SafeTraffic Ledger: Inmutabilidad y Transparencia Vial con Blockchain (Stacks/Web3)

Autores: Mateo Vargas, David Zema, Fabio García Meza

Usuarios de GitHub: MateoDVE, fabio-gmoo, David-ZemaP  
  
proyecto Deployado: [Transacción STX - 0x4da65... bdffb](https://explorer.hiro.so/txid/0x4da65d11ef479beb9859497caffeb15b765fb6b6a49e54d5be125139864bdffb?chain=mainnet)

|  |  |
| --- | --- |
| **Dirección del contrato** | [**SP33WHCRVBZ237GADZ2854SYQ20BJNVCPSNKSSW9X.inicedent-register**](https://explorer.hiro.so/txid/SP33WHCRVBZ237GADZ2854SYQ20BJNVCPSNKSSW9X.inicedent-register?chain=mainnet) |
| **ID de transacción** | [**0x4da65d11ef479beb9859497caffeb15b765fb6b6a49e54d5be125139864bdffb**](https://explorer.hiro.so/address/0x4da65d11ef479beb9859497caffeb15b765fb6b6a49e54d5be125139864bdffb?chain=mainnet) |
| **De** | [**SP33WHCRVBZ237GADZ2854SYQ20BJNVCPSNKSSW9X**](https://explorer.hiro.so/address/SP33WHCRVBZ237GADZ2854SYQ20BJNVCPSNKSSW9X?chain=mainnet) |

Institución: Universidad Católica Boliviana

Fecha: 26 de octubre de 2025

# Resumen

El presente informe describe el diseño, justificación y plan de implementación de SafeTraffic Ledger, una plataforma descentralizada para registrar multas, accidentes e incidentes viales de manera inmutable y auditable utilizando Stacks/Blockchain y principios de Web3. El documento presenta el problema de falta de transparencia y corrupción en registros viales en América Latina, los objetivos y preguntas de validación, los requisitos funcionales y no funcionales, la arquitectura técnica on-chain y off-chain, el modelo de datos y contratos inteligentes, consideraciones éticas y legales, el plan de prototipo (Figma + React) y MVP, así como la evaluación, métricas y riesgos y la hoja de ruta. Se adhiere en lo posible a lineamientos de formato APA para estructura, citas y referencias.

Palabras clave: blockchain, Web3, Stacks, transparencia, tránsito, gobernanza, gobierno abierto, antifraude, evidencia digital.

# Introducción

## Contexto y problema

Los sistemas de registro de infracciones y siniestros viales en diversos países de América Latina enfrentan bajos niveles de digitalización, opacidad y riesgo de manipulación. Esto deriva en impunidad para conductores reincidentes, desconfianza ciudadana y pérdida fiscal. La ausencia de trazabilidad dificulta el diseño de políticas públicas basadas en evidencia.

## Justificación

El uso de blockchain en Stacks ofrece inmutabilidad, sellado temporal y auditoría pública de eventos, mientras que Web3 habilita aplicaciones descentralizadas con wallets para autenticación y firma, reduciendo dependencias de autoridades únicas. Esto eleva el costo de la corrupción y crea un rastro probatorio para sanción administrativa y legal.

## Objetivo general

Diseñar y validar un prototipo funcional de registro y verificación de incidentes viales que aplique commit–reveal con hash, feed público y gobernanza on-chain (roles y disputas), demostrando viabilidad técnica y valor público.

## Objetivos específicos

1. Definir requisitos funcionales y no funcionales del sistema.
2. Especificar arquitectura on-chain y off-chain, modelo de datos y contratos.
3. Diseñar prototipo de Figma y componentes React reutilizables.
4. Implementar un MVP en Stacks Testnet con IPFS/Gaia.
5. Establecer métricas de impacto y plan de evaluación.
6. Identificar riesgos y mitigaciones técnicas y organizacionales.

## Preguntas de validación

¿El commit inmediato de hash reduce la ventana de extorsión “págame o lo subo”? ¿El feed público incrementa la percepción de transparencia y el monitoreo ciudadano? ¿La gobernanza on-chain mejora la rendición de cuentas? ¿El MVP es usable para agentes y auditores y comprensible para ciudadanos?

## Alcance

Incluye registro de incidentes (commit), publicación y revelación (reveal), feed público, denuncias cifradas y gestión básica de roles y disputas. Excluye en esta fase los pagos onchain de multas, integración con aseguradoras o semáforos y analítica avanzada.

# Marco conceptual

La tecnología blockchain es un libro mayor distribuido que permite el registro de transacciones de forma cronológica, inmutable y auditable. Este tipo de infraestructura elimina la dependencia de intermediarios centralizados y ofrece mayor transparencia sobre quién hizo qué y cuándo lo hizo (Dziundziuk, 2025). En el contexto de una plataforma para gestión vial, esta propiedad tiene relevancia porque los registros de infracciones o accidentes suelen depender de entidades que pueden sufrir manipulación o corrupción. Con blockchain cada evento queda sellado con un timestamp y un hash que impide su modificación retroactiva, generando un rastro de auditoría confiable.

El ecosistema Web3 complementa esta infraestructura con capas de interacción, identidad digital y aplicaciones descentralizadas (dApps). En Web3 los usuarios controlan sus propias wallets, firman transacciones y participan en contratos inteligentes sin depender completamente de una autoridad centralizada. Una plataforma como SafeTraffic Ledger conecta conceptos de identidad, evidencia digital, firma de transacciones y registros verificables, fortaleciendo mecanismos de transparencia (Huubse, 2025).

Dentro de los esquemas técnicos que refuerzan la integridad de los datos se encuentra el modelo commit–reveal. Este consiste en publicar primero un compromiso (commit), típicamente un hash de los datos, y luego revelar los datos originales junto con la sal que permite demostrar su coincidencia con el hash inicial. Se usa para mitigar ataques de manipulación de tiempos y cambios en entornos de blockchain donde las transacciones son públicas (Canidio & Danos, 2023). En este proyecto, el uso de commit–reveal permite sellar inmediatamente un rastro inmutable y luego revelar detalles, reduciendo la ventana de vulnerabilidad.

La gobernanza on-chain es otro componente conceptual. No basta con registrar datos; es necesario definir quién puede proponer registros, quién valida, quién abre disputas y cómo se resuelven. La literatura indica que los modelos de gobernanza blockchain deben articular derechos de decisión, incentivos, rendición de cuentas y transparencia (Liu, Lu, Yu, Paik & Zhu, 2021). En el contexto de la gestión vial en América Latina, introducir roles definidos y procesos de disputa visibles puede aumentar la confianza ciudadana y la integridad institucional (Ibrahimy, 2024). Sin embargo, se advierte que estos sistemas presentan desafíos: concentración de poder, falta de participación genuina y dinámicas de voto que no siempre favorecen la transparencia (Feichtinger, Fritsch, Vonlanthen & Wattenhofer, 2023).

Además, la literatura sobre blockchain en gobierno abierto sugiere que la simple adopción de la tecnología no garantiza resultados positivos. Un informe del World Economic Forum concluye que la blockchain puede servir como catalizador de transparencia, pero debe acompañarse de procesos claros, gobernanza y cultura de apertura (World Economic Forum, 2018). En el caso de SafeTraffic Ledger, esto se traduce en compromisos institucionales, auditoría ciudadana, acceso público y sanciones para prácticas corruptas.

En conclusión, el marco conceptual de SafeTraffic Ledger se basa en cuatro ideas clave: la inmutabilidad y auditabilidad de blockchain, la autonomía de interacción de Web3, el modelo commit–reveal para asegurar compromiso y revelación posterior, y la gobernanza on-chain para regular roles y procesos. Esta combinación técnica y organizativa busca mitigar las causas de manipulación de registros viales: falta de trazabilidad, discrecionalidad y opacidad.

Referencias

Canidio, A., & Danos, V. (2023). Commit–Reveal schemes against front‑running attacks. In Tokenomics 2022 (pp. 7:1–7:5). Schloss Dagstuhl – Leibniz‑Zentrum für Informatik. https://doi.org/10.4230/OASIcs.Tokenomics.2022.7

Dziundziuk, B. (2025). Methodology for integrating blockchain into digital governance systems: Focus on effective implementation in the public sector. Science & Public Policy. https://doi.org/10.1093/scipol/scaf022/8127872

Feichtinger, R., Fritsch, R., Vonlanthen, Y., & Wattenhofer, R. (2023). The hidden shortcomings of (D)AOs — An empirical study of on‑chain governance. arXiv. https://arxiv.org/abs/2302.12125

Huubse, M. (2025). Blockchain strategies to enhance transparency, efficiency and accountability in governance. Walden University Dissertations.

Ibrahimy, M. M. (2024). Blockchain‑based governance models supporting corruption transparency: A review of best practices. Journal of Governance and Innovation, X(X), 1‑12.

Liu, Y., Lu, Q., Yu, G., Paik, H.‑Y., & Zhu, L. (2021). Defining blockchain governance principles: A comprehensive framework. arXiv. https://arxiv.org/abs/2110.13374

Stacks Foundation. (2024). What is Stacks? https://docs.stacks.co/concepts/stacks101/what-is-stacks

World Economic Forum. (2018). Blockchain Technology for Government Transparency.

WEF. https://www3.weforum.org/docs/WEF\_Blockchain\_Government\_Transparency\_Report.p df

# Metodología

El enfoque se basa en investigación aplicada y diseño-construcción. Se trabajará con iteraciones cortas, centradas en el diseño del prototipo sin retroalimentación de usuarios aún, dado que el objetivo es construir un MVP técnico. Se emplearán entrevistas y revisión de normativa para ajustar la arquitectura conceptual y se definirán criterios de aceptación funcionales, de seguridad, usabilidad, desempeño y auditabilidad.

# Viabilidad Técnica y Alineación con el Ecosistema Stacks

La arquitectura de SafeTraffic Ledger se ha diseñado para maximizar la seguridad, la inmutabilidad y la transparencia, aprovechando las ventajas clave que ofrece **Stacks como capa de contratos inteligentes de Bitcoin (L2)**. La viabilidad técnica no solo se confirma por el diseño de la solución, sino por la integración directa de los componentes principales del ecosistema Stacks.

| **Componente Clave** | **Solución en Stacks** | **Justificación Técnica y Viabilidad** |
| --- | --- | --- |
| **Inmutabilidad de los Registros** | **Anclaje a Bitcoin (Proof-of-Transfer)** | Cada commit\_incident se sella en Stacks y, a través del mecanismo PoX, se ancla y finaliza en la cadena de Bitcoin. Esto dota a cada registro de la **seguridad inigualable de Bitcoin**, haciendo que el evento vial (el hash) sea inmutable y auditable a largo plazo, una característica no disponible en bases de datos centralizadas. |
| **Lógica de Negocio y Gobernanza** | **Contratos Inteligentes en Clarity** | Clarity es el lenguaje de *smart contracts* de Stacks, diseñado para ser **decidible** y **seguro**, ideal para gestionar la lógica crítica de la aplicación, como la máquina de estados del incidente (pending a revealed), la gestión de roles (proposer, auditor) y la resolución de disputas. Esto garantiza que la lógica de gobernanza on-chain sea predecible. |
| **Identidad y Autenticación** | **Wallets Stacks y Stacks Connect** | La autenticación del agente y el auditor se realiza mediante su **wallet Web3** (ej. Hiro Wallet) y Stacks Connect1. Esto elimina la necesidad de un sistema centralizado de *login* y provee **no-repudiación**, ya que cada acción (commit, reveal, open\_dispute) se firma criptográficamente por la entidad responsable. |
| **Escalabilidad y Almacenamiento** | **IPFS / Gaia (Off-Chain Storage)** | El modelo **commit-reveal** asegura que solo se almacena un *hash* de 32 bytes del archivo voluminoso (foto/video) on-chain2. El archivo real se sube a IPFS o Gaia3. Esto mantiene la cadena Stacks ligera, optimiza el costo de las transacciones (gas) y garantiza la escalabilidad del sistema sin comprometer la integridad de la evidencia. |

# Modelo de Negocio, Monetización y Rentabilidad de la Aplicación

El éxito de *SafeTraffic Ledger* se basa en un modelo **B2G (Business-to-Government)**, donde el producto se vende como una **Infraestructura de Transparencia Digital y Antifraude** a entidades gubernamentales.

## Modelo de Negocio: Valor Antifraude y Eficiencia Fiscal

La propuesta de valor se centra en el **retorno de la inversión (ROI)** para las agencias gubernamentales, mitigando las pérdidas por corrupción y opacidad.

**Clientes Objetivo:** Agencias de Tránsito Nacionales o Municipales y entes de control (Fiscalías, Contralorías) en países con alta percepción de corrupción vial.

**Problema Resuelto (ROI):** El sistema **elimina la ventana de extorsión** al obligar al agente a hacer el commit inmediato del incidente. La evidencia inmutable **aumenta la tasa de éxito en el cobro de multas**, ya que el ciudadano no puede disputar la validez del registro de forma sencilla.

**Valor para el Gobierno:** **Reducción de pérdidas fiscales** (por corrupción y multas no cobradas) y **mejora de la imagen institucional** por la transparencia pública.

**Valor para el Ciudadano:** Acceso a un *feed* de incidentes transparente y un **canal de denuncia anónima cifrada** para reportar coimas sin temor a represalias.

## Modelo de Monetización: Estrategia Híbrida SaaS B2G y Web3

La monetización combina la estabilidad de los contratos institucionales con los incentivos de la red Stacks.

**Licencia Institucional (SaaS B2G - Principal)**:

**Mecanismo:** La agencia de tránsito paga una **suscripción anual o multianual (en Fiat o STX)** por el derecho de uso de la plataforma, el soporte técnico, el alojamiento del indexador de eventos y la customización a las leyes locales.

**Flujo de Ingreso:** La fuente principal de ingresos para el equipo desarrollador.

**Tarifa de Transacción Subsidiada (Web3 Recurrente)**:

**Mecanismo:** Cada commit\_incident y reveal\_incident requiere una tarifa de gas en **STX**.

**Flujo de Ingreso:** La agencia de tránsito subsidia estas tarifas de gas para sus agentes. Esto genera un **flujo de STX** para los *miners* de Stacks y asegura la descentralización. A largo plazo, se puede cobrar una pequeña comisión sobre el gas por el uso del contrato de SafeTraffic Ledger.

**Servicios de Integración y Módulos Premium (Valor Añadido)**:

**Integración:** Cobro único por la integración con sistemas Legacy (bases de datos de licencias, registro vehicular).

**Módulos Premium:** Venta de módulos avanzados (ej. integración con **sBTC** para el pago on-chain de multas o **Analítica de Datos Viales** para planificación urbana).

## Rentabilidad y Escalabilidad

El modelo es altamente escalable y rentable debido a su naturaleza de infraestructura digital.

**Bajo Costo Marginal:** Una vez que el contrato y el *front-end* están desarrollados, el costo de adquirir un segundo, tercer y cuarto municipio/país es **mínimo**. El código fuente de Clarity se reutiliza, lo que reduce el **Costo de Adquisición de Cliente (CAC)**.

**Punto de Equilibrio (Break-Even):** Se alcanza con la venta de las primeras **pocas licencias institucionales** grandes, ya que estas cubrirán los costos de desarrollo y mantenimiento, que son los más altos en la fase inicial.

**Escalabilidad Geográfica:** El modelo es perfectamente replicable en toda **América Latina (LATAM)**, cumpliendo con el enfoque del hackathon, ya que los problemas de corrupción y opacidad son sistémicos en la región.

**Mitigación de Riesgos:** La dependencia de la infraestructura descentralizada de Stacks minimiza los costos de *hosting* de la base de datos central (solo se aloja el indexador y el *front-end*), lo que hace que los costos operativos sean predecibles y bajos.

# Requisitos del sistema

Los requisitos funcionales incluyen la posibilidad de registrar incidentes con commit, revelar datos, mantener un feed público de estados, permitir denuncias cifradas y gestionar roles básicos. Los requisitos no funcionales contemplan seguridad, privacidad, escalabilidad, disponibilidad, interoperabilidad, accesibilidad y trazabilidad.

# Arquitectura de solución

La arquitectura integra un contrato inteligente en Clarity en la red Stacks para commit, reveal y gestión de roles, un almacenamiento off-chain en IPFS o Gaia para evidencia y denuncias cifradas, un front-end en React y Next.js con Stacks Connect para autenticación por wallet y un indexador de eventos para mantener el feed. La interacción sigue un flujo donde se registra el hash del incidente, se revela con el CID y se actualiza el estado visible públicamente.

## Diagrama de secuencia textual de commit y reveal

Actor usuario con wallet abre la dApp, selecciona un archivo y completa metadatos mínimos, el cliente calcula evidenceHash y metaHash, el cliente envía commit\_incident al contrato, el contrato persiste el incidente con estado pending y emite evento Committed, el indexador capta el evento y actualiza la cache del feed, el usuario sube el archivo a IPFS o Gaia y obtiene el cid, el usuario ejecuta reveal\_incident con el cid y las pruebas de integridad, el contrato verifica campos y cambia a estado revealed y emite evento Revealed, el indexador actualiza el feed y el front muestra verificación de hash contra cid

## Flujo de denuncia cifrada y anclaje

El usuario completa un formulario de denuncia y adjunta pruebas, el cliente cifra el contenido con la clave pública del auditor, el cliente publica el archivo cifrado en IPFS y obtiene un cid, el cliente calcula whistleHash y llama add\_whistle con el hash o el cid, el contrato registra el índice on-chain y emite evento Whistle, el indexador notifica a auditores fuera de cadena mediante un canal seguro, el auditor recupera y descifra el contenido con su clave privada

## Flujo de disputa y resolución

El auditor abre una disputa con open\_dispute indicando el id del incidente y un reasonHash, el contrato cambia el estado a disputed y emite evento Disputed, el auditor analiza evidencia y resuelve con resolve\_dispute y un nuevo estado permitido resolved o stale, el contrato emite evento Resolved y el front refleja el cierre

## Módulos de software y responsabilidades técnicas

Contrato inteligente maneja estado mínimo, validaciones y eventos, indexador escucha eventos on-chain y mantiene una vista materializada para consultas y filtros, servicio de cifrado y relayer ofrece envío opcional de transacciones preservando anonimato del denunciante, front-end implementa autenticación por wallet, formularios, verificación de hashes y visualización del feed

# Modelo de datos

Los datos on-chain se organizan en un mapa con campos como id, proposer, evidencehash, meta-hash, geo-hash, status, commit-height y reveal-cid. Off-chain se conserva metadata minimalista y las denuncias cifradas que se anclan a través de su hash.

## Esquemas JSON de referencia

Metadatos del incidente

{ “ts”: “2025-10-22T13:45:03Z”, “lat”: -17.392, “lng”: -66.158, “type”: “accident”, “plate”:

“ABC-123”, “agentId”: “AG-045”, “notes”: “Colisión leve sin heridos” }

Evento de feed materializado en el indexador

{ “id”: 42, “status”: “revealed”, “committedAt”: 168000123, “revealedAt”: 168000555,

“evidenceHash”: “0x…”, “metaHash”: “0x…”, “cid”: “ipfs://bafy…”, “geoHash”: “0x…” }

Denuncia cifrada

{ “cid”: “ipfs://bafy…”, “whistleHash”: “0x…”, “auditorPubKey”: “base64…” }

## Máquina de estados del incidente

pending indica que existe un compromiso de evidencia, revealed indica que la evidencia fue revelada y vinculada a un cid, disputed indica que un auditor abrió una disputa, resolved indica cierre con validez o rechazo según política, stale indica que venció el plazo de revelación sin acción y se activan alertas

Transiciones permitidas pending a revealed, pending a stale, revealed a disputed, disputed a resolved

# Especificación de contratos inteligentes

Las funciones clave incluyen commit\_incident, reveal\_incident, open\_dispute, resolve\_dispute, add\_whistle y set\_param. Se emiten eventos para cada cambio de estado y las reglas de negocio establecen plazos para revelar, validación de roles y verificación de integridad.

## Interfaz del contrato y tipos

commit\_incident recibe evidence-hash buff32, meta-hash buff32, geo-hash buff32 y type uint y devuelve id uint, reveal\_incident recibe id uint, cid string, meta-hash buff32, salt buff32 y evidence-hash-check buff32 y devuelve bool, open\_dispute recibe id uint y reason-hash buff32 y devuelve bool, resolve\_dispute recibe id uint y new-status uint y devuelve bool, add\_whistle recibe id uint y whash buff32 y devuelve bool, set\_param recibe reveal-deadline uint

## Eventos esperados en el indexador

Committed con id y proposer, Revealed con id y cid, Disputed con id, Resolved con id y new-status, Whistle con id

## Parámetros y límites

El plazo de revelación se maneja en minutos, el tamaño máximo de cid se restringe para caber en string utf8 del contrato, las funciones públicas validan existencia de id y coherencia de hashes, la verificación criptográfica del archivo se realiza en el cliente y puede ser revalidada por el indexador fuera de cadena

# Diseño de interfaz y front-end

El prototipo en Figma contempla pantallas de inicio y feed, registro de incidentes, revelación, denuncias, detalle de incidentes y panel de auditor. En React se implementarán componentes como IncidentCard, CommitForm, RevealForm,

WhistleForm, StatusBadge y HashVerifier. La interfaz seguirá estándares de accesibilidad y usabilidad.

## Jerarquía de componentes y rutas

App envuelve autenticación y proveedor de Stacks, rutas principales home, incident new, incident reveal, incident detail y auditor panel, componentes compartidos para formularios y badges de estado, hook useEventsFeed que consume la vista materializada del indexador y hook useContractCall para invocar funciones del contrato

## API del indexador y del relay

GET /feed devuelve lista paginada de eventos con filtros por estado y fecha, GET /incidents/:id devuelve detalle materializado, POST /relay/commit recibe evidenceHash metaHash geoHash type y firma o reenvía a la red, POST /relay/reveal recibe id cid metaHash salt evidenceHashCheck, POST /relay/whistle recibe whistleHash o cid cifrado, todas las respuestas incluyen id y txId cuando aplica

Ejemplo de respuesta

{ “ok”: true, “id”: 42, “txId”: “0xabc…” }

# Seguridad, privacidad y ética

Se aplicará minimización de datos, cifrado extremo a extremo para denuncias, cumplimiento legal en protección de datos y medidas contra amenazas como doxxing o spam. Las mitigaciones incluyen límites de tasa, pruebas de humanidad ligeras y listas de roles verificadas. Se evitará almacenar información personal sensible on-chain, se ofrecerá relayer para preservar el anonimato del denunciante y se publicarán políticas claras de retención y acceso a la evidencia off-chain.

## Modelo de amenazas resumido

Suplantación de agente mitigada con autenticación por wallet y listas de roles, extorsión en sitio mitigada por commit inmediato y plazos de revelación con alertas, manipulación de archivos mitigada con hashes y verificación contra cid, denegación de servicio mitigada por límites de tasa y colas, fuga de datos mitigada por cifrado de denuncias y minimización de metadatos

# Evaluación y métricas

Se evaluará la tasa de éxito de transacciones, latencia, disponibilidad del feed, tiempo por tarea y error rate. El impacto se medirá en porcentaje de commits que llegan a reveal, tiempo promedio, denuncias procesadas y percepción de transparencia en pruebas piloto.

## Plan de pruebas

Pruebas unitarias de contrato con claridad y herramientas de testnet, pruebas de integración del indexador escuchando eventos de una red local y de test, pruebas de extremo a extremo en la dApp simulando un flujo completo de commit y reveal, pruebas de regresión sobre la máquina de estados, pruebas de seguridad sobre validación de entradas y verificación de hashes

# Riesgos y mitigaciones

Los riesgos principales incluyen extorsión en sitio, abuso de privilegios, spam y desgaste de la experiencia de usuario. Las mitigaciones incluyen commit inmediato con deadlines, roles on-chain, controles de validación y flujos guiados.

# Plan de trabajo

En la primera semana se abordará la definición de requisitos, contrato base y wireframes. En la segunda se implementará la dApp y el indexador. En la tercera se incorporarán denuncias cifradas y pruebas de usabilidad. En la cuarta se realizará el hardening, el guion de demo y el pitch. Los entregables serán el contrato en testnet, el prototipo en Figma, la dApp React operativa y la documentación.

## Plan de despliegue y entornos

Despliegue del contrato en Stacks testnet con un proceso reproducible y versionado, despliegue del front en un hosting estático y del indexador en un contenedor con logs centralizados, configuración de un servicio de pinning IPFS o Gaia con políticas de retención, registro de variables de entorno y claves en un gestor seguro

## Observabilidad y monitoreo

Recolección de métricas de transacciones, colas del indexador, tiempos de respuesta del feed y errores del front, alertas básicas cuando el porcentaje de reveal por debajo del umbral esperado o cuando se detecten estados stale en aumento

## Plan de retroceso

En caso de fallo crítico del contrato en testnet se inmoviliza la interfaz de escritura y se mantienen solo funciones de lectura, se documenta el incidente y se prepara una migración a una nueva versión del contrato con un mapa de ids, para el front e indexador se conserva la versión estable anterior y se habilita un interruptor de rollback en el orquestador

# Conclusiones

SafeTraffic Ledger combina inmutabilidad, transparencia y gobernanza para reducir la manipulación de registros viales. El prototipo propuesto prioriza medidas realistas como commit–reveal, feed público y roles con disputas en una arquitectura sencilla y escalable. Los siguientes pasos incluyen validación institucional y ajustes normativos.