WHITEPAPER · VERSIÓN 0.4

El Ojo

Protocolo descentralizado de supervisión ciudadana verificable

Hoja de ruta desde una plataforma de transparencia de datos públicos hacia un protocolo descentralizado, basado en blockchain y gobernado por su comunidad.

ESTADO Primer borrador público

SITIO elojo.xyz

Aviso legal

Este documento tiene fines exclusivamente informativos y describe un proyecto en desarrollo. No constituye una oferta de valores, instrumentos financieros ni tokens, ni una solicitud de inversión de ningún tipo.

Esta versión del protocolo no contempla la emisión ni la venta de tokens. Las donaciones descritas son aportes voluntarios, sin expectativa de retorno financiero para quien dona.

Las secciones referidas a fases futuras contienen proyecciones sujetas a cambios técnicos, económicos y regulatorios. Los parámetros económicos (comisiones, montos, umbrales y costos) son ilustrativos y se calibrarán con datos reales.

Nada de lo aquí expuesto constituye asesoría legal, tributaria ni financiera. Los indicadores, alertas y hallazgos a los que se refiere son candidatos a revisar, nunca afirmaciones de irregularidad ni de incumplimiento sobre personas o instituciones. El Ojo no formula denuncias: cada hallazgo y su narrativa son de exclusiva responsabilidad de quien los publica.

Información del documento

Campo	Valor
Título	El Ojo — Whitepaper
Versión	0.4 (borrador)
Fecha	Septiembre de 2026
Autor	Sauron Nakamoto
Sitio	elojo.xyz
Licencia del código	AGPL-3.0 (prevista desde el inicio de la fase 2)

Índice

Resumen

El Ojo es una plataforma de transparencia que ordena, cruza y analiza los datos que publican los Estados de América Latina, hoy en cinco países y con la puerta abierta a sumar otros. Su propósito es generar alertas tempranas tanto sobre el eventual mal uso de los recursos públicos como sobre el cumplimiento presupuestario y la calidad del gasto fiscal. Este documento describe su evolución hacia un **protocolo descentralizado de supervisión ciudadana**, en el que cualquier persona puede publicar **hallazgos verificables**, la comunidad puede comprobarlos de forma independiente y los autores reciben recompensas directas por su trabajo.

Un hallazgo verificable reúne una o más consultas reproducibles sobre versiones fechadas de la base de datos, de modo que cualquiera puede auditar sus resultados. Para que ese registro resista la censura y la manipulación, las versiones se anclan en la red Base mediante raíces Merkle, los datos se resguardan de manera permanente en Arweave y la validez de cada hallazgo la atestiguan verificadores independientes que responden con un depósito en garantía. Las donaciones llegan de forma transparente y sin intermediarios a las billeteras de los autores, a través de contratos inteligentes y sin que El Ojo custodie los fondos en ningún momento.

La transición ocurre en cinco fases, cuyo avance depende del cumplimiento de métricas de adopción real y no de plazos fijos. Su propósito es transformar una iniciativa unipersonal en una infraestructura pública duradera, que no dependa de una sola persona y que quede al servicio de la supervisión democrática y fiscal en Chile y la región.

> **Principio rector: «Mirar no es acusar».** Todo indicador, alerta o hallazgo es un candidato a revisar, nunca una afirmación de irregularidad ni de incumplimiento. El protocolo verifica datos y cálculos; no emite juicios sobre intenciones o personas.

1. Introducción

1.1 Contexto

El Ojo opera hoy cinco ediciones, Chile, Argentina, Colombia, Uruguay y Perú, publicadas en elojo.xyz, y puede incorporar nuevos países cuando se decida, porque cada edición se construye con la misma especificación de modelos y esquemas. Cada una cruza información de presupuesto, compras públicas, lobby, declaraciones de patrimonio e intereses y actividad legislativa, usando llaves de identificación normalizadas por país, lo que permite seguir tanto el eventual mal uso de los recursos como el cumplimiento presupuestario y la calidad del gasto fiscal. Todo ese material proviene de fuentes oficiales que los propios Estados publican por mandato legal (sección 3.1).

1.2 Propuesta

Sobre esa base, el protocolo agrega tres componentes:

1. **Un estándar de hallazgo verificable.** Se trata de una o más consultas reproducibles sobre versiones fechadas de la base, cuyos resultados cualquiera puede recalcular.
2. **Un mecanismo de recompensa ciudadana.** Quien encuentra un patrón relevante, a quien este documento llama *generador de hallazgos*, puede recibir donaciones directas de quienes valoran ese trabajo, y la plataforma retiene una comisión para sostener su operación.
3. **Una capa de confianza on-chain.** Las versiones de la base, los hallazgos y sus verificaciones quedan registrados en una cadena de bloques (Nakamoto, 2008; Buterin, 2014), y los datos, en almacenamiento permanente, de modo que nadie, tampoco El Ojo, pueda alterarlos sin que se note.

1.3 Ruta de cinco fases

Fase	Nombre	Hito central
1	Génesis	Datos y alertas abiertos; hallazgos ciudadanos off-chain; donaciones para sostener el sitio
2	Canalización	El Ojo recibe donaciones y las dirige a hallazgos
3	Billeteras vinculadas	Donaciones directas a hallazgos específicos en billeteras Web3
4	Verificabilidad on-chain	Versiones, hallazgos y verificaciones registrados en Base
5	Descentralización plena	Datos en Arweave, tesorería y gobernanza comunitaria

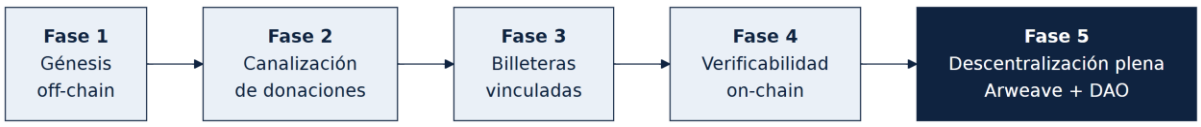

Figura 1. *Secuencia de fases del protocolo.*

El proyecto nace como una iniciativa personal, financiada por donaciones que sostienen el sitio y remuneran a su fundador, y aspira a terminar convertido en un bien público gobernado por su comunidad. Esta estrategia de traspaso gradual, conocida como descentralización progresiva (Walden, 2020), permite validar primero que el producto sirve y sólo después ceder su control.

2. El problema

Los Estados publican cada vez más datos, pero muy pocas personas los revisan. La información está dispersa, es técnica y revisarla no reporta beneficio alguno a quien lo hace, de manera que la tarea queda en manos de unos pocos, o incluso desierta. Esto vale tanto para los indicios de mal uso de recursos como para preguntas más cotidianas y no menos importantes: si los presupuestos aprobados se cumplen, si el gasto se ejecuta a tiempo y si rinde lo que promete.

2.1 Brechas de la supervisión ciudadana

Brecha	Descripción
Dispersión	Presupuesto, compras, lobby, declaraciones y Congreso viven en portales distintos, con llaves y formatos distintos.
Costo de cruce	Detectar un patrón (un proveedor concentrado, una reunión de lobby previa a una adjudicación, una ejecución presupuestaria que se acumula a fin de año) exige ETL, limpieza y cruces que un ciudadano difícilmente puede hacer solo.
Opacidad del ciclo presupuestario	La ley de presupuestos, sus modificaciones y la ejecución mensual se publican en formatos técnicos que pocos pueden leer, y casi nadie los compara entre sí ni con lo aprobado.
Sin incentivo	La supervisión ciudadana funciona como un bien público: sus beneficios se reparten entre todos, pero su costo lo asume quien la ejerce, por lo que tiende a producirse menos de lo necesario (Olson, 1965).
Sin prueba de integridad	Un hallazgo basado en una captura de pantalla o en un CSV descargado es fácil de impugnar, porque nadie puede demostrar qué decía el dato ni cuándo.
Dependencia de intermediarios	La supervisión depende de medios, ONG, fiscalías y órganos de control, todos con capacidad y recursos limitados.

2.2 Antecedentes

La idea de que los ciudadanos controlen a sus autoridades no es nueva. O'Donnell (1998) distinguió entre la rendición de cuentas horizontal, que ejercen unos órganos del Estado sobre otros, y la vertical, que ejercen los votantes en las elecciones. Smulovitz y Peruzzotti (2000) agregaron una tercera vía, la *accountability* social, para describir el control que ejercen asociaciones, movimientos y medios de comunicación en las democracias latinoamericanas, y que opera aun cuando no existan mecanismos formales de sanción (Peruzzotti y Smulovitz, 2006). El Ojo se sitúa en esta última tradición: no reemplaza a contralorías, tribunales ni fiscalías, sino que busca entregar a la ciudadanía herramientas para activarlos.

La evidencia empírica, sin embargo, advierte que la transparencia por sí sola no basta. En un experimento de campo con más de 600 proyectos viales en Indonesia, Olken (2007) encontró que aumentar la participación ciudadana en el monitoreo tuvo un efecto promedio pequeño, limitado por problemas de acción colectiva y por la captura de las élites locales, mientras que las auditorías estatales redujeron de manera significativa los gastos no justificados. Es el problema que Olson (1965) describió para los bienes públicos: cuando el beneficio se reparte entre todos y el costo lo asume uno, pocos están dispuestos a asumirlo. Fox (2015), al revisar la evidencia sobre *accountability* social, llega a una conclusión parecida, pues las intervenciones que se limitan a entregar información rinden menos que aquellas que la combinan con capacidad de acción colectiva y con instituciones dispuestas a responder.

Por otra parte, cuando la información llega a quien puede usarla, sí produce efectos. Ferraz y Finan (2008) mostraron que la divulgación de auditorías municipales en Brasil redujo la probabilidad de reelección de los alcaldes en cuyas comunas se detectaron irregularidades, sobre todo donde había radios locales que difundieran los resultados. En materia de compras públicas existe, además, una línea de trabajo consolidada que construye indicadores de riesgo, o banderas rojas, a partir de datos abiertos de contratación (Fazekas et al., 2016; Fazekas y Kocsis, 2020; Open Contracting Partnership y Development Gateway, 2016), que es precisamente lo que hacen los criterios de alerta de El Ojo.

La supervisión ciudadana tampoco se agota en la detección de irregularidades. Existe un campo consolidado de transparencia fiscal que se ocupa de algo más amplio: que el presupuesto sea público y comprensible, que su ejecución pueda compararse con lo aprobado y que la ciudadanía pueda participar en su discusión. Así lo recogen el Código de Transparencia Fiscal del Fondo Monetario Internacional (2019), el marco PEFA para evaluar la gestión de las finanzas públicas (PEFA Secretariat, 2016) y los principios de la Iniciativa Global para la Transparencia Fiscal, que la Asamblea General de las Naciones Unidas acogió en 2012 (Global Initiative for Fiscal Transparency, 2012), y lo mide cada dos años la Encuesta de Presupuesto Abierto (International Budget Partnership, 2024). La evidencia asocia una mayor apertura fiscal con más disciplina y credibilidad presupuestaria (Alt y Lassen, 2006; de Renzio y Wehner, 2017), aunque advierte, igual que en materia de corrupción, que la información sólo produce efectos cuando alguien con capacidad de actuar la usa (Khagram et al., 2013).

De esta revisión se desprenden dos lecciones que orientan el diseño del protocolo. La primera es que la supervisión ciudadana necesita incentivos, porque la buena voluntad no alcanza para sostenerla en el tiempo. La segunda es que la información sirve cuando se integra a las decisiones de quienes la usan (Fung et al., 2007), y para eso tiene que ser confiable: cualquiera debe poder comprobar de dónde viene y cómo se calculó. Ambas valen por igual para el control de la probidad y para el seguimiento del presupuesto y del gasto.

2.3 Tesis

La tesis de este documento es que la supervisión ciudadana, tanto la que busca indicios de mal uso de los recursos como la que sigue el cumplimiento del presupuesto y la calidad del gasto, puede crecer si se cumplen tres condiciones: que el dato sea de fácil acceso, que el hallazgo sea reproducible y que la recompensa llegue directamente a quien hizo el trabajo. Cada pieza del protocolo responde a una de esas condiciones o a la dependencia que hoy existe respecto de los intermediarios:

- **El cruce** lo resuelve El Ojo, con bases por país, llaves normalizadas (RUT, CUIT, NIT, RUC) y criterios de alerta que señalan dónde mirar.
- **La confianza** descansa en el hallazgo verificable, porque cualquiera puede recalcular el resultado sobre la misma versión de la base.
- **La integridad y el pago** quedan a cargo de la cadena de bloques, que ancla versiones y hallazgos de forma inmutable y permite donar sin intermediarios.
- **La independencia** se logra con la descentralización, para que el protocolo sobreviva a su fundador y ningún actor pueda capturarlo por sí solo.

3. La plataforma base

El Ojo ya está en producción. Hoy son cinco ediciones independientes, que se publican juntas detrás de una portada común en elojo.xyz, y cada país que se incorpore se sumará como una edición más. Todas ellas comparan ya la ejecución de cada institución con su presupuesto vigente, lo que permite seguir el cumplimiento presupuestario y advertir anomalías en el gasto.

3.1 Ediciones y fuentes

Edición	Fuentes principales
Chile	CGR, ChileCompra, InfoLobby, Congreso, SERVEL, InfoProbidad
Argentina	Presupuesto (MECON), COMPR.AR, CONTRATAR, Lobby (RUA), DDJJ (OA), Congreso
Colombia	Presupuesto PGN (SIIF), SECOP II, DDJJ y conflictos (DAFP), UTL del Congreso
Uruguay	Compras ARCE (OCDS), RUPE, vínculos ONSC, omisos JUTEP, Parlamento
Perú	SEACE/CONOSCE, visitas PCM, Congreso, Invierte.pe (MEF)

Buena parte de estas fuentes existe porque una ley obliga a publicarlas. En Chile, las principales son la Ley 20.285 sobre acceso a la información pública, que entre otras cosas obliga a publicar el presupuesto asignado a cada organismo y los informes sobre su ejecución; la Ley 19.886 de compras públicas, modernizada en 2023 por la Ley 21.634; la Ley 20.730 del lobby, y la Ley 20.880 sobre probidad. En los demás países cumplen un papel equivalente sus leyes de acceso a la información pública, como la Ley 27.275 en Argentina, la Ley 1712 de 2014 en Colombia, la Ley 18.381 en Uruguay y la Ley 27806 en Perú.

3.2 Stack tecnológico

- **Datos:** una base DuckDB por país (Raasveldt y Mühleisen, 2019), que los sitios abren en modo de solo lectura. Hoy suman unos 4,5 GB y se proyecta que lleguen a 100 GB en régimen.
- **Aplicación:** FastAPI, Jinja2, HTMX y Plotly. Cada edición corre sola o montada bajo la portada.
- **Nube:** Cloud Run con las bases versionadas en Google Cloud Storage y un puntero de versión activa por país.
- **Principios comunes:** toda alerta es un candidato a revisar y nunca una afirmación; las versiones de las dependencias están fijadas y cada país funciona aislado de los demás.

Como cimiento del protocolo, la plataforma aporta tres cosas. La primera son **datos ya normalizados y cruzados**, que representan el trabajo más costoso de la supervisión. La segunda es un **versionado existente**, porque cada publicación de la base ya es una versión identificable y, por lo tanto, el insumo natural del anclaje. La tercera es la **reproducibilidad**, ya que las vistas analíticas son consultas SQL que cualquier tercero puede volver a ejecutar.

3.3 Rol de El Ojo en el protocolo

El Ojo no produce hallazgos. Su papel es proveer la materia prima ordenada y los puntos de partida, mientras que el hallazgo es obra del ciudadano.

Rol	Qué aporta	Ejemplos
El Ojo	Información ordenada, cruzada y tratada	Presupuesto por institución, compras por proveedor, audiencias de lobby, declaraciones de patrimonio, dotación del Congreso
El Ojo	Criterios de alerta, siempre como candidatos a revisar	Desviaciones presupuestarias, contrataciones express, proveedores concentrados (reglas R1–R19 en Chile, A1–A14 en Argentina)
Ciudadano	Hallazgos verificables: combina consultas, fichas y alertas para mostrar un patrón	Funcionarios de un parlamentario cruzados con la variación de su patrimonio; ejecución de un servicio comparada con su presupuesto y con la de sus pares
Comunidad (desde F4)	Verificación independiente de los hallazgos	Reejecución y atestación on-chain

Esta separación de roles resguarda el principio «Mirar no es acusar», porque la plataforma nunca afirma nada: se limita a ordenar la información y a alertar sobre lo que merece revisión.

El Ojo jamás denuncia. La narrativa de cada hallazgo, y el documento que la acompañe, son de responsabilidad exclusiva de su autor, que responde personalmente por ellos, incluso ante los tribunales.

4. El hallazgo verificable

Un hallazgo verificable es un **conjunto de una o más consultas reproducibles**, cada una ejecutada sobre una versión fechada de la base, que en conjunto muestran un patrón. Es la unidad de valor del protocolo, pues es lo que el ciudadano publica, lo que la comunidad verifica y lo que los donantes recompensan. La idea recoge una exigencia conocida de la investigación empírica, la de que cualquier resultado pueda reproducirse a partir de los mismos datos y del mismo código (Peng, 2011), y la traslada a la supervisión del Estado.

4.1 Definición formal

Un hallazgo *H* es la tupla:

$$H = (\{C_1,\ \ldots,\ C_n\},\ N,\ D,\ A), \qquad C_i = (\text{país}_i,\ Q_i,\ P_i,\ V_i,\ h(R_i))$$

Cada consulta *C* tiene su país, su SQL *Q*, sus parámetros *P*, su versión de base *V* y el hash de su resultado *h(R)*. *N* es la narrativa que las une; *D*, un documento narrativo opcional, por ejemplo un PDF (sección 4.6), y *A*, su autor. *H* es válido si y sólo si **todas** sus consultas, ejecutadas otra vez sobre la versión indicada, reproducen el hash declarado. Para que el hallazgo no pueda alterarse después de publicado, su hash se calcula como la raíz Merkle (Merkle, 1988) de hojas que describen cada consulta completa, y no sólo su resultado, más los hashes de la narrativa y del documento:

$$h(C_i) = \text{SHA-256}\,(\text{país}_i \parallel Q_i \parallel P_i \parallel V_i \parallel h(R_i))$$

$$h(H) = \text{MerkleRoot}\,(h(C_1),\ \ldots,\ h(C_n),\ h(N),\ h(D))$$

El autor firma *h(H)* con su billetera desde la fase 3, conforme al estándar EIP-712 (Bloemen et al., 2017), lo que acredita la autoría sin que nadie custodie sus llaves. Así, si cambia una consulta, un parámetro, la versión de la base, un resultado, la narrativa o un solo byte del documento, cambia también *h(H)* y la firma deja de ser válida. Cuando el hallazgo no trae documento, *h(D)* simplemente se omite.

4.2 Ejemplos

El primer ejemplo explora un posible conflicto de intereses en el Congreso:

Consulta	Ficha de origen	Qué muestra
C1	Dotación y asesores del Congreso	Funcionarios contratados por el parlamentario, por año
C2	Declaraciones de patrimonio	Variación del patrimonio declarado del mismo parlamentario
C3	Compras públicas	Contratos adjudicados a personas vinculadas a esos funcionarios
N	—	Narrativa: el patrón conjunto y por qué merece revisión

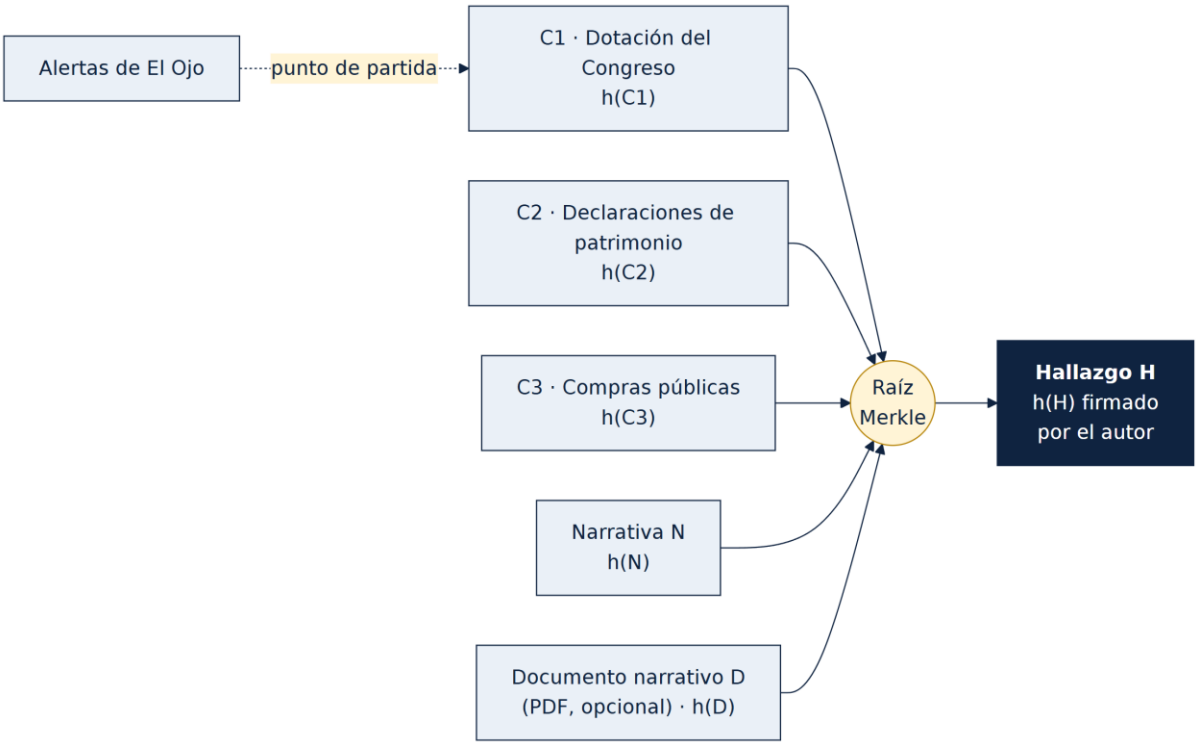

Figura 2. *Composición de un hallazgo a partir de sus consultas, su narrativa y su documento.*

Ninguna de estas consultas constituye por sí sola un hallazgo. Lo que las convierte en uno es el cruce que propone el ciudadano y la narrativa con que lo explica. Las consultas pueden combinar fichas distintas y, más adelante, también países distintos.

El mismo esquema sirve para hallazgos de cumplimiento presupuestario y de calidad del gasto, que no suponen irregularidad alguna. El segundo ejemplo muestra un servicio cuya ejecución se concentra al final del año:

Consulta	Ficha de origen	Qué muestra
C1	Ejecución presupuestaria	Presupuesto vigente y gasto devengado del servicio, mes a mes
C2	Ejecución presupuestaria	Proporción del gasto anual devengada en diciembre, frente a servicios similares
C3	Compras públicas	Órdenes de compra emitidas por el servicio en ese mismo mes
N	—	Narrativa: el patrón de concentración y por qué merece revisión como posible problema de planificación

En este caso, el hallazgo no sugiere que alguien haya actuado de forma indebida, sino que el presupuesto podría estar planificándose o ejecutándose mal. Es el tipo de hallazgo que interesa a las comisiones de presupuesto y a los propios servicios, y el protocolo lo trata igual que a cualquier otro.

4.3 Esquema de datos

Nivel	Campo	Contenido	Regla
Hallazgo	`id`	Identificador único	UUID en fases 1–3; desde la fase 4, el propio h(H)
Hallazgo	`narrativa`	Explicación del patrón	Plantilla fija: patrón observado, fuentes, por qué merece revisión

Nivel	Campo	Contenido	Regla
Hallazgo	`documento`	Documento narrativo opcional (por ejemplo, un PDF) y su SHA-256	Opcional; tamaño máximo como parámetro; su hash forma parte de h(H)
Hallazgo	`autor`	Generador de hallazgos	Seudónimo; billetera desde la fase 3
Hallazgo	`firma`	Firma EIP-712 del autor sobre h(H)	Desde la fase 3
Hallazgo	`hash_hallazgo`	Raíz Merkle h(H) de las consultas completas, la narrativa y el documento	Cambia si cambia cualquier consulta, parámetro, versión, resultado, la narrativa o el documento
Hallazgo	`alertas`	Alertas de El Ojo usadas como punto de partida	Opcional
Hallazgo	`estado`	Ciclo de vida	Ver figura 3
Consulta	`orden`	Posición en el hallazgo	1…n
Consulta	`pais`	Edición sobre la que se ejecuta	Una de las cinco
Consulta	`consulta_sql`	La consulta	Sólo SELECT, contra una lista permitida de vistas públicas
Consulta	`parametros`	Valores enlazados	Nunca interpolados en el SQL
Consulta	`version_base`	Versión de la base usada	Fechada; desde la fase 4, su raíz Merkle on-chain
Consulta	`hash_resultado`	SHA-256 del resultado canónico	Orden estable y tipos normalizados
Consulta	`entidades`	Instituciones, cargos o proveedores involucrados	Por su llave de cruce

4.4 Ciclo de vida

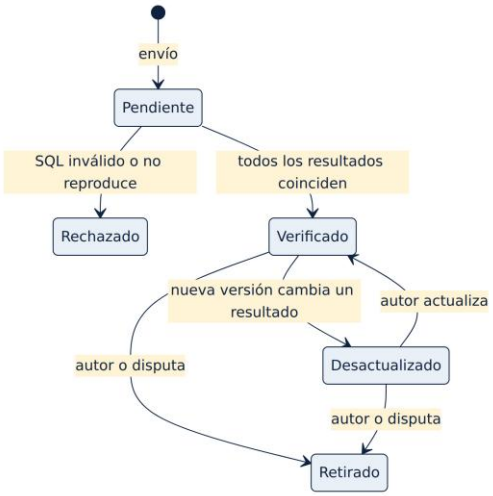

Figura 3. *Estados de un hallazgo.*

Cada nueva versión de la base dispara la reverificación de todos los hallazgos vigentes, es decir, la ejecución de sus consultas sobre los datos actualizados. Si el resultado cambia, el hallazgo pasa a «desactualizado», pero nunca se pierde, porque sigue fijado a las versiones sobre las que se hizo y sigue siendo reproducible sobre ellas. Por eso conviene que las consultas acoten el periodo que analizan: una consulta sobre años cerrados seguirá dando el mismo resultado, mientras que una sobre el año en curso cambiará con cada publicación o con cada corrección de la fuente.

4.5 Verificación automática

La verificación consiste en volver a ejecutar las consultas, no en emitir una opinión sobre ellas. Eso permite operar sin moderadores, porque es la máquina la que determina si el dato y el cálculo son correctos. El único componente que requiere criterio humano es la narrativa, que forma parte del hallazgo y queda sellada en su hash, pero cuyo mérito el protocolo no juzga: esa valoración ocurre fuera del mundo de El Ojo, en el uso que periodistas, autoridades y ciudadanos hagan del hallazgo.

- **Aislamiento:** DuckDB en modo de solo lectura, sin acceso a la red, con la configuración bloqueada y límites de memoria, hilos, tiempo de ejecución y filas.
- **Separación:** el verificador corre como un proceso aparte, nunca dentro del sitio público.
- **Determinismo:** el resultado se canonicaliza (orden estable y tipos normalizados) antes de calcular su hash.

4.6 Documento narrativo

Además de la narrativa breve que exige la plantilla, cada hallazgo puede incluir un documento narrativo, por ejemplo un PDF, en el que el autor desarrolla el análisis con la extensión, los gráficos y las referencias que estime necesarios. El documento no reemplaza a la narrativa *N*, que sigue siendo obligatoria y se ajusta a la plantilla, sino que la complementa, y queda sellado en el hallazgo del mismo modo que las consultas: su hash *h(D)* es una hoja más de la raíz Merkle, de manera que cualquier cambio posterior en el archivo, aunque sea de un solo byte, cambia *h(H)* e invalida la firma del autor.

A diferencia de los resultados de las consultas, el documento no necesita canonicalizarse: *h(D)* es el SHA-256 (National Institute of Standards and Technology, 2015) de los bytes exactos del archivo que el autor firma, y verificarlo consiste simplemente en descargarlo y volver a calcular ese hash.

El análisis de factibilidad muestra que el costo adicional es marginal. Calcular el hash toma milisegundos y, como *h(D)* viaja dentro de la raíz del hallazgo, no agrega transacciones ni gas en la cadena. El almacenamiento tampoco pesa: con documentos de 2 MB en promedio, 10.000 hallazgos suman unos 20 GB, que en Google Cloud Storage cuestan menos de un dólar al mes (Google Cloud, s. f.) y en Arweave, con la banda de la sección 7.6, entre 40 y 400 USD por una sola vez, frente a los 100 GB que ocupará la base en régimen. Por eso el documento narrativo se incorpora al diseño desde la fase 1, con un tamaño máximo por archivo que la gobernanza podrá ajustar.

Los costos que sí importan no son monetarios, y condicionan la forma de implementarlo:

- **Preservación:** se recomendará PDF/A, la variante del formato pensada para la conservación a largo plazo (International Organization for Standardization, 2005), para que el documento siga siendo legible con los años.
- **Seguridad:** un PDF puede contener código o archivos incrustados, por lo que el sitio mostrará una versión saneada y conservará el original sólo para su descarga y para verificar el hash.
- **Metadatos:** los PDF suelen registrar el nombre de quien los creó y el programa usado, lo que podría revelar la identidad de un autor que publica con seudónimo. El editor ofrecerá limpiarlos antes de firmar, porque después de la firma cualquier cambio altera el hash.
- **Responsabilidad:** El Ojo no revisa el contenido del documento; como la narrativa, es de responsabilidad exclusiva de su autor.

5. Arquitectura del protocolo

En su estado final, El Ojo se organiza en cuatro capas: datos, anclaje, verificación y financiamiento. Sólo la aplicación web depende de infraestructura propia, y aun ella es reemplazable, porque cualquiera podría levantar otra interfaz sobre los mismos datos y contratos. La tabla que sigue a la figura agrupa esos componentes según cómo se implementan.

5.1 Visión general

Figura 4. *Arquitectura del protocolo en su estado final (fase 5).*

Capa	Componentes	Tecnología	Fase
Datos	ETL, bases por país, exportación a Parquet, manifiestos	Python, DuckDB, Parquet, GCS → Arweave	Existe; exportación en F4; Arweave en F5
Verificación	Verificador aislado, cliente CLI abierto	Python, DuckDB, Cloud Run Jobs	F1 central; F4 abierto
Contratos	Donaciones, registros, staking y disputas, tesorería, gobernanza	Solidity en Base, USDC, EAS, Safe	F3 a F5
Aplicación	Portada, cinco ediciones, módulo de hallazgos, conexión de billeteras	FastAPI, Jinja2, HTMX, SIWE	Existe; hallazgos en F1; billeteras en F3

5.2 Contratos inteligentes

Contrato	Función	Fase	Dueño inicial	Dueño final
`DonacionHallazgo`	Donación directa con comisión separada; membresías	F3	Multisig	DAO
`RegistroVersiones`	Raíz Merkle y manifiesto por versión y país	F4	Multisig	DAO
`RegistroHallazgos`	Hash, autor, versiones y estado de cada hallazgo	F4	Sin dueño	Sin dueño
`StakingVerificadores`	Stake, asignación aleatoria, commit-reveal, recompensas, slashing y disputas	F4	Multisig	DAO
`Tesoreria`	Comisiones, membresías, rondas de matching	F5	—	DAO
`Gobernanza` + `Timelock`	Cambios de parámetros y actualizaciones	F5	—	Comunidad

5.3 Versionado de datos

Cada publicación de la base genera, por país, un manifiesto que lista sus particiones en formato Parquet (Apache Software Foundation, s. f.) con un doble hash. El primero, `hash_archivo`, se calcula sobre los bytes y sirve para el almacenamiento y la descarga; el segundo, `hash_logico`, se calcula sobre el contenido canónico y se mantiene estable aunque cambie la versión del motor. El manifiesto cumple así la función de un registro de procedencia de los datos (Buneman et al., 2001). Lo que se ancla on-chain es su raíz Merkle, siguiendo la técnica de sellado de tiempo de Haber y Stornetta (1991): basta con publicar un hash para probar, más tarde, que un documento existía en esa forma y en esa fecha.

```
{
  "pais": "chile", "version": "2027-03-31", "duckdb": "1.x.y",
  "particiones": [
    {"tabla": "fct_compras", "clave": "2027/03", "filas": 184233,
     "hash_archivo": "sha256:…", "hash_logico": "sha256:…", "uri": "ar://…"}
  ],
  "vistas": {"v_proveedor_concentrado": "sha256:<hash del SQL>"},
  "merkle_root": "0x…", "version_anterior": "2027-02-28"
}
```

Figura 5. *Estructura del manifiesto de versión.*

Requisito crítico: exportación determinista. Las filas deben ordenarse por su clave natural, con la versión de DuckDB y los parámetros de Parquet fijos y sin metadatos variables. Sin eso, dos exportaciones del mismo contenido producirían hashes distintos y se perdería la deduplicación entre versiones. Las vistas no se exportan; lo que se publica es su SQL, con un hash propio.

5.4 Almacenamiento permanente

Elemento	Diseño
Formato	Parquet particionado por país, tabla, año y mes
Versiones nuevas	Sólo se suben las particiones cuyo hash cambió; las demás se referencian por su transacción existente
Manifiesto	También en Arweave; su identificador queda registrado en `RegistroVersiones`
Documentos narrativos	Se guardan junto a su hallazgo y, en la fase 5, también en Arweave; su hash ya forma parte de h(H)
Consulta	DuckDB lee Parquet por HTTP desde gateways de Arweave, sin descargar la base completa
Espejo	GCS sigue sirviendo el sitio por rendimiento, pero deja de ser la fuente de verdad

5.5 Principios de diseño

- **En cadena, sólo hashes, punteros y dinero.** Los datos pesados viven en almacenamiento direccionado por contenido, y ningún dato personal se escribe en la cadena.
- **Contratos mínimos y auditados** antes de cada despliegue en mainnet, con pruebas previas en Base Sepolia.
- **Roles separados:** un publicador automatizado con permisos acotados, revocable por el multisig.
- **Aislamiento:** ningún proceso escribe sobre las bases que sirven los sitios; los nuevos módulos (hallazgos, anclaje) viven como paquetes propios.

6. Mecanismo de verificación

En las fases 1 a 3, la verificación la realiza el proceso automático de El Ojo. Desde la fase 4 puede verificar cualquiera que deposite una garantía (*stake*) y ejecute el cliente abierto. Como la ejecución es determinista, basta con que un solo participante honesto revise un hallazgo para detectar un error y abrir una disputa. Es el mismo supuesto de seguridad sobre el que descansan los sistemas de verificación optimista, como TrueBit (Teutsch y Reitwießner, 2019) o Arbitrum (Kalodner et al., 2018). Esos mismos trabajos advierten, eso sí, del llamado dilema del verificador: si verificar cuesta y casi todo resulta válido, los participantes racionales dejan de verificar (Luu et al., 2015). Las protecciones de la sección 6.3 están pensadas para ese problema.

6.1 Verificadores

Elemento	Diseño
Quién	Cualquiera con stake en USDC: periodistas de datos, ONG de transparencia, estudiantes y académicos, técnicos con infraestructura ociosa, otros generadores de hallazgos (nunca sobre sus propios hallazgos)
Requisito técnico	Bajo: sólo descarga las particiones que usan las consultas del hallazgo, no la base completa
Cliente	Verificador de código abierto (CLI en Python + DuckDB) que cualquiera puede correr localmente
Asignación	*k* verificadores aleatorios por hallazgo, ponderados por stake, con aleatoriedad verificable (VRF; Micali et al., 1999)
Recompensa	Una fracción de la comisión de las donaciones al hallazgo verificado; la tesorería puede subsidiar la verificación de hallazgos que todavía no reciben donaciones
Penalización	Slashing del stake si una disputa demuestra una atestación falsa

6.2 Procedimiento

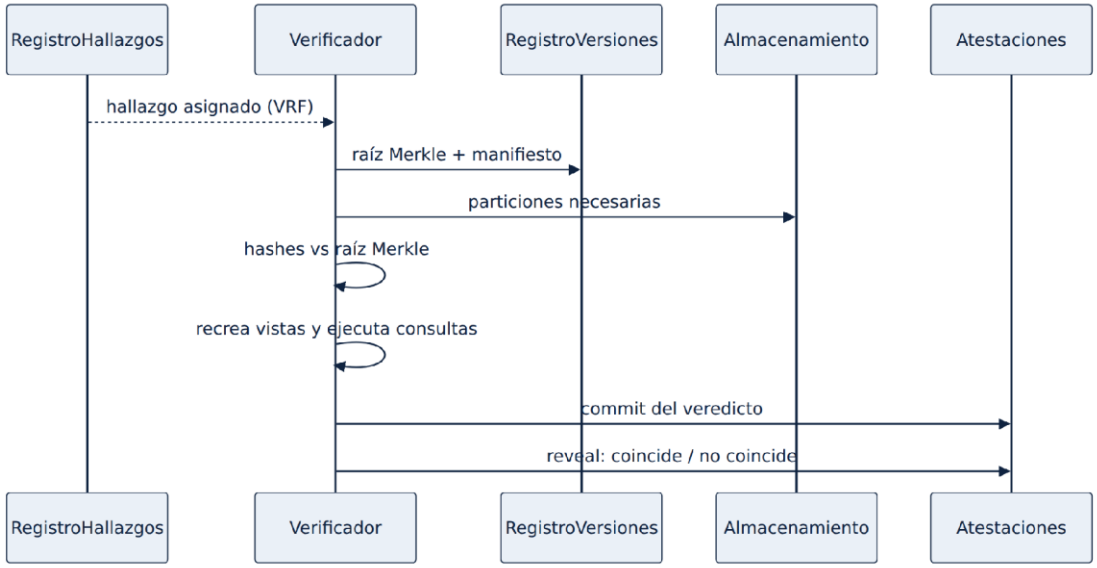

Figura 6. *Secuencia de verificación de un hallazgo.*

1. **Recibe la asignación** de un hallazgo desde `RegistroHallazgos`.

2. **Obtiene las versiones:** por cada consulta, la raíz Merkle y el manifiesto de su versión en `RegistroVersiones`.
3. **Descarga sólo lo necesario:** las particiones de las tablas que tocan las consultas.
4. **Comprueba la integridad:** recalcula el hash de cada partición y lo cuadra con la raíz anclada, lo que prueba que los datos son los publicados sin necesidad de confiar en El Ojo.
5. **Recrea las vistas** desde el SQL publicado en el manifiesto.
6. **Ejecuta cada consulta** en DuckDB aislado y canonicaliza su resultado.
7. **Compara** cada hash con el declarado, incluido el del documento narrativo si lo hay, y recalcula la raíz del hallazgo.
8. **Compromete y revela:** primero publica el hash de su veredicto combinado con un valor secreto y, al cerrar la ronda, revela ambos, con lo que su atestación queda registrada.
9. **Cobra** su parte de la recompensa si el hallazgo alcanza *k* atestaciones coincidentes y su veredicto está entre ellas.

6.3 Incentivos y protecciones

Problema	Descripción	Protección
Verificador perezoso	Copia atestaciones ajenas sin ejecutar	Commit-reveal: nadie ve el veredicto ajeno antes de comprometer el propio (Blum, 1983)
Colusión	Un grupo se pone de acuerdo	Asignación aleatoria ponderada por stake; la protección funciona mientras ningún grupo controle la mayor parte del stake total
Atestación falsa	Declara «coincide» sin que coincida	Disputa abierta a cualquiera; se resuelve reejecutando, y quien mintió pierde su stake
Complacencia	Aprueba todo porque casi todo es válido	Hallazgos trampa: el protocolo inyecta hallazgos con un hash incorrecto conocido y penaliza a quien los aprueba, como los errores forzados de TrueBit
Autoverificación	El autor verifica con otra billetera (ataque Sybil; Douceur, 2002)	Stake y membresía encarecen las cuentas múltiples; la asignación aleatoria reduce la probabilidad de que a alguien le toque su propio hallazgo

Los hallazgos trampa requieren un cuidado especial. Deben ser indistinguibles de los reales y su generación no puede quedar en manos de un solo actor, porque quien los crea sabe cuáles son. En la fase 4 los generará El Ojo, comprometiendo de antemano el hash de cada trampa para revelarlo al cierre de la ronda, y en la fase 5 lo hará un proceso definido por la gobernanza.

6.4 Disputas

Como la verificación es determinista, una disputa no se zanja por mayoría de opiniones, sino volviendo a ejecutar la consulta. Ahora bien, un contrato no puede ejecutar DuckDB, de modo que alguien tiene que llevar ese resultado a la cadena. La disputa la dirime entonces un panel más amplio de verificadores elegidos al azar, que vuelve a correr el hallazgo y deposita su propia garantía. Formalmente es una votación, pero sobre un hecho comprobable y no sobre una opinión, lo que convierte a la respuesta correcta en el punto focal natural del panel (Schelling, 1960), en la línea del arbitraje descentralizado de Kleros (Lesaege

et al., 2019). Mientras la red de verificadores sea pequeña, el multisig actuará como última instancia, y a largo plazo la idea es reemplazar el panel por pruebas criptográficas (sección 6.6).

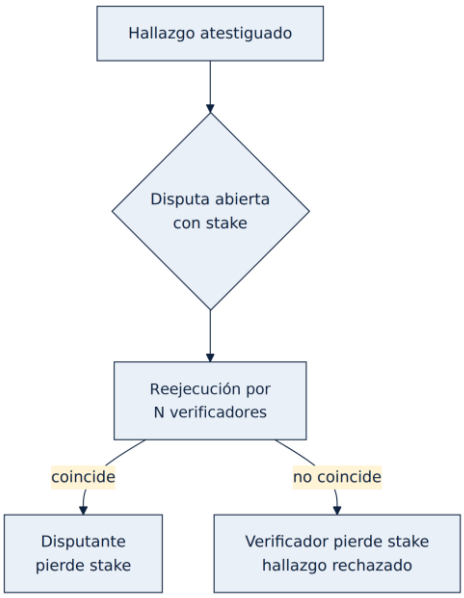

Figura 7. *Resolución de una disputa.*

`DonacionHallazgo` sólo libera los fondos cuando el hallazgo reúne al menos *k* atestaciones coincidentes. Mientras tanto, las donaciones quedan retenidas en el propio contrato (*escrow*) y, si el hallazgo es rechazado, vuelven al donante.

6.5 Alcance de la verificación

Prueba	No prueba
Que los datos son los publicados por El Ojo en esa versión	Que la fuente oficial esté correcta
Que cada consulta produce exactamente el resultado declarado	Que el cruce sea pertinente o no esté sesgado
Que el hallazgo no fue alterado después de publicarse	Que exista una irregularidad o un incumplimiento

La verificación es, entonces, técnica y binaria. La calidad del hallazgo, en cambio, la juzga la comunidad, ya sea con sus donaciones, con hallazgos de réplica que discutan o maticen el original o, si llega a adoptarse, mediante un rol de curadores que no tendría poder alguno sobre la validez técnica.

6.6 Horizonte: verificación criptográfica

Existen desarrollos académicos de pruebas criptográficas para consultas SQL, en los que quien ejecuta una consulta genera una prueba que un contrato puede validar sin volver a ejecutarla (Zhang et al., 2017; Li et al., 2023). Hoy estas técnicas todavía son inmaduras y demasiado costosas para bases del orden de 100 GB. En la fase 5 podrían complementar o incluso reemplazar a los verificadores, pero se tratan como una línea de investigación y no como una dependencia del diseño.

7. Modelo económico

En régimen, el protocolo se financia con tres fuentes: una comisión sobre las donaciones, la membresía de los generadores de hallazgos y los aportes a la tesorería. Durante las primeras fases se suman las donaciones vía Patreon al fundador. Todos los porcentajes de esta sección son parámetros ilustrativos, que se calibrarán con los datos de la fase 2, y conviene reiterar que esta versión del protocolo **no contempla la emisión ni la venta de tokens**.

7.1 Fuentes de ingreso

Fuente	F1	F2	F3	F4	F5
Patreon al fundador	Sí	Sí	Sí	Transición	No
Comisión sobre donaciones	—	Sí	Sí	Sí	Sí
Membresía de generador de hallazgos (1 USDC/mes)	—	—	Sí	Sí	Sí
Grants y fundaciones	Opcional	Opcional	Opcional	Sí	Sí (matching)

7.2 Destino de los ingresos

Fase	Destino
F1	Mantención del sitio (nube, datos, desarrollo) y remuneración del fundador
F2–F3	La comisión cubre la operación y la remuneración del fundador; el resto de cada donación va al generador de hallazgos
F4	Transición: la comisión se reparte entre operación, verificadores y un aporte creciente a la tesorería
F5	Todo ingreso del protocolo va a la tesorería de la DAO; el fundador cobra sólo si la DAO lo contrata como a cualquier contribuidor

7.3 Reparto de una donación

Para una donación ilustrativa de 100 USDC a un hallazgo verificado:

Destino	F2 y F3 (USDC)	F4 y F5 (USDC)
Generador de hallazgos	90	90
Plataforma / tesorería	10	7
Verificadores del hallazgo	—	3

7.4 Membresía de generador de hallazgos

$$I_{\text{membresías}} = N_{\text{generadores}} \times 1\ \text{USDC} \times 12$$

Con 500 generadores de hallazgos activos, la membresía reportaría 6.000 USDC al año. Su función principal, sin embargo, no es recaudar sino filtrar el spam y las cuentas falsas, porque el ingreso relevante proviene

de la comisión y del matching. Como filtro contra cuentas múltiples, 1 USDC mensual es una barrera modesta, por lo que la membresía debe complementarse con la reputación acumulada por cada autor.

7.5 Tesorería y matching

- **Tesorería del protocolo:** recibe las comisiones y las membresías.
- **Matching cuadrático:** rondas periódicas en las que la tesorería y las fundaciones aportan un fondo que se reparte según la fórmula de financiamiento cuadrático (Buterin et al., 2019), que favorece a los hallazgos con muchos donantes distintos por sobre los que reciben pocos aportes grandes. Así pesa más el apoyo amplio que el de un único donante.
- **Financiamiento retroactivo:** recompensas adicionales para hallazgos que hayan tenido consecuencias verificables, como un sumario, una auditoría, una corrección presupuestaria, una mejora en la ejecución del gasto o cobertura de prensa, bajo la lógica de que es más fácil ponerse de acuerdo sobre lo que resultó útil que sobre lo que lo será (Optimism, 2021). Este criterio debe aplicarse con cuidado, para no premiar el efecto mediático por sobre la solidez del hallazgo, además de requerir oráculos.
- **Presupuesto operativo:** almacenamiento, verificadores y desarrollo, aprobado por gobernanza.

7.6 Costos

Rubro	Fase	Orden de magnitud
Cloud Run, GCS y verificador	F1 en adelante	Infraestructura actual más procesos de verificación
Procesador de pagos fiat	F2	Comisión del procesador por transacción
Auditoría de contratos	F3 y F4	A cotizar; es el mayor gasto único antes de mainnet
Gas en Base	F3 en adelante	Centavos por transacción
Almacenamiento en Arweave	F5	Ver tabla siguiente
Documentos narrativos	F1 en adelante	Marginal: menos de 1 USD al mes por cada 10.000 documentos (sección 4.6)

Arweave cobra un pago único por almacenamiento permanente (Williams et al., 2019), cuyo precio en dólares depende de la cotización de su token. La tarifa ha rondado los 0,5 AR por GB (Arweave, s. f.), pero su equivalente en dólares ha variado en más de un orden de magnitud en los últimos años. Por eso la tabla usa una banda ilustrativa y conservadora de 2 a 20 USD por GB, aplicada a un volumen de 100 GB en régimen.

Estrategia	Volumen anual	Carga inicial	Costo anual
Snapshot completo mensual	1,2 TB	200–2.000 USD	2.400–24.000 USD
Deltas por partición (~3 GB/mes)	~36 GB	200–2.000 USD	72–720 USD

Por eso el diseño por deltas no es opcional, ya que reduce el costo recurrente unas 30 veces.

7.7 Punto de equilibrio

$$D_{\text{equilibrio}} = \frac{\text{Costo mensual} - \text{Membresías}}{\text{Comisión}}$$

Con una comisión de 10 %, cada 1.000 USD de costo mensual que no cubran las membresías exige 10.000 USD mensuales en donaciones a hallazgos. Desde la fase 4, en que la plataforma retiene 7 % y no 10 %, la misma cifra sube a unos 14.300 USD. Si existe o no ese volumen de donaciones es la pregunta de validación central de la fase 2.

8. Gobernanza

El Ojo nace con un solo desarrollador y aspira a terminar sin dueño. El control se traspasa en el mismo orden en que se construye el valor, empezando por el producto, siguiendo por la participación de la comunidad y terminando con la propiedad, que es la secuencia que propone la descentralización progresiva (Walden, 2020). La experiencia acumulada muestra que ese último paso es el más delicado, porque cuando el voto se basa en tokens transferibles el poder tiende a concentrarse en quienes pueden comprarlos (Buterin, 2021). Por eso el diseño de la gobernanza se deja para el final y se inspira en lo que Ostrom (1990) observó en las comunidades que administran con éxito bienes comunes: reglas definidas por quienes las usan, monitoreo a cargo de la propia comunidad y sanciones graduales.

8.1 Control por fase

Decisión	F1–F2	F3	F4	F5
Código y despliegue	Fundador	Fundador	Fundador + contribuidores	Comunidad
Publicación de versiones	Fundador	Fundador	Publicador revocable por multisig	Publicadores aprobados por la DAO
Validez de un hallazgo	Verificador de El Ojo	Verificador de El Ojo	Verificadores independientes	Verificadores independientes
Fondos de donaciones	El Ojo (custodia)	Contrato, sin custodia	Contrato, sin custodia	Contrato, sin custodia
Parámetros económicos	Fundador	Multisig	Multisig	DAO con timelock

8.2 Parámetros gobernados

Parámetro	Ejemplo
Comisión de la plataforma	Porcentaje sobre donaciones
Membresía de generador de hallazgos	Monto mensual
Umbral de verificación	*k* atestaciones coincidentes
Stake de verificadores y disputas	Montos mínimos y slashing
Lista permitida de vistas	Qué vistas admite el verificador
Documento narrativo	Tamaño máximo y formatos admitidos
Países y fuentes nuevas	Incorporación de ediciones
Publicadores de versiones	Quién puede anclar una base nueva

El mecanismo de voto se definirá al cierre de la fase 4, con datos reales de participación. Las alternativas en estudio son un token de gobernanza, una reputación no transferible obtenida por hallazgos verificados, en la línea de los *soulbound tokens* que proponen Weyl et al. (2022), o un modelo mixto.

8.3 Riesgo de desarrollador único

El riesgo de que un proyecto dependa de una sola persona es bien conocido en el software libre, donde buena parte de la infraestructura que todos usan descansa en muy pocos mantenedores (Eghbal, 2020). Las mitigaciones previstas son las siguientes:

Riesgo	Mitigación
Capacidad	Alcance acotado: las ediciones y vistas existentes en F1; verificación automática en lugar de moderación
Bus factor	Documentación de traspaso por componente, runbook de operación y guía de contribución
Llaves	Multisig 2-de-3 (Safe, s. f.) desde el primer contrato en mainnet, con al menos dos firmantes independientes del fundador; ninguna llave de administración en una sola billetera
Credibilidad del traspaso	Licencia AGPL-3.0, hoja de ruta pública y criterios de paso medibles
Contribuidores	Países y fuentes nuevas como tareas autocontenidas, gracias a la documentación canónica de modelos y esquemas
Comunidad inicial	Hallazgos como trabajo práctico en cursos universitarios y como insumo para el periodismo de investigación, las agencias de gobierno y las comisiones de presupuesto

8.4 Compromisos y traspaso

- Publicar el código completo y las bases al inicio de la fase 2, cuando comiencen a entregarse donaciones a los generadores de hallazgos, que es cuando la transparencia de la plataforma debe ser total; la fase 1 se dedica a construir y preparar esa apertura.
- Rendir cuentas públicamente de lo recibido por Patreon y de las donaciones canalizadas.
- No reservarse privilegios en los contratos al cierre de la fase 5.

El traspaso final sigue tres pasos:

1. Los contratos pasan de dueño multisig a un contrato de gobernanza con timelock.
2. El repositorio, los dominios y las credenciales de infraestructura pasan a una entidad de la comunidad (una fundación u otra figura equivalente).
3. El fundador conserva sólo los derechos que tenga cualquier participante.

9. Hoja de ruta

Cada fase agrega una capa de descentralización sobre la anterior. Las fases 1 y 2 sirven para validar que existen generadores de hallazgos y donantes; la fase 3 elimina la custodia de fondos; la 4, la necesidad de confiar en El Ojo como verificador, y la 5, la dependencia de su infraestructura y de su fundador.

	F1 Génesis	F2 Canalización	F3 Billeteras	F4 Verificabilidad	F5 Descentralización plena
Hallazgos	Off-chain	Off-chain	Off-chain, firmados con billetera	Registrados on-chain	Registrados on-chain
Donaciones	A El Ojo (Patreon)	A El Ojo, dirigidas a hallazgos	Directas a la billetera del autor	Directas, condicionadas a verificación	Directas + matching
Custodia	El Ojo	El Ojo	Contrato de reparto	Contrato de reparto	Contratos del protocolo
Verificación	Automática, central	Automática, central	Automática, central	Independiente + disputas	Independiente + disputas
Datos	GCS	GCS	GCS	GCS, anclados en Base	Arweave, anclados en Base
Control	Fundador	Fundador	Fundador + multisig	Multisig	DAO

9.1 Fase 1 — Génesis

Objetivo: demostrar que hay ciudadanos dispuestos a producir hallazgos verificables aun sin un incentivo económico directo. En esta fase, las donaciones vía Patreon sostienen el sitio y remuneran a su fundador durante el desarrollo de la infraestructura que exige el paso a la segunda fase. El Ojo aporta los datos y las alertas, y los ciudadanos, los hallazgos.

Entregable	Descripción
Módulo de hallazgos	Paquete común a las cinco ediciones y agnóstico al país
Almacén de hallazgos	Base separada de las bases de país, que son de solo lectura
Editor de hallazgo	Una o más consultas SQL, autocompletado de vistas permitidas, enlace a alertas, plantilla de narrativa y carga del documento narrativo
Verificador	Proceso aislado que ejecuta las consultas y calcula sus hashes
Página pública por hallazgo	Consultas, resultados, versiones, narrativa y botón «reejecutar»
Ranking de generadores de hallazgos	Reputación por hallazgos verificados y vigentes
Enlace a Patreon	En la portada y en cada edición

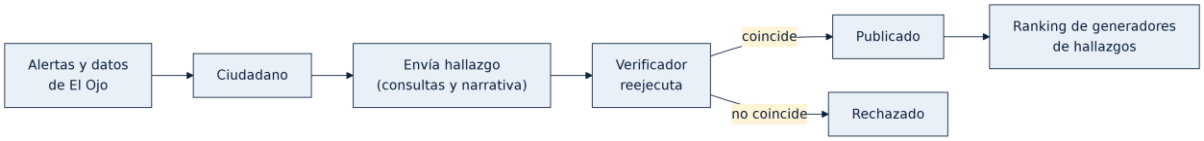

Figura 8. *Flujo de un hallazgo en la fase 1.*

Alcance: las cinco ediciones que ya están montadas, sólo con las vistas existentes y sin billeteras ni cadena de bloques. **Decisiones habilitantes:** algunas se toman desde ya para no tener que rehacer nada después. El hash se calcula de la misma forma en que se calculará on-chain, cada hallazgo guarda la versión exacta de la base, el repositorio se prepara para publicarse con licencia AGPL-3.0 (Free Software Foundation, 2007) al inicio de la fase 2 y la narrativa sigue una plantilla fija.

9.2 Fase 2 — Canalización

Objetivo: probar que los ciudadanos, las ONG, las entidades de gobierno y las entidades privadas, entre otros, están dispuestos a pagar por hallazgos que mejoren la acción del Estado y la calidad de su gasto. El Ojo recibe las donaciones, las atribuye a hallazgos concretos en función de lo expresado por cada donante y las liquida periódicamente a sus autores, reteniendo una comisión. Al inicio de esta fase se publican el código y las bases con licencia AGPL-3.0 (sección 8.4).

Elemento	Diseño
Entrada	Botón «recompensar este hallazgo» en cada hallazgo verificado
Medios de pago	Fiat (procesador de pagos) y USDC a una billetera de El Ojo
Atribución	Cada donación queda registrada contra un hallazgo en un libro público
Fondo común	Donaciones no dirigidas, repartidas mensualmente según el ranking
Comisión	Porcentaje fijo y publicado sobre cada donación
Liquidación	Mensual, a cuenta bancaria o billetera, con umbral mínimo
Transparencia	Libro de donaciones y liquidaciones con totales conciliables

Figura 9. *Flujo de donaciones en la fase 2.*

Limitación asumida: en esta fase, donante y generador de hallazgos deben confiar en El Ojo como custodio. El libro público atenúa esa necesidad de confianza y la fase 3 la elimina.

9.3 Fase 3 — Billeteras vinculadas

Objetivo: eliminar la custodia. Cada generador de hallazgos vincula una billetera de Base a su perfil y las donaciones a un hallazgo específico le llegan directamente, en USDC sobre esa red (Base, s. f.). La comisión se separa dentro del mismo contrato, de modo que El Ojo nunca toca los fondos del generador de hallazgos. La identidad se resuelve con Sign-In with Ethereum (Chang et al., 2021) y la autoría, con firmas de datos estructurados (Bloemen et al., 2017). Pese a sus nombres, ninguno de los dos estándares obliga a operar en la red principal de Ethereum ni tiene costo por sí mismo: ambos son firmas que la misma billetera de Base genera fuera de la cadena, y sólo se paga gas cuando un contrato registra o comprueba una firma, algo que en Base cuesta centavos (sección 7.6).

Componente	Diseño
Identidad	Inicio de sesión con la billetera (Sign-In with Ethereum, EIP-4361)
Firma del hallazgo	El autor firma h(H) (EIP-712), lo que prueba la autoría sin custodia de llaves
`DonacionHallazgo`	Recibe USDC para un hallazgo y reparte comisión y monto al autor en la misma transacción
Membresía	1 USDC mensual para publicar hallazgos elegibles a donaciones; filtro anti-spam y anti-Sybil
Registro de autoría	Mapa hallazgo → billetera del autor, publicado tras la verificación y respaldado por la firma del autor
Rampa de entrada	Rampas de entrada (on-ramp) y billeteras embebidas para donantes sin cripto

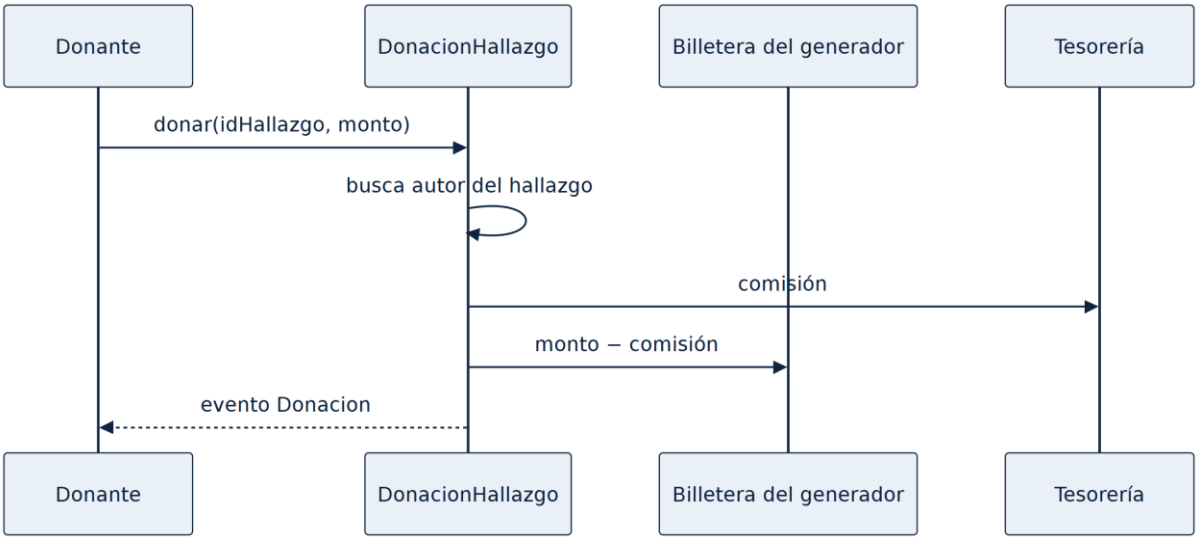

Figura 10. *Secuencia de una donación directa.*

```
function donar(bytes32 idHallazgo, uint256 monto) external;
// En F3 sólo El Ojo registra autores, y sólo con la firma EIP-712 del autor
function registrarAutor(bytes32 idHallazgo, address autor,
                        bytes calldata firmaAutor) external;
function pagarMembresia(uint256 meses) external;
event Donacion(bytes32 indexed idHallazgo, address indexed donante,
               uint256 monto, uint256 comision);
```

Figura 11. *Interfaz mínima de DonacionHallazgo (Solidity).*

Qué sigue centralizado: El Ojo verifica y registra la autoría, aunque no puede asignar un hallazgo a una billetera distinta de la que lo firmó. Los permisos de administración quedan bajo un multisig 2-de-3 desde el primer despliegue en mainnet.

9.4 Fase 4 — Verificabilidad on-chain

Objetivo: que nadie tenga que confiar en El Ojo para saber si un hallazgo es válido. Para eso se despliegan los registros on-chain y la red de verificadores descritos en las secciones 5 y 6, y las atestaciones se registran con Ethereum Attestation Service (s. f.).

Registro	Qué guarda	Quién escribe
`RegistroVersiones`	Por país y versión: raíz Merkle, manifiesto y versión anterior	Publicador autorizado por el multisig

Registro	Qué guarda	Quién escribe
`RegistroHallazgos`	Hash del hallazgo, autor, versiones referenciadas, estado	El autor, firmando con su billetera
Atestaciones	Veredicto de cada verificador por hallazgo y versión (EAS)	Verificadores con stake

9.5 Fase 5 — Descentralización plena

Objetivo: que El Ojo siga funcionando aunque su fundador y su infraestructura desaparezcan. Los datos pasan a Arweave (sección 5.4), la tesorería y los parámetros quedan bajo gobernanza (sección 8) y los contratos se traspasan a una organización autónoma descentralizada, o DAO (Hassan y De Filippi, 2021). En esta fase se habilitan también el matching cuadrático y el financiamiento retroactivo (sección 7.5). Se la llama descentralización plena porque es la última etapa de la descentralización progresiva, aquella en que la comunidad pasa a ser dueña del protocolo (Walden, 2020), y porque corresponde a lo que Ostrom (1990) describe como el autogobierno de un recurso común por quienes lo usan.

9.6 Criterios de paso

Se avanza de fase cuando se cumplen todos sus criterios. Los umbrales son iniciales y se ajustan con datos reales.

Paso	Criterios
F1 → F2	50 hallazgos verificados de al menos 20 autores distintos; verificador estable; reverificación automática en cada publicación de base; código y bases listos para publicarse con licencia AGPL-3.0
F2 → F3	Donaciones recurrentes durante 3 meses; al menos 30 % de hallazgos con alguna donación; comisión y umbrales calibrados
F3 → F4	`DonacionHallazgo` auditado y operando en mainnet; mayoría de donaciones vía billetera; multisig operativo
F4 → F5	Exportación determinista probada en CI; al menos 5 verificadores independientes; una disputa resuelta en testnet; commit-reveal, asignación aleatoria y hallazgos trampa operativos; mecanismo de voto definido
F5 completa	Datos históricos en Arweave, previo análisis de datos personales; contratos bajo gobernanza con timelock; fundador sin privilegios

10. Riesgos

El mayor riesgo no es técnico sino de adopción, esto es, que no haya suficientes generadores de hallazgos o donantes. Por ello, las fases 1 y 2 existen para medirlo antes de invertir en contratos.

Riesgo	Tipo	Fase	Mitigación
Pocos generadores de hallazgos o donantes	Adopción	F1–F2	Criterios de paso medibles; comunidad universitaria; no avanzar sin tracción
Hallazgo correcto pero engañoso	Calidad	Todas	Plantilla de narrativa; consultas públicas y reejecutables; hallazgos de réplica
Lectura equivocada de la ejecución presupuestaria	Calidad	Todas	Comparar contra el presupuesto vigente y no sólo contra el inicial; considerar la estacionalidad normal del gasto; normalizar unidades (el presupuesto puede venir en miles y la ejecución en pesos)
Acusaciones implícitas contra personas	Reputacional y legal	Todas	El Ojo nunca denuncia: cada generador de hallazgos responde personalmente por su narrativa, incluso ante los tribunales; principio «Mirar no es acusar»; narrativa en formato de candidato a revisar; derecho de réplica para las personas mencionadas
Exportación no determinista	Técnico	F4–F5	Versiones fijas, orden canónico y prueba en CI de que dos exportaciones son idénticas
Fuente oficial caída o incompleta	Datos	Todas	Versión anterior conservada y etiquetada; el hallazgo queda fijado a su versión
Vulnerabilidad en contratos	Seguridad	F3–F5	Contratos mínimos, auditoría, testnet, límites de monto al inicio, pausa por multisig
Cuentas falsas y donaciones circulares	Económico	F3–F5	Membresía; matching cuadrático ponderado por reputación
Colusión de verificadores	Económico	F4–F5	Stake y slashing; asignación aleatoria; commit-reveal; hallazgos trampa; disputas por reejecución
Captura de la gobernanza	Gobernanza	F5	Timelock; reputación no transferible en el voto; parámetros acotados por contrato
Volatilidad del precio de AR	Costo	F5	Deltas por partición; compra de almacenamiento en momentos de precio bajo
Consulta pesada que degrada el sitio	Operativo	F1+	Verificador aislado del sitio, con límites de memoria, tiempo y filas
Dependencia de Base	Infraestructura	F3–F5	Base opera hoy con un secuenciador único a cargo de Coinbase (L2BEAT, s. f.); los anclajes son baratos de replicar en otra red o directamente en Ethereum
Congelamiento de USDC	Económico	F3–F5	El emisor de USDC puede bloquear direcciones; aceptar más de un activo estable y no acumular saldos en los contratos
Datos personales en almacenamiento permanente	Legal	F4–F5	En cadena sólo hashes; minimización y seudonimización antes de exportar; análisis previo al paso a Arweave
Regulación de la canalización de fondos	Legal	F2	Procesador de pagos regulado; montos acotados y rendición pública
Documento narrativo malicioso o engañoso	Seguridad y calidad	Todas	Tamaño máximo; el sitio muestra una versión saneada y guarda el original sólo para verificar su hash; responsabilidad del autor

11. Conclusión

El Ojo ya resolvió una de las partes más costosas de la supervisión ciudadana, que es reunir, normalizar y cruzar los datos presupuestarios, de compras, de lobby y de probidad que publican los Estados. El protocolo descrito aquí agrega lo que falta para que esa supervisión escale y se haga permanente en el tiempo a través del establecimiento de incentivos adecuados: una unidad de trabajo verificable, una recompensa directa para quien la produce y una capa de confianza que no depende de nadie en particular.

La secuencia es deliberadamente gradual. Cada fase se justifica por la evidencia que deja la anterior y reduce la confianza que es necesario depositar en El Ojo y en su fundador. Si el camino se completa, El Ojo dejará de ser un proyecto personal para convertirse en infraestructura pública de supervisión democrática, gobernada por quienes la usan. Y si los datos de las primeras fases muestran que no hay suficientes generadores de hallazgos o donantes, el diseño permite detenerse a tiempo, antes de haber invertido en contratos que nadie usaría.

La idea central es establecer un mecanismo que, con los incentivos adecuados, mejore la supervisión ciudadana sobre el accionar del Estado, desde el eventual mal uso de los recursos hasta el cumplimiento del presupuesto y la calidad del gasto fiscal, y contribuya así a tener mejores Estados y, con ello, un mayor bienestar para la población.

Referencias

Las referencias se agrupan por tema y siguen las normas APA (7.ª edición). Las fuentes de datos de cada edición se documentan por separado en el repositorio del proyecto.

Transparencia, control ciudadano, corrupción y gasto público

Alt, J. E. y Lassen, D. D. (2006). Fiscal transparency, political parties, and debt in OECD countries. *European Economic Review, 50*(6), 1403–1439. https://doi.org/10.1016/j.euroecorev.2005.04.001

de Renzio, P. y Wehner, J. (2017). The impacts of fiscal openness. *The World Bank Research Observer, 32*(2), 185–210. https://doi.org/10.1093/wbro/lkx004

Fazekas, M. y Kocsis, G. (2020). Uncovering high-level corruption: Cross-national objective corruption risk indicators using public procurement data. *British Journal of Political Science, 50*(1), 155–164. https://doi.org/10.1017/S0007123417000461

Fazekas, M., Tóth, I. J. y King, L. P. (2016). An objective corruption risk index using public procurement data. *European Journal on Criminal Policy and Research, 22*(3), 369–397. https://doi.org/10.1007/s10610-016-9308-z

Ferraz, C. y Finan, F. (2008). Exposing corrupt politicians: The effects of Brazil's publicly released audits on electoral outcomes. *The Quarterly Journal of Economics, 123*(2), 703–745. https://doi.org/10.1162/qjec.2008.123.2.703

Fondo Monetario Internacional (2019). *The fiscal transparency code*. FMI. https://www.imf.org/external/np/fad/trans/Code2019.pdf

Fox, J. A. (2015). Social accountability: What does the evidence really say? *World Development, 72*, 346–361. https://doi.org/10.1016/j.worlddev.2015.03.011

Fung, A., Graham, M. y Weil, D. (2007). *Full disclosure: The perils and promise of transparency*. Cambridge University Press.

Global Initiative for Fiscal Transparency (2012). *High-level principles on fiscal transparency, participation and accountability*. https://fiscaltransparency.net/gift-principles/

International Budget Partnership (2024). *Open Budget Survey 2023*. https://internationalbudget.org/open-budget-survey-2023/

Khagram, S., Fung, A. y de Renzio, P. (Eds.) (2013). *Open budgets: The political economy of transparency, participation, and accountability*. Brookings Institution Press.

O'Donnell, G. (1998). Horizontal accountability in new democracies. *Journal of Democracy, 9*(3), 112–126. https://doi.org/10.1353/jod.1998.0051

Olken, B. A. (2007). Monitoring corruption: Evidence from a field experiment in Indonesia. *Journal of Political Economy, 115*(2), 200–249. https://doi.org/10.1086/517935

Olson, M. (1965). *The logic of collective action: Public goods and the theory of groups*. Harvard University Press.

Open Contracting Partnership y Development Gateway (2016). *Red flags for integrity: Giving the green light to open data solutions*. https://www.open-contracting.org/wp-content/uploads/2016/11/OCP2016-Red-flags-for-integrityshared.pdf

PEFA Secretariat (2016). *Framework for assessing public financial management*. https://www.pefa.org/resources/pefa-2016-framework

Peruzzotti, E. y Smulovitz, C. (Eds.) (2006). *Enforcing the rule of law: Social accountability in the new Latin American democracies*. University of Pittsburgh Press.

Smulovitz, C. y Peruzzotti, E. (2000). Societal accountability in Latin America. *Journal of Democracy, 11*(4), 147–158. https://doi.org/10.1353/jod.2000.0087

Reproducibilidad, criptografía y verificación

Blum, M. (1983). Coin flipping by telephone: A protocol for solving impossible problems. *ACM SIGACT News, 15*(1), 23–27. https://doi.org/10.1145/1008908.1008911

Buneman, P., Khanna, S. y Tan, W.-C. (2001). Why and where: A characterization of data provenance. En J. Van den Bussche y V. Vianu (Eds.), *Database Theory — ICDT 2001* (pp. 316–330). Springer. https://doi.org/10.1007/3-540-44503-X_20

Douceur, J. R. (2002). The Sybil attack. En P. Druschel, F. Kaashoek y A. Rowstron (Eds.), *Peer-to-Peer Systems* (pp. 251–260). Springer. https://doi.org/10.1007/3-540-45748-8_24

Haber, S. y Stornetta, W. S. (1991). How to time-stamp a digital document. *Journal of Cryptology, 3*(2), 99–111. https://doi.org/10.1007/BF00196791

Kalodner, H., Goldfeder, S., Chen, X., Weinberg, S. M. y Felten, E. W. (2018). Arbitrum: Scalable, private smart contracts. En *Proceedings of the 27th USENIX Security Symposium* (pp. 1353–1370). USENIX Association.

Lesaege, C., Ast, F. y George, W. (2019). *Kleros: Short paper v1.0.7*. https://kleros.io/assets/whitepaper.pdf

Li, X., Weng, C., Xu, Y., Wang, X. y Rogers, J. (2023). ZKSQL: Verifiable and efficient query evaluation with zero-knowledge proofs. *Proceedings of the VLDB Endowment, 16*(8), 1804–1816. https://doi.org/10.14778/3594512.3594513

Luu, L., Teutsch, J., Kulkarni, R. y Saxena, P. (2015). Demystifying incentives in the consensus computer. En *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (pp. 706–719). ACM. https://doi.org/10.1145/2810103.2813659

Merkle, R. C. (1988). A digital signature based on a conventional encryption function. En C. Pomerance (Ed.), *Advances in Cryptology — CRYPTO '87* (pp. 369–378). Springer. https://doi.org/10.1007/3-540-48184-2_32

Micali, S., Rabin, M. y Vadhan, S. (1999). Verifiable random functions. En *40th Annual Symposium on Foundations of Computer Science* (pp. 120–130). IEEE. https://doi.org/10.1109/SFFCS.1999.814584

Peng, R. D. (2011). Reproducible research in computational science. *Science, 334*(6060), 1226–1227. https://doi.org/10.1126/science.1213847

Raasveldt, M. y Mühleisen, H. (2019). DuckDB: An embeddable analytical database. En *Proceedings of the 2019 International Conference on Management of Data* (pp. 1981–1984). ACM. https://doi.org/10.1145/3299869.3320212

Schelling, T. C. (1960). *The strategy of conflict*. Harvard University Press.

Teutsch, J. y Reitwießner, C. (2019). *A scalable verification solution for blockchains* (arXiv:1908.04756). https://arxiv.org/abs/1908.04756

Zhang, Y., Genkin, D., Katz, J., Papadopoulos, D. y Papamanthou, C. (2017). vSQL: Verifying arbitrary SQL queries over dynamic outsourced databases. En *2017 IEEE Symposium on Security and Privacy* (pp. 863–880). IEEE. https://doi.org/10.1109/SP.2017.43

Bienes públicos, gobernanza y cadenas de bloques

Buterin, V. (2014). *Ethereum: A next-generation smart contract and decentralized application platform*. https://ethereum.org/whitepaper/

Buterin, V. (2021). *Moving beyond coin voting governance*. https://vitalik.eth.limo/general/2021/08/16/voting3.html

Buterin, V., Hitzig, Z. y Weyl, E. G. (2019). A flexible design for funding public goods. *Management Science, 65*(11), 5171–5187. https://doi.org/10.1287/mnsc.2019.3337

Eghbal, N. (2020). *Working in public: The making and maintenance of open source software*. Stripe Press.

Hassan, S. y De Filippi, P. (2021). Decentralized autonomous organization. *Internet Policy Review, 10*(2). https://doi.org/10.14763/2021.2.1556

Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system*. https://bitcoin.org/bitcoin.pdf

Optimism (2021). *Retroactive public goods funding*. Optimism PBC Blog. https://medium.com/ethereum-optimism/retroactive-public-goods-funding-33c9b7d00f0c

Ostrom, E. (1990). *Governing the commons: The evolution of institutions for collective action*. Cambridge University Press.

Walden, J. (2020). *Progressive decentralization: A playbook for building crypto applications*. a16z crypto. https://a16zcrypto.com/posts/article/progressive-decentralization-crypto-product-management/

Weyl, E. G., Ohlhaver, P. y Buterin, V. (2022). *Decentralized society: Finding Web3's soul*. SSRN. https://doi.org/10.2139/ssrn.4105763

Infraestructura y estándares técnicos

Apache Software Foundation (s. f.). *Apache Parquet*. https://parquet.apache.org

Arweave (s. f.). *Arweave Fees: calculadora de tarifas de almacenamiento*. https://ar-fees.arweave.net

Base (s. f.). *Base: red de capa 2 de Ethereum*. https://base.org

Bloemen, R., Logvinov, L. y Evans, J. (2017). *EIP-712: Typed structured data hashing and signing*. Ethereum Improvement Proposals. https://eips.ethereum.org/EIPS/eip-712

Chang, W., Rocco, G., Millegan, B., Johnson, N. y Terbu, O. (2021). *EIP-4361: Sign-In with Ethereum*. Ethereum Improvement Proposals. https://eips.ethereum.org/EIPS/eip-4361

Ethereum Attestation Service (s. f.). *Ethereum Attestation Service (EAS)*. https://attest.org

Free Software Foundation (2007). *GNU Affero General Public License, versión 3*. https://www.gnu.org/licenses/agpl-3.0.html

Google Cloud (s. f.). *Cloud Storage pricing*. https://cloud.google.com/storage/pricing

International Organization for Standardization (2005). *ISO 19005-1:2005. Document management — Electronic document file format for long-term preservation — Part 1: Use of PDF 1.4 (PDF/A-1)*. ISO.

L2BEAT (s. f.). *Base: análisis de riesgos*. https://l2beat.com/scaling/projects/base

National Institute of Standards and Technology (2015). *Secure Hash Standard (SHS)* (FIPS PUB 180-4). https://doi.org/10.6028/NIST.FIPS.180-4

Safe (s. f.). *Safe: billeteras multifirma*. https://safe.global

Williams, S., Diordiiev, V., Berman, L., Raybould, I. y Uemlianin, I. (2019). *Arweave: A protocol for economically sustainable information permanence* [Yellow paper]. https://www.arweave.org/yellow-paper.pdf

Normativa

Chile. Ley 19.886, de bases sobre contratos administrativos de suministro y prestación de servicios (2003).

Chile. Ley 20.285, sobre acceso a la información pública (2008).

Chile. Ley 20.730, que regula el lobby y las gestiones que representen intereses particulares ante las autoridades y funcionarios (2014).

Chile. Ley 20.880, sobre probidad en la función pública y prevención de los conflictos de intereses (2016).

Chile. Ley 21.634, que moderniza la Ley 19.886 y otras leyes, para mejorar la calidad del gasto público, aumentar los estándares de probidad y transparencia e introducir principios de economía circular en las compras del Estado (2023).

Argentina. Ley 27.275, de derecho de acceso a la información pública (2016).

Colombia. Ley 1712 de 2014, de transparencia y del derecho de acceso a la información pública nacional.

Perú. Ley 27806, de transparencia y acceso a la información pública (2002).

Uruguay. Ley 18.381, sobre el derecho de acceso a la información pública (2008).

Anexo A. Glosario

Término	Definición
Alerta	Criterio de El Ojo que señala un candidato a revisar (desviación presupuestaria, contratación express); punto de partida, no hallazgo
Atestación	Declaración firmada on-chain de que un hallazgo reproduce (o no) sobre una versión
Billetera (wallet)	Aplicación que guarda las llaves criptográficas de una persona y le permite firmar mensajes y transacciones
Calidad del gasto	Grado en que el gasto público logra sus objetivos con eficiencia, medido, por ejemplo, frente a instituciones comparables
Commit-reveal	Esquema de compromiso en dos pasos: primero se publica el hash del veredicto combinado con un valor secreto y, después, el veredicto y el valor, para que nadie pueda copiar el ajeno
Cumplimiento presupuestario	Grado en que la ejecución del gasto se ajusta al presupuesto vigente, en monto y en calendario
DAO	Organización autónoma descentralizada: comunidad que se gobierna mediante reglas codificadas en contratos inteligentes
Descentralización plena	Última fase de la hoja de ruta: datos en almacenamiento permanente, contratos y tesorería bajo gobernanza comunitaria y fundador sin privilegios
Documento narrativo	Archivo opcional, por ejemplo un PDF, en que el autor desarrolla la narrativa; su hash forma parte del hallazgo
Escrow	Retención de fondos en un contrato hasta que se cumple una condición, como reunir *k* atestaciones coincidentes
Generador de hallazgos	Persona que publica hallazgos verificables
Hallazgo trampa	Hallazgo con hash incorrecto conocido que el protocolo inyecta para detectar verificadores que aprueban sin ejecutar
Hallazgo verificable	Conjunto de una o más consultas reproducibles sobre versiones fechadas de la base, cuyos resultados cualquiera puede recalcular
Manifiesto	Documento que lista las particiones de una versión, sus hashes y su ubicación
Matching cuadrático	Reparto de un fondo que premia el número de donantes distintos más que el monto
Multisig	Billetera que requiere varias firmas (p. ej., 2 de 3) para operar
Raíz Merkle	Hash único que resume un conjunto de hashes; cambia si cambia cualquiera de ellos
Secuenciador	Nodo que ordena las transacciones de una red de capa 2 antes de publicarlas en Ethereum
Stake / slashing	Depósito en garantía y su pérdida por conducta demostrada como falsa
Timelock	Retraso obligatorio entre la aprobación de un cambio y su ejecución
Verificador	Participante con stake que reejecuta hallazgos y emite atestaciones
Versión de base	Publicación fechada de la base de un país, identificada por su raíz Merkle
VRF	Función de aleatoriedad verificable; se usa para asignar verificadores
zk-SQL	Prueba criptográfica de que una consulta SQL produjo cierto resultado, que puede validarse sin reejecutarla