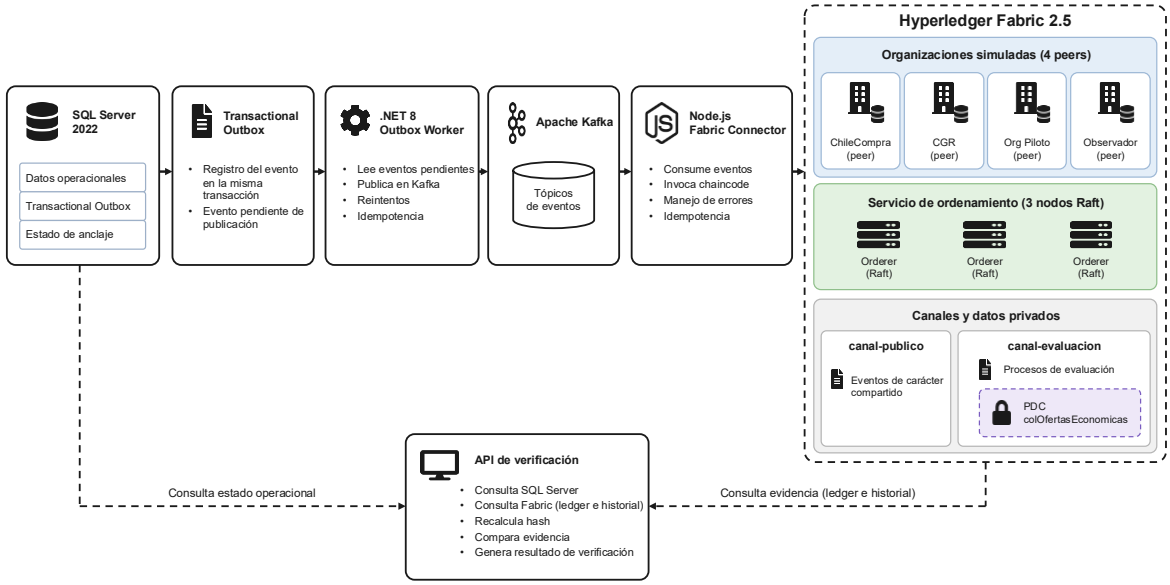


Figura 1. Arquitectura general de ChainGov

La arquitectura separó así dos responsabilidades: SQL Server mantuvo la información operacional necesaria para gestionar el proceso, mientras que Fabric conservó evidencia destinada a verificar la integridad histórica de los eventos seleccionados.

3.3 Persistencia, canonicalización y evidencia

Para que un hash pudiera utilizarse como evidencia, fue necesario asegurar que una misma representación lógica produjera una entrada determinista. Por ello, antes de calcular el hash se construyó una representación canónica de los atributos relevantes del evento. Sobre esta representación se aplicó SHA-256 [8].

La evidencia registrada permitió posteriormente volver a construir la representación correspondiente a un evento, recalcular su hash y contrastarlo con el valor previamente anclado. Una coincidencia indicó que la representación verificada era consistente con la evidencia registrada; una diferencia permitió detectar una alteración respecto de aquella evidencia.

Este mecanismo no convirtió el contenido original en secreto ni demostró por sí mismo la veracidad semántica del dato al momento de su creación. Su función fue proporcionar evidencia para detectar modificaciones posteriores respecto del valor que había sido anclado.

3.4 Transactional Outbox y sincronización

La integración entre SQL Server y la capa distribuida se implementó mediante Transactional Outbox [13]. El evento destinado a Fabric se persistió como parte de la misma transacción local que modificó el estado de negocio. Con ello se evitó realizar una escritura en SQL Server y una publicación externa como dos operaciones independientes sin coordinación.

Posteriormente, el Outbox Worker en .NET 8 recuperó eventos pendientes y los publicó en Apache Kafka. El uso de Kafka desacopló la generación del evento de su procesamiento por el componente encargado de interactuar con Fabric. El conector desarrollado en Node.js consumió los mensajes y realizó las operaciones correspondientes sobre la red blockchain.

La solución contempló reintentos, por lo que un evento podía volver a entregarse. En consecuencia, la arquitectura no afirmó proporcionar semántica exactly-once de extremo a extremo. Se utilizó una estrategia de al menos una vez con efecto lógico idempotente, evitando que la repetición de un mismo evento produjera múltiples efectos equivalentes en el estado verificable.

Una vez procesados los eventos, el estado asociado al anclaje permitió distinguir entre información todavía pendiente y evidencia cuya incorporación a la capa blockchain había sido confirmada o rechazada.

3.5 Red Hyperledger Fabric

La red experimental se implementó con Hyperledger Fabric 2.5 y representó cuatro organizaciones institucionales simuladas, cada una con un peer. El servicio de ordenamiento utilizó tres nodos Raft. Como se indicó en el marco teórico, Raft proporcionó tolerancia a fallas por caída y no se interpretó como un protocolo de tolerancia a fallas bizantinas [10].

Se configuraron dos canales. canal-publico se utilizó para información destinada al dominio compartido definido por el MVP, mientras que canal-evaluacion concentró operaciones relacionadas con la etapa de evaluación. Dentro de este último se utilizó la Private Data Collection colOfertasEconomicas para restringir la distribución de información sensible a los participantes autorizados, manteniendo las propiedades de verificación proporcionadas por Fabric [9].

La política de endorsement implementada para las operaciones relevantes requirió el respaldo conjunto de las organizaciones simuladas ChileCompra y CGR. Esta configuración representó el requisito de que la aceptación de determinadas operaciones no dependiera de una sola organización.

La topología tuvo finalidad experimental. Las organizaciones representaron dominios institucionales diferenciados lógicamente, pero el MVP no constituyó un despliegue productivo distribuido físicamente entre instituciones públicas reales.

3.6 Chaincode, secuencia e idempotencia

El chaincode se implementó en Go y gestionó el registro de evidencia asociada a los hitos considerados por el MVP. Además de validar los parámetros necesarios para cada operación, aplicó reglas sobre la secuencia esperada del proceso.

La máquina de estados utilizada contempló la progresión:

CREADA → PUBLICADA → RECEPCION_OFERTAS → APERTURA → EVALUACION → ADJUDICADA → CONTRATO_FIRMADO

También se consideró el estado terminal ANULADA, alcanzable desde CREADA, PUBLICADA, RECEPCION_OFERTAS, APERTURA, EVALUACION o ADJUDICADA, y sujeto al registro obligatorio de un motivo de anulación. CONTRATO_FIRMADO y ANULADA no admitieron transiciones posteriores.

El identificador lógico de los eventos y las validaciones del chaincode se utilizaron para evitar que reintentos provenientes de la integración generaran efectos duplicados. Esta propiedad resultó necesaria debido a la semántica de entrega adoptada por la capa de mensajería e integración.

La validación de secuencia no pretendió modelar exhaustivamente todas las reglas de contratación pública chilena. Correspondió a una abstracción técnica del proceso utilizada para implementar y evaluar el MVP.

3.7 Privacidad y commit-reveal

Determinada información asociada a ofertas económicas requirió un tratamiento diferente al de los eventos de carácter compartido. Para este propósito se utilizó colOfertasEconomicas dentro de canal-evaluacion, aprovechando el mecanismo de datos privados de Fabric [9].

Adicionalmente, se implementó un esquema commit-reveal. Durante la fase de compromiso se registró evidencia criptográfica asociada al contenido sin exponer anticipadamente su valor. Posteriormente, durante la fase de revelación, el contenido correspondiente podía contrastarse con el compromiso previamente registrado.

El mecanismo permitió representar técnicamente la separación entre compromiso y revelación requerida por el escenario experimental. No se interpretó como una demostración de cumplimiento normativo ni como sustituto de los controles jurídicos y organizacionales aplicables a un sistema productivo de contratación pública.

3.8 Verificación de integridad

La capa de verificación permitió contrastar información mantenida en SQL Server con la evidencia disponible en Hyperledger Fabric. Para ello se reconstruyó la representación canónica correspondiente, se recalculó el hash y se comparó con la evidencia anclada.

La verificación clasificó los casos en estados tales como INTEGRA, cuando la información contrastada coincidía con la evidencia registrada, y ALTERADA, cuando se detectaba una diferencia. También se contemplaron los estados PENDIENTE_ANCLAJE y ANCLAJE_RECHAZADO, necesarios para distinguir una inconsistencia criptográfica de un evento cuyo procesamiento distribuido todavía no había concluido satisfactoriamente.

Además de comprobar el estado de eventos individuales, la implementación utilizó las capacidades de historial del ledger de Fabric, mediante GetHistoryForKey, para recuperar la secuencia verificable asociada a las claves evaluadas. Esta capacidad constituyó la base técnica de la medición de completitud verificable utilizada posteriormente como VD3.

La distinción entre estados evitó interpretar automáticamente la ausencia de un anclaje confirmado como una alteración de datos, mejorando la precisión semántica del procedimiento de auditoría.

3.9 Seguridad y alcance del MVP

ChainGov fue implementado como un MVP experimental. Su objetivo fue validar la arquitectura y generar evidencia reproducible sobre verificabilidad y rendimiento, no demostrar que el sistema estuviera preparado para explotación productiva.

El modelo incorporó mecanismos de identidad y respaldo proporcionados por Fabric, separación lógica mediante canales y datos privados, hashes SHA-256, controles de secuencia e idempotencia. Sin embargo, la evaluación no constituyó una auditoría integral de ciberseguridad ni una certificación de cumplimiento normativo.

Asimismo, las organizaciones y actores utilizados durante la experimentación fueron simulados. Por ello, los resultados obtenidos deben interpretarse dentro del entorno controlado definido para el estudio y no como evidencia directa del comportamiento que tendría un despliegue interinstitucional real.

Estas delimitaciones fueron relevantes para la validez externa: permitieron evaluar las propiedades técnicas centrales de la propuesta sin atribuir al prototipo garantías que no fueron sometidas a validación.

3.10 Síntesis

ChainGov separó la persistencia operacional de la evidencia distribuida de integridad. SQL Server mantuvo el procesamiento transaccional; Transactional Outbox, el worker y Kafka transportaron los eventos de forma desacoplada; el conector interactuó con Hyperledger Fabric; y el chaincode registró y validó la evidencia correspondiente.

La implementación incorporó canonicalización, SHA-256, idempotencia, separación de información mediante canales y datos privados, commit-reveal y recuperación de historial verificable. Sobre esta arquitectura se construyó la validación experimental descrita en el capítulo siguiente.

4 Metodología de validación

4.1 Diseño experimental

La validación se realizó mediante un diseño cuasiexperimental que comparó dos condiciones arquitectónicas sobre el mismo escenario funcional. La variable independiente correspondió a la condición arquitectónica experimental, con dos niveles: T0, línea base basada en SQL Server, y T1, arquitectura híbrida ChainGov.

La evaluación consideró tres variables dependientes. VD1 correspondió al tiempo requerido para ejecutar la auditoría de trazabilidad; VD2, al rendimiento transaccional bajo carga sostenida; y VD3, a la completitud verificable de los eventos procesados. Adicionalmente, en las pruebas de rendimiento se registraron percentiles de latencia P50, P95 y P99.

La hipótesis H1 se evaluó mediante dos criterios preespecificados: C1, mejora del tiempo de auditoría igual o superior al 40%, y C2, degradación del rendimiento transaccional no superior al 15%. Ambos criterios debían cumplirse simultáneamente para respaldar H1.

Tabla 1. Operacionalización de las variables

Variable	Definición operacional	Indicador
VI	Condición arquitectónica	T0: SQL Server; T1: ChainGov
VD1	Ejecución de auditoría de trazabilidad	Tiempo de auditoría
VD2	Rendimiento bajo carga sostenida	Operaciones de negocio por segundo
VD3	Eventos cuya evidencia histórica pudo verificarse respecto del total esperado	Completitud verificable (%)
Complementaria	Latencia observada durante VD2	P50, P95 y P99 (ms)

VD3 se utilizó como indicador complementario de la capacidad verificable de la arquitectura. No se estableció retrospectivamente un umbral para esta variable, ya que no formó parte de los criterios cuantitativos originales de H1.

4.2 Entorno e instrumentación

El experimento se ejecutó sobre el MVP descrito en el Capítulo 3. La condición T0 mantuvo el procesamiento relacional utilizado como línea base, mientras que T1 habilitó la intervención asociada a ChainGov. En esta última, el procesamiento sincrónico incluyó las operaciones necesarias para generar la representación del evento, calcular su evidencia criptográfica y persistir el registro correspondiente en el Outbox. El transporte mediante Kafka y el posterior procesamiento en Fabric se realizaron asincrónicamente.

La instrumentación experimental definió previamente los tamaños de las pruebas, clases de casos, niveles de concurrencia, repeticiones y criterios de evaluación. Los objetos pertenecientes al esquema experimental exp.* se utilizaron para preparar y controlar la experimentación. Estos objetos no constituyeron una fuente histórica independiente disponible para T0 y, por tanto, no fueron considerados evidencia que permitiera atribuir a la línea base capacidades de auditoría que no formaban parte de su implementación.

Para VD1 se fijó un conjunto de 500 casos. Para VD2 se utilizaron cinco niveles de concurrencia (10, 50, 100, 250 y 500 usuarios) con tres repeticiones oficiales por condición y nivel. La configuración del sistema y de la carga se mantuvo congelada durante la ejecución para evitar modificar el protocolo en función de resultados parciales; la única modificación afectó al criterio operativo de aceptación del warm-up y se describe en la sección 4.6. El MVP completo y el generador de carga se ejecutaron en un mismo equipo, con un procesador Intel Core i7-7820X de 8 núcleos y 16 hilos y 31 GiB de memoria asignados al entorno WSL2 de Windows, sobre Docker Desktop.

4.3 VD1: Auditoría y ground truth

VD1 evaluó el mecanismo de auditoría implementado en ChainGov. El conjunto experimental estuvo compuesto por 500 casos distribuidos en cuatro clases: A0, 200 casos sin alteración; A1, 100 casos con modificación directa de su contenido en SQL Server; A2, 100 casos con regresión de su versión de anclaje; y A3, 100 casos cuyo registro fue eliminado de SQL Server. Esta distribución permitió disponer tanto de casos que debían conservar su integridad como de casos sometidos a las alteraciones definidas en el protocolo experimental.

El ground truth se mantuvo separado del mecanismo auditor. Su función fue conservar la clasificación esperada de los casos para contrastarla posteriormente con el resultado producido por ChainGov. De esta manera, la

evaluación de concordancia no dependió de que el propio auditor definiera simultáneamente la respuesta esperada.

La prueba contempló una medición T1 previa a las alteraciones y una auditoría posterior sobre el conjunto experimental intervenido. Se registraron el número de casos auditados, las clasificaciones obtenidas, la cantidad de eventos verificables respecto del total esperado y el tiempo empleado por la ejecución.

Esta estrategia permitió evaluar dos aspectos diferentes: la capacidad de ChainGov para clasificar los casos de acuerdo con el ground truth y la capacidad de recuperar la evidencia histórica necesaria para realizar la verificación.

4.4 Limitación de equivalencia funcional de C1

El criterio temporal C1 requería comparar el tiempo de auditoría de T0 y T1. Durante el análisis metodológico se identificó, sin embargo, una limitación fundamental: la condición T0 implementada no disponía de una fuente histórica independiente equivalente al ledger utilizado por ChainGov para comprobar retrospectivamente si un estado había sido alterado.

Una consulta al estado relacional actual de T0 no constituía una tarea funcionalmente equivalente a la verificación realizada en T1. Medir únicamente dicha consulta habría comparado operaciones con semánticas distintas y habría favorecido artificialmente a la línea base. Del mismo modo, asignar a T0 un tiempo infinito, un timeout arbitrario o construir retrospectivamente un historial que no existía en la implementación habría modificado el experimento después de su ejecución.

La condición T0 se ejecutó igualmente sobre los mismos 500 casos, mediante un procedimiento que consultó el estado relacional vigente de licitaciones y ofertas. La ejecución registró 400 casos existentes, 100 inexistentes y ningún caso verificable históricamente, con una duración de 12,249 ms medida en el servidor de base de datos. Este resultado confirmó la limitación: T0 solo pudo constatar la ausencia de los registros eliminados y no dispuso de evidencia para distinguir los casos íntegros de los modificados o con regresión de versión. Además, su tiempo comprendió únicamente la ejecución del bloque SQL, mientras que el de T1 incluyó la consulta del historial en el ledger y su contraste con SQL Server, medidos en la API de verificación. Por ambas razones, este valor no se utilizó para calcular C1. El resultado se conservó como registro de la salida de consola, ya que la tabla experimental que lo almacenó no formó parte de los artefactos exportados.

En consecuencia, VD1 se ejecutó en ambas condiciones, pero solo T1 produjo evidencia de clasificación de integridad histórica y de completitud verificable; C1 quedó no determinado bajo una comparación T0/T1 funcionalmente equivalente.

Esta decisión preservó la validez de constructo del estudio: el indicador temporal no fue sustituido por una operación distinta solo para obtener un valor cuantitativo.

4.5 VD2: Rendimiento transaccional

VD2 evaluó el impacto de la condición híbrida sobre el rendimiento transaccional mediante Apache JMeter y acceso JDBC. Se utilizaron cinco niveles de concurrencia: U = {10, 50, 100, 250, 500} usuarios concurrentes.

Para cada nivel se realizaron tres repeticiones oficiales por condición, T0 y T1, lo que produjo 30 corridas principales de medición. Cada configuración consideró un período de calentamiento de cinco minutos y posteriormente una ventana oficial de medición de 30 minutos.

T0 y T1 ejecutaron la misma carga de negocio. La diferencia experimental correspondió a la intervención arquitectónica: en T0 no se encontraba habilitado el procesamiento sincrónico asociado al Outbox de ChainGov,

mientras que en T1 sí se ejecutaban las operaciones necesarias para producir y persistir dicha evidencia. El procesamiento posterior mediante Kafka y Fabric permaneció desacoplado del cierre de la operación de negocio. Por la misma razón, las latencias P50, P95 y P99 correspondieron al tiempo de respuesta de un ciclo completo de diez procedimientos y no al de una operación individual.

Una muestra de JMeter representó un ciclo completo compuesto por diez llamadas a procedimientos de negocio. Por esta razón se distinguieron dos unidades: ciclos completos por segundo y operaciones de negocio por segundo, calculadas como:

$$Operaciones\ de\ negocio\ (por\ seg) = ciclos\ (por\ seg)\ \times 10 \tag{1}$$

En el informe se utilizó principalmente la segunda medida para expresar el throughput, evitando interpretar una muestra de JMeter como una única operación de negocio.

4.6 Métricas y criterio C2

Para cada corrida se obtuvieron throughput, latencias P50, P95 y P99 y número de errores. Debido a que se realizaron tres repeticiones por configuración, el análisis final se mantuvo descriptivo, sin introducir pruebas de significancia estadística que no habían sido previstas para un número tan reducido de repeticiones. Como medida consolidada de cada condición y nivel se utilizó la mediana de las tres repeticiones, menos sensible que la media a una repetición atípica. Dado que el protocolo congelado había previsto la media aritmética, la degradación también se calculó con medias como análisis de sensibilidad.

La degradación de throughput para cada nivel de concurrencia se calculó como:

$$\text{Degradación}(\%) = \frac{TPS_{T0} - TPS_{T1}}{TPS_{T0}} \times 100 \tag{2}$$

El criterio C2 exigió que la degradación no superara el 15% en los niveles de carga preespecificados. Para evitar seleccionar posteriormente una carga favorable, la evaluación se realizó conservadoramente sobre el conjunto completo U = {10, 50, 100, 250, 500}. Por tanto:

$$C2:\ \max_{u \in U}\big(\text{Degradación}(u)\big) \leq 15\% \tag{3}$$

Los períodos de calentamiento se utilizaron para estabilizar el entorno y no fueron tratados como corridas oficiales de medición. Inicialmente, el procedimiento de ejecución exigía cero errores durante los cinco minutos de warm-up. Tras tres intentos abortados por errores aislados de tiempo de espera de 60 s al inicio de la rampa, uno de ellos en T0, y con dos corridas oficiales ya aceptadas, el criterio se ajustó para exigir cero errores en los últimos 120 s del warm-up. Seis corridas T1, de 250 y 500 usuarios, registraron entre 4 y 65 errores de este tipo, equivalentes a lo más al 0,19% de sus muestras de warm-up y todos iniciados dentro de sus primeros 10 s. Las 24 corridas restantes también cumplieron el criterio original, y ninguna corrida oficial registró errores en su ventana de medición.

4.7 VD3: Completitud verificable

VD3 midió qué proporción de los eventos esperados disponía de evidencia recuperable y verificable en la capa distribuida. Se definió como:

$$Completitud\ verificable(\%) = \frac{eventos\ verificables}{eventos\ esperados} \times 100 \tag{4}$$

La recuperación del historial se apoyó en la capacidad GetHistoryForKey utilizada por la implementación de Fabric. Esta medición permitió distinguir la clasificación de integridad de la disponibilidad del historial necesario para efectuar la comprobación.

VD3 complementó a VD1: una clasificación correcta de los casos no debía interpretarse automáticamente como demostración de que todos los eventos esperados estuvieran disponibles. Por ello, ambas dimensiones se reportaron separadamente.

4.8 Evaluación de H1

La hipótesis fue formulada como una condición conjuntiva. Su evaluación se realizó con los criterios definidos antes del análisis de resultados.

Los umbrales de C1 y C2 se establecieron antes de la experimentación, como criterios de decisión que hacían falsable a H1 y no se modificaron tras conocer los resultados. El 40% fijó el nivel de mejora considerado sustantivo para el tiempo de auditoría: suficiente para justificar la incorporación de una capa adicional de infraestructura y no atribuible a la variabilidad del entorno. El 15% operacionalizó el umbral aceptable de degradación enunciado en la pregunta de investigación, entendido como el máximo tolerable para un sistema transaccional en operación; fijarlo previamente resultó necesario porque el costo de una arquitectura híbrida depende del diseño, la configuración y la carga evaluadas [14]. Ninguno de los dos valores correspondió a una estimación derivada de mediciones previas del sistema evaluado.

Tabla 2. Criterios de evaluación de H1

Criterio	Regla preespecificada
C1	Mejora del tiempo de auditoría ≥ 40%
C2	Máxima degradación del throughput ≤ 15%
H1	Requiere cumplimiento simultáneo de C1 y C2

La mejora temporal prevista para C1 se había definido como:

$$Mejora\ de\ auditoría(\%) = \frac{T_{T0} - T_{T1}}{T_{T0}} \times 100 \quad (5)$$

Sin embargo, esta expresión solo podía aplicarse si T0 y T1 medían tareas funcionalmente equivalentes. Debido a la limitación expuesta en la sección 4.4, C1 se reportó como no determinado bajo una comparación funcionalmente equivalente, en lugar de imputarle un valor artificial.

La imposibilidad de determinar C1 no fue interpretada como evidencia de que ChainGov empeorara o mejorara el tiempo de auditoría. La decisión respecto de H1 se realizó conservando su regla conjuntiva y utilizando únicamente los resultados que podían sostenerse con la evidencia experimental disponible.

4.9 Validez, confiabilidad y alcance

La validez interna se reforzó manteniendo niveles de concurrencia, duración, número de repeticiones y carga de negocio previamente definidos para ambas condiciones. El uso del mismo escenario funcional redujo diferencias ajenas a la intervención arquitectónica.

Una amenaza relevante a la validez interna fue la ejecución del generador de carga y de todos los componentes del MVP en un mismo equipo. En T1, el worker, Kafka, el conector, los peers, los orderers y CouchDB procesaron eventos durante la carga y pudieron competir por CPU, memoria y E/S con SQL Server y JMeter,

mientras que en T0 esos componentes permanecieron activos, pero sin eventos que procesar. Por ello, la degradación observada pudo incluir el efecto de esta contención y no solo el costo sincrónico de la intervención.

La validez de constructo se protegió distinguiendo trazabilidad, clasificación de integridad, completitud verificable y rendimiento. En particular, no se sustituyó la auditoría histórica de T0 por una consulta relacional de menor alcance para calcular C1.

La confiabilidad se favoreció mediante configuraciones experimentales congeladas, tres repeticiones oficiales por nivel de carga, registro de métricas y conservación de los artefactos de ejecución. El ground truth de VD1 permaneció separado del procedimiento auditor, permitiendo contrastar resultados esperados y observados.

La validez externa fue más limitada. El experimento se realizó sobre un MVP controlado, con datos y actores simulados y una red que representó lógicamente a distintas organizaciones. En consecuencia, los resultados caracterizan la implementación y configuración evaluadas y no deben generalizarse directamente a un despliegue productivo interinstitucional.

4.10 Consideraciones éticas y reproducibilidad

La experimentación no utilizó participantes humanos ni encuestas y se realizó con actores y datos simulados. Las denominaciones institucionales empleadas en el MVP tuvieron una función representativa dentro del escenario experimental y no implicaron participación, autorización o despliegue de esas instituciones en la ejecución del estudio.

Para favorecer la transparencia, se conservaron las configuraciones experimentales, resultados de las corridas y artefactos utilizados para analizar VD1, VD2 y VD3. La presentación de resultados distinguió las corridas oficiales de períodos de calentamiento y ejecuciones no pertenecientes al conjunto final de medición.

Se utilizó inteligencia artificial generativa como apoyo en la estructuración, revisión y reformulación de borradores. No se utilizó para generar datos experimentales ni para sustituir la verificación de fuentes. El autor verificó contenidos, referencias y resultados y asumió la responsabilidad integral sobre el informe.

4.11 Síntesis

La metodología comparó T0 y T1 mediante indicadores de auditoría, rendimiento y completitud verificable. VD1 utilizó 500 casos y un ground truth independiente; VD2 comprendió 30 corridas oficiales distribuidas entre cinco niveles de concurrencia y dos condiciones; y VD3 cuantificó la disponibilidad de la evidencia histórica esperada.

La evaluación mantuvo explícita la limitación que impidió determinar C1 mediante tareas T0/T1 funcionalmente equivalentes. En lugar de reconstruir retrospectivamente una línea base, esta restricción se incorporó como parte del análisis de validez. El capítulo siguiente presenta los resultados obtenidos y su relación con los criterios definidos para H1.

5 Resultados y discusión

5.1 Resultados experimentales

La evaluación produjo evidencia para las tres variables dependientes definidas en el Capítulo 4. VD1 permitió evaluar la clasificación de integridad y el tiempo de auditoría en ChainGov; VD2 cuantificó el impacto de la condición híbrida sobre el rendimiento transaccional; y VD3 midió la completitud verificable del conjunto experimental.

Los resultados se presentan utilizando únicamente las corridas y artefactos correspondientes al protocolo final. Los períodos de calentamiento y ejecuciones interrumpidas o preliminares no se incorporaron a las métricas oficiales. Entre estas últimas se encontró la prueba piloto final, de 60 s por nivel de concurrencia y condición, sin warm-up, destinada a validar la instrumentación y dimensionar la carga; sus diez ejecuciones finalizaron sin errores.

5.2 Resultados de VD1

La auditoría final de T1 procesó los 500 casos definidos para el experimento. De ellos, 200 fueron clasificados como INTEGRA y 300 como ALTERADA. La clasificación obtenida presentó concordancia con los 500 casos del ground truth experimental.

Asimismo, se recuperaron y verificaron los 3500 eventos esperados, correspondientes a siete hitos por cada uno de los 500 casos. El tiempo total registrado para la auditoría final fue de 3963,57 ms.

Antes de aplicar las alteraciones experimentales se había ejecutado una medición T1 sobre el mismo número de casos. En esa ejecución los 500 casos fueron clasificados como INTEGRA, se verificaron igualmente 3500 de 3500 eventos y el tiempo observado fue de 3963,13 ms.

Tabla 3. Resultados de auditoría VD1 en T1

Medición	Casos	INTEGRA	ALTERADA	Eventos verificables	Completitud	Tiempo
Previa a alteraciones	500	500	0	3500/3500	100%	3963,13 ms
Final	500	200	300	3500/3500	100%	3963,57 ms

La diferencia entre ambos tiempos fue de 0,44 ms. Esta diferencia se reporta únicamente de manera descriptiva como estabilidad temporal entre dos ejecuciones T1 de igual tamaño. No constituye una comparación T0/T1 ni permite calcular C1.

Los resultados sí mostraron que, dentro del escenario experimental, ChainGov fue capaz de distinguir los casos alterados de los íntegros de acuerdo con el ground truth y de recuperar la totalidad de los eventos esperados para realizar dicha verificación.

5.3 Resultados del criterio C1

El criterio C1 establecía una mejora del tiempo de auditoría igual o superior al 40% respecto de T0. Sin embargo, como se explicó en la sección 4.4, T0 no disponía de una capacidad histórica independiente funcionalmente equivalente a la utilizada por T1.

Por esta razón, no se utilizó como línea base el tiempo de una consulta al estado relacional actual, ya que dicha operación no verificaba si los registros habían sido modificados respecto de evidencia histórica independiente. Tampoco se asignó a T0 un valor artificial, infinito o derivado de un timeout. La ejecución de T0 descrita en la sección 4.4 confirmó esta diferencia de capacidad: ninguno de los 500 casos resultó verificable históricamente.

En consecuencia, el resultado de C1 fue: no determinado bajo una comparación funcionalmente equivalente.

Este resultado no significa que VD1 no se haya ejecutado ni constituye evidencia de una mejora o empeoramiento temporal de ChainGov. Indica específicamente que el porcentaje definido originalmente para C1 no pudo estimarse con validez utilizando las capacidades implementadas en ambas condiciones.

5.4 Resultados de VD3

La completitud verificable obtenida durante la auditoría final fue:

$$VD3 = (3500 / 3500) \times 100 = 100\%$$

Por tanto, todos los eventos esperados del conjunto experimental utilizado para VD1 pudieron recuperarse y verificarse mediante la evidencia disponible en ChainGov.

Este resultado debe interpretarse dentro del alcance del experimento. El 100% corresponde al conjunto controlado de 500 casos y no constituye una garantía de completitud para cargas, fallas o despliegues distintos. Asimismo, VD3 no poseía un umbral preespecificado dentro de H1, por lo que este valor se reportó como evidencia complementaria y no como un criterio añadido posteriormente para aceptar la hipótesis. Además, VD3 se midió una vez completado el anclaje del conjunto de VD1 y sin carga concurrente, por lo que no caracteriza la completitud alcanzable bajo la carga sostenida de VD2 descrita en la sección 5.5.

5.5 Resultados de VD2

Las 30 corridas oficiales de VD2 finalizaron sin errores durante sus ventanas de medición. En todos los niveles de concurrencia se observó un throughput inferior en T1 respecto de T0.

Tabla 4. Resultados de rendimiento por nivel de concurrencia

Usuarios	T0 (op/s)	T1 (op/s)	Degradación	P95 T0 (ms por ciclo)	P95 T1 (ms por ciclo)
10	3181,73	1056,84	66,78%	41	187
50	5238,64	1207,95	76,94%	133	817
100	5922,31	1145,45	80,66%	237	1614
250	5386,43	1195,52	77,81%	628	2718
500	5170,59	1072,91	79,25%	1182	5808

En T0, el mayor throughput mediano se observó con 100 usuarios concurrentes, alcanzando 5922,31 op/s. En T1, los valores se mantuvieron aproximadamente entre 1057 y 1208 op/s para las cargas evaluadas. La máxima degradación relativa se produjo con 100 usuarios y alcanzó 80,66%.

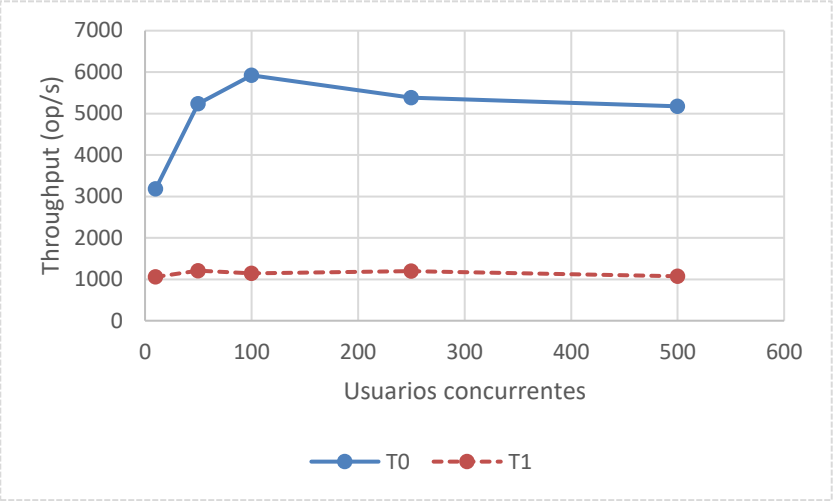


Figura 2. Throughput mediano de operaciones de negocio en T0 y T1 según concurrencia

Aplicando el criterio definido en la sección 4.6:

$$\max_{u \in U} \text{Degradación}(u) = 80{,}66\% > 15\%$$

Por tanto, C2 no se cumplió.

La magnitud observada superó ampliamente el umbral preespecificado, por lo que esta conclusión no dependió de seleccionar un nivel particular de concurrencia favorable o desfavorable.

Calculada con medias aritméticas, la degradación osciló entre 66,86% y 80,61%, por lo que la conclusión sobre C2 no dependió de la medida de consolidación utilizada.

Las corridas T1 permitieron observar, además, el procesamiento asincrónico, que no formaba parte de las variables preespecificadas. Cada ciclo de negocio generó 15 eventos en el Outbox, de modo que la carga produjo entre 1134 y 2053 eventos/s durante las ventanas de medición. En el mismo período, el worker publicó en Kafka entre 84 y 110 eventos/s y la red Fabric confirmó entre 3,7 y 5,0 eventos/s. Al término de cada corrida permanecían entre 2,38 y 4,07 millones de eventos pendientes de publicación, y los eventos anclados representaban entre 0,24% y 0,36% del total generado. Por tanto, el throughput de T1 correspondió al registro de las operaciones con evidencia diferida: bajo carga sostenida, la capacidad de anclaje fue más de dos órdenes de magnitud inferior a la tasa de generación de eventos, y los eventos pendientes de publicación se acumularon de forma creciente durante toda la ventana de medición. Mientras un evento permanece pendiente, su verificación devuelve un veredicto de anclaje pendiente, de modo que la evidencia distribuida todavía no está disponible para un tercero.

Esta capacidad no se aisló experimentalmente. Resulta compatible con la configuración del conector evaluado, que procesó secuencialmente los mensajes de cada partición de Kafka, con un máximo de seis particiones concurrentes, y esperó la confirmación del commit de cada transacción antes de continuar, con un BatchTimeout de 1 s en el servicio de ordenamiento.

5.6 Resultados de latencia

La latencia P95 por ciclo de negocio aumentó con la concurrencia en ambas condiciones, pero el crecimiento fue mayor en T1. Con 10 usuarios, P95 fue de 41 ms en T0 y 187 ms en T1. Con 500 usuarios, alcanzó 1182 ms y 5808 ms, respectivamente.

El comportamiento de P95 es consistente con la diferencia de throughput observada: al aumentar la carga, T1 sostuvo un número menor de operaciones de negocio por segundo y presentó mayores tiempos en la cola superior de la distribución de latencias.

No obstante, estos resultados corresponden a la condición T1 completa. El experimento no incorporó una prueba de ablación que aislara individualmente el costo de canonicalización, SHA-256, escritura en Outbox, middleware u otros componentes. Por ello, la diferencia observada no se atribuye exclusivamente a Hyperledger Fabric ni a un componente particular.

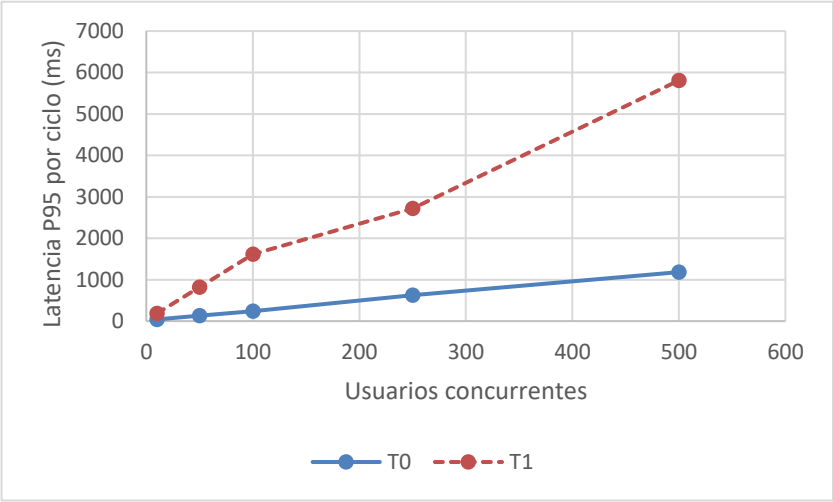


Figura 3. Latencia P95 por ciclo de negocio en T0 y T1 según concurrencia

5.7 Evaluación de H1

La evaluación final se realizó utilizando exclusivamente los criterios definidos antes del análisis.

Tabla 5. Evaluación de la hipótesis H1

Criterio	Umbral	Resultado	Evaluación
C1	Mejora de auditoría ≥40%	No determinado bajo equivalencia funcional	No cuantificable
C2	Degradación máxima ≤15%	80,66%	No cumple
H1	C1 y C2 simultáneamente	C2 no cumplió	No respaldada

En consecuencia, H1 no quedó respaldada bajo los criterios preespecificados. Esta conclusión se sustenta en que H1 exigía simultáneamente ambos criterios y C2 fue incumplido de forma observable.

La indeterminación de C1 debe interpretarse separadamente. No se utilizó como evidencia negativa contra ChainGov ni como causa del incumplimiento de C2. Constituyó una limitación del diseño comparativo original que impidió establecer válidamente el porcentaje de mejora temporal previsto.

5.8 Discusión

Los resultados muestran un compromiso entre verificabilidad y rendimiento. ChainGov permitió disponer de evidencia distribuida con la cual clasificar correctamente los 500 casos del conjunto experimental y recuperar los 3500 eventos esperados. Al mismo tiempo, habilitar la condición híbrida produjo una degradación considerable del throughput frente a T0.

Este comportamiento es coherente con la literatura que advierte que la incorporación de propiedades verificables y mecanismos distribuidos introduce costos que dependen de la arquitectura, configuración y carga evaluadas [3] [4] [14]. Los resultados de este estudio no deben interpretarse como una medida general del rendimiento de Hyperledger Fabric, sino como evidencia correspondiente a la implementación concreta de ChainGov.

Desde la perspectiva arquitectónica, los resultados respaldan la viabilidad de separar persistencia operacional y evidencia verificable, enfoque relacionado con sistemas híbridos y verificables estudiados previamente [4] [5]. Sin embargo, la implementación evaluada no satisfizo el límite de rendimiento establecido para H1. Esto muestra que la viabilidad funcional de una arquitectura híbrida no implica automáticamente que su costo operacional resulte aceptable para los criterios de un escenario determinado.

La observación sobre la capacidad de anclaje matiza este resultado. El desacoplamiento mediante Outbox y Kafka evitó que la confirmación en Fabric limitara directamente el throughput de T1, pero trasladó el costo al rezago de la evidencia, que bajo carga sostenida no se mantuvo acotado. En consecuencia, la aceptabilidad de una arquitectura de este tipo depende tanto de la degradación sincrónica como de la capacidad de anclaje frente a la tasa de generación de eventos.

La indeterminación de C1 produjo además un hallazgo metodológico. Comparar el tiempo de dos mecanismos solo resulta válido cuando ambos resuelven una tarea equivalente. En T0 podía consultarse el estado actual de los datos, pero no comprobar retrospectivamente ese estado contra una fuente histórica independiente equivalente al ledger. Convertir una consulta SQL de menor alcance en la línea base temporal habría aumentado la aparente comparabilidad numérica a costa de reducir la validez de constructo.

Por ello, una evaluación futura debería separar explícitamente dos dimensiones: capacidad de verificación histórica y tiempo de ejecución entre mecanismos que ya posean dicha capacidad. Esta separación evitaría exigir una comparación temporal directa entre sistemas que ofrecen garantías funcionales diferentes.

5.9 Limitaciones

Los resultados poseen varias limitaciones. Primero, el estudio se ejecutó en un entorno controlado y con actores y datos simulados, por lo que no representa directamente un despliegue productivo entre instituciones públicas.

Segundo, VD2 utilizó tres repeticiones por configuración. Esta cantidad permitió una comparación descriptiva consistente con el protocolo definido, pero no se utilizó para inferencia estadística.

Tercero, el experimento comparó las condiciones agregadas T0 y T1, sin pruebas de ablación destinadas a determinar qué componente explicó cada fracción de la degradación observada. Por esta razón, no es posible asignar causalmente el 80,66% a Fabric, Kafka, el Outbox o cualquier componente individual.

Cuarto, el diseño original de C1 asumió que T0 permitiría establecer una línea base temporal funcionalmente equivalente para auditoría histórica. La implementación mostró que dicha equivalencia no estaba disponible, impidiendo calcular válidamente el porcentaje de mejora previsto.

Quinto, el generador de carga y todos los componentes compartieron un mismo equipo, y la capacidad de anclaje asincrónico no formó parte de las variables preespecificadas; por ello, el estudio no determinó el rezago admisible de la evidencia ni la capacidad de anclaje de configuraciones alternativas del conector.

Finalmente, las organizaciones de Fabric fueron representadas lógicamente dentro del entorno experimental. No se evaluaron latencias de red, políticas operacionales ni condiciones de infraestructura propias de un despliegue físico interinstitucional.

5.10 Síntesis

ChainGov auditó correctamente los 500 casos del conjunto experimental final, diferenciando 200 casos íntegros y 300 alterados, con concordancia respecto del ground truth. La completitud verificable fue de 3500/3500 eventos (100%).

C1 no pudo determinarse mediante una comparación T0/T1 funcionalmente equivalente. En rendimiento, T1 presentó menor throughput y mayor P95 en todos los niveles evaluados. La degradación máxima alcanzó 80,66%, superando el límite de 15%, por lo que C2 no se cumplió.

En consecuencia, H1 no quedó respaldada bajo los criterios preespecificados. Los resultados demostraron la capacidad funcional de la arquitectura para incorporar verificabilidad al escenario experimental, pero también identificaron un costo de rendimiento que deberá abordarse antes de considerar una aplicación de características productivas.

6 Conclusiones

6.1 Conclusiones generales y respuesta a la pregunta de investigación

Esta investigación diseñó, implementó y evaluó ChainGov, una arquitectura híbrida blockchain-base de datos orientada a incorporar verificabilidad criptográfica a la trazabilidad de hitos críticos de procesos de compras públicas. La solución mantuvo SQL Server como sistema transaccional e incorporó Hyperledger Fabric como capa distribuida de verificabilidad, mediante canonicalización, hashes SHA-256, Transactional Outbox y procesamiento asincrónico con Kafka.

Respecto de la pregunta de investigación, ChainGov incorporó capacidad de verificación histórica en el escenario experimental, pero no satisfizo simultáneamente las condiciones establecidas en la hipótesis. En T1 se auditaron 500 casos con concordancia respecto del ground truth y se verificaron 3500 de 3500 eventos esperados. Sin embargo, C1 no pudo determinarse mediante una comparación T0/T1 funcionalmente equivalente y la degradación máxima de throughput alcanzó 80,66%, superando el límite preespecificado de 15%.

Los resultados, por tanto, no demuestran una superioridad general de blockchain frente a una base de datos relacional. Evidencian, para la implementación evaluada, un compromiso entre la incorporación de verificabilidad distribuida y su costo de rendimiento.

6.2 Evaluación de la hipótesis

H1 requería simultáneamente una mejora del tiempo de auditoría de al menos 40% y una degradación del rendimiento transaccional no superior al 15%.

La mejora del tiempo de auditoría no pudo determinarse válidamente: T0 no disponía de una fuente histórica independiente comparable con la utilizada por ChainGov, de modo que ambas condiciones no ofrecían capacidades funcionalmente equivalentes y cualquier porcentaje calculado habría carecido de validez de constructo. Utilizar una consulta al estado relacional actual, construir retrospectivamente un historial o asignar un tiempo artificial habría modificado la semántica del experimento. Esta indeterminación es una limitación metodológica del diseño comparativo original —no una falla de la implementación ni una omisión de la experimentación— y no evidencia un empeoramiento ni una mejora de ChainGov en esa dimensión.

C2 sí pudo evaluarse y presentó una degradación máxima de 80,66%, superior al límite de 15%. En consecuencia, H1 no quedó respaldada bajo los criterios preespecificados, dado que requería el cumplimiento simultáneo de ambas condiciones y C2 no se cumplió.

6.3 Cumplimiento de objetivos y contribuciones

El objetivo general se abordó mediante el diseño, implementación y validación experimental del MVP de ChainGov. Los objetivos específicos permitieron caracterizar el problema de verificabilidad, definir la separación entre persistencia operacional y evidencia distribuida, implementar los mecanismos de integración y evaluar el comportamiento de la solución.

La contribución arquitectónica consistió en separar los datos transaccionales mantenidos en SQL Server de la evidencia criptográfica registrada mediante Hyperledger Fabric, evitando utilizar blockchain como sustituto general de la base de datos. Como contribución técnica, el MVP integró canonicalización, SHA-256, Transactional Outbox, Kafka, Fabric, idempotencia y mecanismos de commit-reveal.
La contribución empírica fue cuantificar el comportamiento de esta implementación: se obtuvo concordancia con el ground truth en los 500 casos auditados y una completitud verificable de 100% en el conjunto experimental, junto con una degradación de throughput superior al umbral establecido. Finalmente, como contribución metodológica, el estudio evidenció la necesidad de separar la capacidad de verificación histórica del tiempo requerido para ejecutarla cuando se comparan arquitecturas con garantías funcionales diferentes.

6.4 Limitaciones e implicancias

El estudio se ejecutó sobre un MVP controlado, con datos y actores simulados y organizaciones de Fabric representadas lógicamente. Por ello, los resultados no incorporan todas las condiciones de red, operación y gobernanza propias de un despliegue interinstitucional.

Las pruebas de rendimiento utilizaron tres repeticiones oficiales por configuración y se analizaron descriptivamente. Además, T0 y T1 se compararon como condiciones agregadas, sin pruebas de ablación; en consecuencia, la degradación observada no puede atribuirse individualmente a Fabric, Kafka, Outbox u otro componente.

La ausencia en T0 de una capacidad histórica independiente funcionalmente equivalente limitó el contraste temporal C1. Asimismo, bajo carga sostenida la tasa de generación de eventos superó en más de dos órdenes de magnitud la capacidad de anclaje observada, por lo que los eventos pendientes se acumularon durante toda la medición. Una operación continua exige que la capacidad de anclaje iguale o supere la tasa de generación; mientras esa condición no se cumpla, el rezago crece sin cota y la evidencia verificable permanece diferida. En consecuencia, la configuración evaluada no resulta apta para un uso productivo continuo sin optimizar antes la sincronización con la capa distribuida y dimensionar su capacidad de anclaje a la carga esperada.

Por estas razones, los resultados caracterizan la arquitectura y configuración evaluadas y no deben generalizarse como una medida del rendimiento de Hyperledger Fabric ni como demostración de cumplimiento normativo o aptitud inmediata para producción.

6.5 Trabajo futuro

El trabajo futuro debería priorizar pruebas de ablación que separen el costo de canonicalización y hashing, persistencia en Outbox, middleware, Kafka y procesamiento asociado a Fabric, con el propósito de identificar los principales cuellos de botella de T1. En particular, se requiere medir la capacidad de anclaje y el rezago de consistencia eventual como variables explícitas, y dimensionar la capacidad de anclaje a la carga esperada mediante la agrupación de varios eventos por transacción, un mayor paralelismo de envío en el conector y la distribución de los componentes en equipos separados.

También se propone reformular la evaluación de auditoría en dos dimensiones: capacidad de verificación histórica (VD1a) y tiempo de auditoría entre mecanismos funcionalmente equivalentes (VD1b). Una evaluación futura de VD1b requeriría definir previamente una línea base que disponga de evidencia histórica independiente comparable. Una línea base candidata son las tablas ledger de SQL Server, que ofrecen un historial verificable dentro del propio motor y resultan funcionalmente más próximas a T1 que la consulta del estado vigente.

Finalmente, resulta pertinente ampliar las repeticiones y escenarios de carga, evaluar fallas y recuperación, distribuir físicamente las organizaciones y analizar configuraciones alternativas de privacidad, endorsement y gobernanza. Un eventual escenario productivo requeriría además evaluaciones específicas de seguridad, interoperabilidad, privacidad y requisitos jurídicos aplicables.

6.6 Cierre

ChainGov demostró la viabilidad técnica de complementar una base de datos relacional con una capa distribuida de evidencia verificable sin trasladar a blockchain toda la persistencia operacional. En el conjunto experimental, la arquitectura permitió detectar las alteraciones definidas y verificar la totalidad de los eventos esperados.

Al mismo tiempo, el costo de rendimiento observado superó el límite establecido en la hipótesis y el anclaje no alcanzó la tasa de generación de eventos; ambos aspectos deben resolverse antes de una operación continua. El aporte del trabajo consiste, por tanto, en aportar evidencia experimental sobre el compromiso entre verificabilidad y rendimiento de una arquitectura híbrida concreta, junto con una base metodológica para evaluar y optimizar futuras implementaciones.

Referencias

[1] Dirección de Compras y Contratación Pública, "Cuenta Pública Participativa 2026," 2026. [Online]. Disponible en: https://www.chilecompra.cl/wp-content/uploads/2026/07/2026-CuentaPublicaChileCompra.pdf. [Revisado el 23 9 2026].

[2] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco and J. Yellick, "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," in *13th EuroSys Conference (EuroSys '18)*, Porto, 2018.

[3] M. Belotti, N. Božić, G. Pujolle and S. Secci, "A Vademecum on Blockchain Technologies: When, Which, and How," *IEEE Communications Surveys & Tutorials,* vol. 21, no. 4, pp. 3796-3838, 2019.

[4] C. Zhu, J. Li, Z. Zhong, C. Yue and M. Zhang, "A Survey on the Integration of Blockchains and Databases," *Data Science and Engineering,* vol. 8, pp. 196-219, 2023.

[5] L. Allen, P. Antonopoulos, A. Arasu, J. Gehrke, J. Hammer, J. Hunter, R. Kaushik, D. Kossmann, J. Lee, R. Ramamurthy, S. Setty, J. Szymaszek, A. van Renen and R. Venkatesan, "Veritas: Shared Verifiable Databases and Tables in the Cloud," in *9th Biennial Conference on Innovative Data Systems Research (CIDR)*, 2019.

[6] K. Govindan, P. Jain, R. K. Singh and R. Mishra, "Blockchain Technology as a Strategic Weapon to Bring Procurement 4.0 Truly Alive: Literature Review and Future Research Agenda," *Transportation Research Part E: Logistics and Transportation Review,* vol. 181, p. 103352, 2024.

[7] D. A. Batista, "Enhancing Transparency and Accountability in Public Procurement: Exploring Blockchain Technology to Mitigate Records Fraud," *Records Management Journal,* vol. 34, no. 2-3, pp. 151-170, 2024.

[8] National Institute of Standards and Technology, "Secure Hash Standard (SHS)," NIST, Gaithersburg, 2015.

[9] Hyperledger Foundation, "Private Data," [Online]. Disponible en: https://hyperledger-fabric.readthedocs.io/en/release-2.5/private-data/private-data.html. [Revisado el 19 9 2026].

[10] Hyperledger Foundation, "The Ordering Service," [Online]. Disponible en: https://hyperledger-fabric.readthedocs.io/en/release-2.5/orderer/ordering_service.html. [Revisado el 19 9 2026].

[11] P. Antonopoulos, R. Kaushik, H. Kodavalla, S. Rosales Aceves, R. Wong, J. Anderson and J. Szymaszek, "SQL Ledger: Cryptographically Verifiable Data in Azure SQL Database," in *2021 International Conference on Management of Data (SIGMOD '21)*, Virtual Event, China, 2021.

[12] Microsoft, "Ledger Overview," [Online]. Disponible en: https://learn.microsoft.com/en-us/sql/relational-databases/security/ledger/ledger-overview?view=sql-server-ver16. [Revisado el 19 9 2026].

[13] C. Richardson, "Pattern: Transactional Outbox," [Online]. Disponible en: https://microservices.io/patterns/data/transactional-outbox. [Revisado el 19 9 2026].

[14] Z. Ge, D. Loghin, B. C. Ooi, P. Ruan and T. Wang, "Hybrid Blockchain Database Systems: Design and Performance," *Proceedings of the VLDB Endowment,* vol. 15, no. 5, pp. 1092-1104, 2022.

[15] M. El-Hindi, C. Binnig, A. Arasu, D. Kossmann and R. Ramamurthy, "BlockchainDB—A Shared Database on Blockchains," *Proceedings of the VLDB Endowment,* vol. 12, no. 11, pp. 1597-1609, 2019.

[16] Z. Hong, S. Guo, E. Zhou, W. Chen, H. Huang and A. Zomaya, "GriDB: Scaling Blockchain Database via Sharding and Off-Chain Cross-Shard Mechanism," *Proceedings of the VLDB Endowment,* vol. 16, no. 7, pp. 1685-1698, 2023.

[17] Q. Lin, B. Gu and F. Nawab, "RollStore: Hybrid Onchain-Offchain Data Indexing for Blockchain Applications," *IEEE Transactions on Knowledge and Data Engineering,* vol. 36, no. 12, pp. 9176-9191, 2024.

[18] G. P. Andrade, J. C. Andrade de Abreu and R. C. dos Santos, "The Impact of Blockchain on Brazilian Public Procurement Processes from the Perspective of Transaction Costs: Scenarios as Perceived by Experts," *International Journal of Organizational Analysis,* vol. 33, no. 2, pp. 365-389, 2025.

[19] S. Semenzin, D. Rozas and S. Hassan, "Blockchain-Based Application at a Governmental Level: Disruption or Illusion? The Case of Estonia," *Policy and Society,* vol. 41, no. 3, pp. 386-401, 2022.

Anexos

I. Configuración del entorno experimental

Este anexo resume los principales componentes y parámetros utilizados durante la implementación y evaluación experimental de ChainGov. Su propósito es complementar la descripción metodológica y facilitar la reproducibilidad del estudio.

Tabla 6. Componentes principales del entorno experimental

Componente	Configuración o función
Equipo de ejecución	Intel Core i7-7820X (8 núcleos, 16 hilos); 31 GiB de memoria asignados a WSL2; Windows con WSL2 (kernel 5.15.167.4)
Contenedores	Docker 29.7.2 y Docker Compose v5.5.1 (Docker Desktop)
Base de datos transaccional	Microsoft SQL Server 2022, compilación 16.0.4275.2 (imagen 2022-latest)
Middleware de integración	Worker .NET 8 (Confluent.Kafka 2.6.1; Microsoft.Data.SqlClient 5.2.2); lotes de hasta 100 eventos cada 500 ms
Sistema de mensajería	Apache Kafka 3.9.1, 6 particiones
Conector blockchain	Node.js 22 (@hyperledger/fabric-gateway ^1.8.0; kafkajs 2.2.4); hasta 6 particiones concurrentes y procesamiento secuencial por partición
Plataforma blockchain	Hyperledger Fabric 2.5 (peer, orderer y herramientas); Fabric CA 1.5; CouchDB 3.3.3
Chaincode	Go 1.24 (fabric-contract-api-go v2.2.1), desplegado como servicio (CCaaS)
API de verificación	Node.js 22 y Express 5; auditoría por lotes con concurrencia 8
Herramienta de carga	Apache JMeter 5.6.3 mediante JDBC (Microsoft JDBC Driver 13.4.0; OpenJDK 17.0.20); tiempo máximo de consulta de 60 s
Organizaciones Fabric	ChileCompraMSP, CGRMSP, OrganismoPilotoMSP y ObservadorCiudadanoMSP (simuladas), más OrdenamientoMSP
Peers	1 por organización
Servicio de ordenamiento	3 nodos Raft (CFT); BatchTimeout de 1 s; MaxMessageCount de 100
Canales	canal-publico (4 organizaciones) y canal-evaluacion (ChileCompra, CGR y Organismo Piloto)
Private Data Collection	colOfertasEconomicas, con miembros ChileCompraMSP y CGRMSP
Política de endorsement evaluada	AND('ChileCompraMSP.peer', 'CGRMSP.peer')
Función hash	SHA-256 sobre JSON canónico codificado en UTF-8
Integración SQL-mensajería	Transactional Outbox
Semántica de entrega	Al menos una vez, con efecto lógico idempotente

La arquitectura mantuvo SQL Server como sistema de persistencia operacional. En T1, la generación de la representación canónica del evento, el cálculo de su evidencia criptográfica y la persistencia en el Outbox formaron parte del procesamiento asociado a ChainGov. La publicación mediante Kafka y el procesamiento posterior en Hyperledger Fabric se realizaron asincrónicamente.

La red Fabric representó organizaciones institucionales diferenciadas de forma lógica dentro de un entorno experimental. Por tanto, esta configuración no correspondió a un despliegue físico entre instituciones públicas

ni incorporó las condiciones de red, operación y gobernanza de un ambiente productivo. Esta delimitación restringe la validez externa, pero permite identificar claramente el entorno sobre el cual se obtuvieron los resultados reportados.

II. Protocolo experimental de VD1

VD1 evaluó el mecanismo de auditoría de trazabilidad implementado en ChainGov. El universo experimental estuvo compuesto por 500 casos, distribuidos en cuatro clases previamente definidas. El ground truth se mantuvo separado del mecanismo auditor y fue utilizado posteriormente para contrastar la clasificación esperada con la obtenida por ChainGov.

Tabla 7. Distribución del ground truth de VD1

Clase	Alteración experimental	Casos
A0	Sin alteración	200
A1	Modificación directa de contenido	100
A2	Regresión de VersionAnclaje	100
A3	Eliminación de registro SQL	100
Total		500

La correspondencia entre las clases y las alteraciones está explícitamente definida en la instrumentación experimental conservada.

a. Secuencia experimental

El procedimiento comprendió una medición T1 previa a las alteraciones y una auditoría final posterior a la intervención controlada. Antes de efectuar la inyección, la instrumentación comprobaba que existieran exactamente 500 casos y que su distribución correspondiera a 200/100/100/100. También exigía que los registros existieran antes de la intervención y que no hubiera eventos experimentales pendientes de estabilización.

La secuencia utilizada puede resumirse de la siguiente manera:

1. Preparación del conjunto experimental de 500 casos y construcción independiente del ground truth.
2. Estabilización de la evidencia asociada a los casos antes de la intervención.
3. Ejecución de una auditoría T1 previa a las alteraciones.
4. Aplicación controlada de las alteraciones correspondientes a A1, A2 y A3, manteniendo A0 sin alteración.
5. Ejecución de la auditoría final T1.
6. Comparación de las clasificaciones producidas por el auditor con el ground truth.
7. Cálculo de la cantidad de eventos recuperables y verificables respecto del total esperado.
8. Conservación del tiempo total de ejecución de cada auditoría.

La instrumentación de las alteraciones fue diseñada para no producir nueva evidencia que ocultara la modificación experimental: el procedimiento especificó que no debía generar nuevos eventos en el Outbox, modificar la evidencia de ChainGov ni modificar Fabric.

b. Resultados consolidados de VD1

Tabla 8. Resultados de auditoría VD1 en T1

Medición	Casos	INTEGRA	ALTERADA	Eventos verificables	Completitud	Tiempo
Previa a alteraciones	500	500	0	3500/3500	100 %	3963,13 ms
Final	500	200	300	3500/3500	100 %	3963,57 ms

La auditoría final presentó concordancia con los 500 casos del ground truth. La diferencia de 0,44 ms entre las dos ejecuciones T1 solo caracteriza descriptivamente su estabilidad temporal y no constituye una comparación T0/T1.

c. Ejecución de la condición T0

La condición T0 se ejecutó mediante el procedimiento exp.usp_AuditoriaT0 sobre los mismos 500 códigos. El procedimiento consultó el estado vigente de mp.Licitacion y mp.Oferta, asignó el veredicto INEXISTENTE a los casos sin registro y NO_VERIFICABLE_HISTORICAMENTE a los restantes, y registró la duración del bloque SQL en el servidor.

Tabla 9. Resultado de la ejecución T0 de VD1

Indicador	Resultado
Identificador de ejecución	VD1_T0_R1
Casos auditados	500
Casos existentes (NO_VERIFICABLE_HISTORICAMENTE)	400
Casos inexistentes (INEXISTENTE)	100
Casos verificables históricamente	0
Duración del bloque SQL en el servidor	12,249 ms
Registro	Salida de consola del 14-09-2026, 00:42 (hora local)

Los 100 casos inexistentes coincidieron con la clase A3. T0 no dispuso de evidencia histórica para distinguir los casos A0, A1 y A2, y su duración no incluyó la recuperación de historial. Por esta razón, estos valores no deben utilizarse para calcular C1. El resultado metodológicamente válido permanece como C1 no determinado bajo una comparación funcionalmente equivalente.

III. Protocolo experimental de VD2

VD2 evaluó el impacto de la condición arquitectónica sobre el rendimiento transaccional utilizando Apache JMeter mediante acceso JDBC. Se compararon T0 y T1 bajo los mismos niveles preespecificados de concurrencia.

Tabla 10. Matriz de ejecución de VD2

Condición	Usuarios concurrentes	Repeticiones oficiales	Warm-up	Medición oficial
T0	10	3	5 min	30 min
T0	50	3	5 min	30 min
T0	100	3	5 min	30 min
T0	250	3	5 min	30 min
T0	500	3	5 min	30 min
T1	10	3	5 min	30 min
T1	50	3	5 min	30 min
T1	100	3	5 min	30 min
T1	250	3	5 min	30 min
T1	500	3	5 min	30 min
Total	5 niveles × 2 condiciones	30 corridas oficiales		

La configuración del sistema y de la carga fue congelada durante la ejecución para evitar cambios del protocolo en función de resultados parciales. La única modificación afectó al criterio de aceptación del warm-up, detallado al final de este anexo.

T0 y T1 ejecutaron la misma carga de negocio. La diferencia experimental correspondió a la intervención arquitectónica. En T1 se encontraba habilitado el procesamiento sincrónico necesario para generar la representación del evento, calcular la evidencia criptográfica y persistirla en el Outbox. El transporte posterior mediante Kafka y Fabric permaneció desacoplado del cierre de la operación de negocio.

Una muestra de JMeter representó un ciclo completo compuesto por 10 llamadas a procedimientos de negocio. En consecuencia, el throughput utilizado en el análisis se expresó como **Operaciones de negocio/s = ciclos/s × 10**. Las latencias P50, P95 y P99 correspondieron al tiempo de respuesta de un ciclo completo.

Las métricas registradas comprendieron throughput, latencias P50, P95 y P99 y errores. Los períodos de calentamiento no fueron considerados corridas oficiales.

El criterio C2 se evaluó conservadoramente sobre todos los niveles preespecificados de concurrencia, evitando seleccionar retrospectivamente una carga favorable:

$$C2: \max_{u \in U} \text{Degradación}(u) \leq 15\%, \qquad con\ U = \{10, 50, 100, 250, 500\}$$

La medida consolidada por condición y nivel fue la mediana de las tres repeticiones. Como análisis de sensibilidad se calculó también la media aritmética prevista en el protocolo congelado.

Previamente a las corridas oficiales se ejecutó una prueba piloto de 60 s por nivel y condición, sin warm-up, con el mismo plan de JMeter. Sus diez ejecuciones finalizaron sin errores y sus valores no se incorporaron a las métricas oficiales.

Inicialmente, el procedimiento de ejecución exigía cero errores durante los 300 s de warm-up. Con ese criterio se aceptaron dos corridas oficiales, VD2-T1-U10-R1 y VD2-T0-U500-R1, ambas sin errores, y se abortaron tres intentos. A partir de la corrida VD2-T1-U250-R1, iniciada el 16-09-2026 a las 03:55 UTC, el criterio exigió cero errores en los últimos 120 s del warm-up. Todos los errores observados correspondieron a excepciones de tiempo de espera de consulta de 60 s (HY008), en solicitudes iniciadas entre 5 y 10 s después del comienzo del warm-up, durante la rampa de 60 s. Ninguna corrida oficial registró errores en su ventana de medición de 30 min.

Tabla 11. Errores registrados durante el warm-up de VD2

Ejecución	Estado	Muestras de warm-up	Errores	Errores en los últimos 120 s
VD2-T0-U500-R1 (intento 1)	Abortada	141545	4	0
VD2-T1-U250-R1 (intento 1)	Abortada	40855	6	0
VD2-T1-U250-R1 (intento 2)	Abortada	39694	6	0
VD2-T1-U250-R1	Oficial	40171	4	0
VD2-T1-U250-R2	Oficial	32835	6	0
VD2-T1-U250-R3	Oficial	39422	4	0
VD2-T1-U500-R1	Oficial	33579	65	0
VD2-T1-U500-R2	Oficial	35338	56	0
VD2-T1-U500-R3	Oficial	35238	53	0
Otras 24 corridas oficiales	Oficial	—	0	0

IV. Evidencia experimental consolidada

Este anexo presenta en forma compacta la evidencia cuantitativa final utilizada para la evaluación de VD1, VD2 y VD3. Se incluyen exclusivamente resultados pertenecientes al protocolo final; los períodos de calentamiento y las ejecuciones preliminares o interrumpidas no formaron parte de las métricas oficiales.

a. Auditoría y completitud verificable

Tabla 12. Resultados consolidados de VD1 y VD3

Indicador	Resultado
Casos auditados	500
Casos INTEGRA	200
Casos ALTERADA	300
Concordancia con ground truth	500/500
Eventos esperados	3500
Eventos verificables	3500
Completitud verificable VD3	100 %
Tiempo auditoría T1 final	3963,57 ms
Casos verificables históricamente en T0	0 de 500
C1	No determinado bajo equivalencia funcional

El valor de VD3 corresponde exclusivamente al conjunto experimental controlado y no constituye una garantía de completitud para otros despliegues, cargas o escenarios de falla.

b. Rendimiento transaccional

Tabla 13. Resultados consolidados de VD2

Usuarios	T0 (op/s)	T1 (op/s)	Degradación	P95 T0 (ms por ciclo)	P95 T1 (ms por ciclo)
10	3181,73	1056,84	66,78 %	41	187
50	5238,64	1207,95	76,94 %	133	817
100	5922,31	1145,45	80,66 %	237	1614
250	5386,43	1195,52	77,81 %	628	2718
500	5170,59	1072,91	79,25 %	1182	5808

Las 30 corridas oficiales finalizaron sin errores durante sus ventanas de medición. La máxima degradación se observó con 100 usuarios concurrentes y correspondió a 80,66 %, superior al límite preespecificado de 15 %. En consecuencia, C2 no se cumplió.

Estos valores representan el efecto agregado de la condición T1. Debido a que el protocolo no incorporó pruebas de ablación, la diferencia entre T0 y T1 no permite atribuir individualmente la degradación a Fabric, Kafka, Transactional Outbox, canonicalización, SHA-256 u otro componente específico.

Las Tablas 14 y 15 presentan los valores de cada una de las 30 corridas oficiales, a partir de los cuales se calcularon las medidas consolidadas.

Tabla 14. Throughput por corrida oficial de VD2 (op/s)

Condición	Usuarios	R1	R2	R3	Mediana	Media
T0	10	3381,01	3039,05	3181,73	3181,73	3200,60
T0	50	4966,79	5238,64	5348,31	5238,64	5184,58
T0	100	5927,16	5676,19	5922,31	5922,31	5841,88
T0	250	5386,43	5488,38	5166,44	5386,43	5347,08
T0	500	5241,02	4671,76	5170,59	5170,59	5027,79
T1	10	1368,85	1056,84	755,90	1056,84	1060,53
T1	50	1238,01	896,45	1207,95	1207,95	1114,14
T1	100	1183,35	1070,15	1145,45	1145,45	1132,98
T1	250	1281,70	1095,81	1195,52	1195,52	1191,01
T1	500	967,33	1072,91	1105,01	1072,91	1048,41

Tabla 15. Latencia P95 por ciclo en cada corrida oficial de VD2 (ms)

Condición	Usuarios	R1	R2	R3	Mediana
T0	10	40	46	41	41
T0	50	135	133	123	133
T0	100	237	245	224	237
T0	250	628	589	646	628
T0	500	1182	1391	1182	1182
T1	10	123	187	333	187
T1	50	671	1647	817	817
T1	100	1303	2045	1614	1614

Condición	Usuarios	R1	R2	R3	Mediana
T1	250	2413	3599	2718	2718
T1	500	7807,9	5808	5406	5808

c. Procesamiento asincrónico en T1

Las instantáneas del estado del Outbox registradas al término del warm-up y al cierre de cada corrida T1, junto con el registro del conector, permitieron cuantificar el procesamiento asincrónico durante las ventanas de medición. Las tasas de eventos generados y publicados en Kafka se calcularon a partir de las instantáneas SQL sobre 1800 s; la tasa de anclaje, a partir de los eventos confirmados por el conector dentro de cada ventana. En las corridas T0, con los triggers del Outbox deshabilitados, no se generaron eventos.

Tabla 16. Procesamiento asincrónico durante las corridas oficiales T1

Usuarios	R	Generados (eventos/s)	Publicados en Kafka (eventos/s)	Anclados (eventos/s)	Pendientes de publicación al cierre	Anclados / total (%)
10	R1	2053,3	109,7	5,04	4.070.350	0,246%
10	R2	1585,3	103,3	4,75	3.243.095	0,288%
10	R3	1133,9	100,3	4,47	2.379.720	0,359%
50	R1	1857,3	96,4	4,41	3.720.850	0,239%
50	R2	1346,7	85,3	3,72	2.946.033	0,252%
50	R3	1812,4	94,9	4,44	3.763.575	0,239%
100	R1	1775,4	96,8	4,60	3.631.020	0,255%
100	R2	1605,5	90,4	4,06	3.323.655	0,250%
100	R3	1718,5	94,8	4,50	3.520.567	0,258%
250	R1	1923,8	103,7	4,87	3.842.746	0,254%
250	R2	1644,8	93,2	4,54	3.255.925	0,259%
250	R3	1794,2	100,6	4,72	3.603.479	0,263%
500	R1	1454,4	84,1	3,76	2.936.132	0,264%
500	R2	1611,9	100,7	4,59	3.215.407	0,286%
500	R3	1660,4	104,1	4,87	3.295.475	0,294%

d. Síntesis de criterios experimentales

Tabla 17. Trazabilidad entre variables, evidencia y resultado

Variable/criterio	Evidencia disponible	Resultado
VD1	Auditoría de 500 casos T1 y ground truth independiente	500/500 concordantes
C1	Requería comparación temporal T0/T1 funcionalmente equivalente	No determinado
VD3	3500 eventos esperados y 3500 verificables	100 %
VD2	30 corridas oficiales T0/T1	T1 inferior a T0 en todos los niveles
C2	Máxima degradación ≤15 %	80,66 % - No cumple
H1	Cumplimiento simultáneo de C1 y C2	No respaldada
Anclaje asincrónico (no preespecificado)	Estado del Outbox y registro del conector en 15 corridas T1	3,7 - 5,0 eventos/s anclados frente a 1134 - 2053 generados

Esta síntesis mantiene la separación entre capacidad de verificación, tiempo de auditoría y rendimiento transaccional. En particular, la ausencia de una comparación temporal equivalente para C1 no implica que VD1 no haya sido ejecutada, y el incumplimiento observable de C2 fue suficiente para que H1 no quedara respaldada bajo su regla conjuntiva.

V. Validación funcional de extremo a extremo

Previo a la ejecución de VD1, se ejecutó el script de demostración del MVP sobre el entorno completo: SQL Server, worker, Kafka, conector, red Fabric y API de verificación. La Tabla 18 resume los pasos y los resultados observados. Esta validación fue funcional y no forma parte de las mediciones de VD1, VD2 ni VD3. Los registros del worker, del conector y de la API, el estado final del Outbox y el resultado de la verificación posterior a la modificación directa se conservaron con un manifiesto SHA-256.

Tabla 18. Pasos y resultados de la validación funcional de extremo a extremo

Paso	Acción	Resultado observado
1	Simulación de 5 licitaciones completas con 3 ofertas cada una	5 licitaciones en estado CONTRATO_FIRMADO
2	Sincronización SQL Server → Outbox → Kafka → Fabric	35 eventos del canal público y 40 del canal de evaluación en estado ANCLADO
3	Verificación de la licitación SIM-5-LE26	Veredicto INTEGRA: hash vigente en SQL Server igual al hash anclado; inclusión comprobada mediante qscc en canal-publico, bloque 26, código de validación 0 (VALID)
4	Auditoría automatizada de las 5 licitaciones	5 casos INTEGRA; 35 de 35 eventos verificables (100%); 134,92 ms
5	Modificación directa de SIM-5-LE26 en SQL Server, sin pasar por el Outbox	Veredicto ALTERADA: el documento vigente difiere del hash anclado en la secuencia 7
6	Alteración de una oferta sellada antes de la apertura	Veredicto ALTERADA; revelación DISCREPANTE respecto del compromiso sellado