RootCause Blockchain Security

Vigilar un sistema on-chain sin poder tocarlo: watch-only, local y sin claves.

- Inventario multi-chain, postura de control e incidentes **con causa raíz**.
- 13 controles · 22 detecciones · 15 indicadores de inteligencia.
- Aplicación de escritorio para Windows. **0 dependencias externas.**
- Nunca pide una semilla. Nunca firma. Nunca mueve fondos.

El problema no son las claves

Puedes tener la criptografía perfecta y perderlo todo igual.

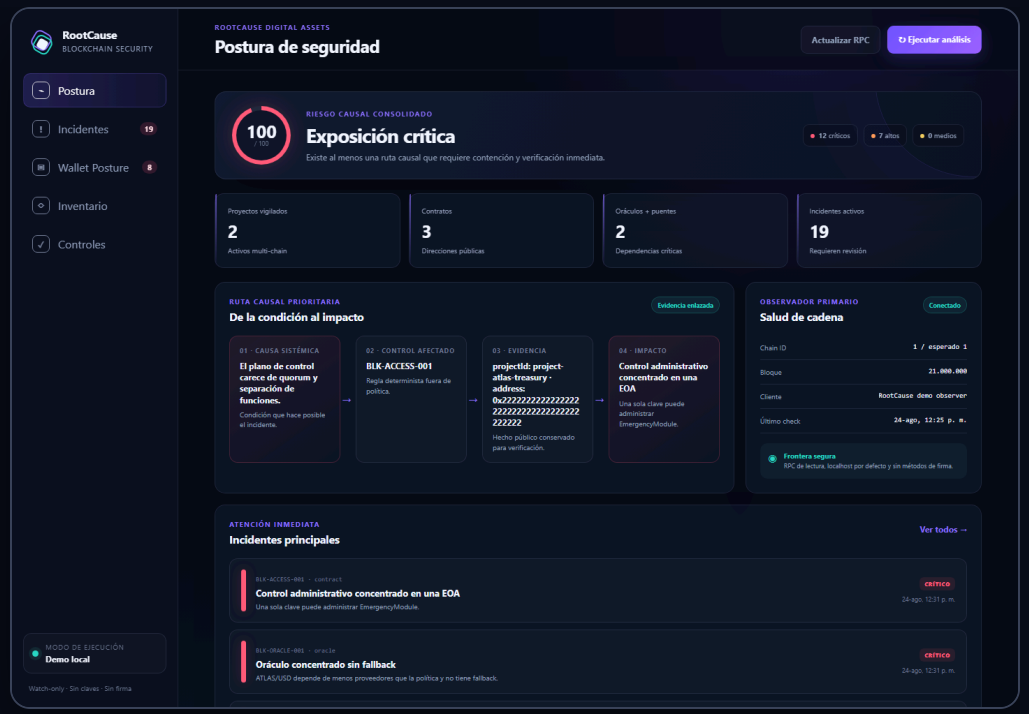
- Una EOA olvidada seguía siendo `owner` del proxy que actualiza el contrato.
- Un oráculo dejó de actualizarse y nadie miró el `heartbeat` hasta la liquidación.
- Un puente con quorum 2-de-3 tenía **los tres firmantes en la misma nube**.

> Ninguna de las tres es una falla criptográfica. Las tres son **fallas de control**, y todas eran visibles desde fuera.

Qué es: una consola local, watch-only

Inventario, postura y evidencia. Todo en la máquina del operador.

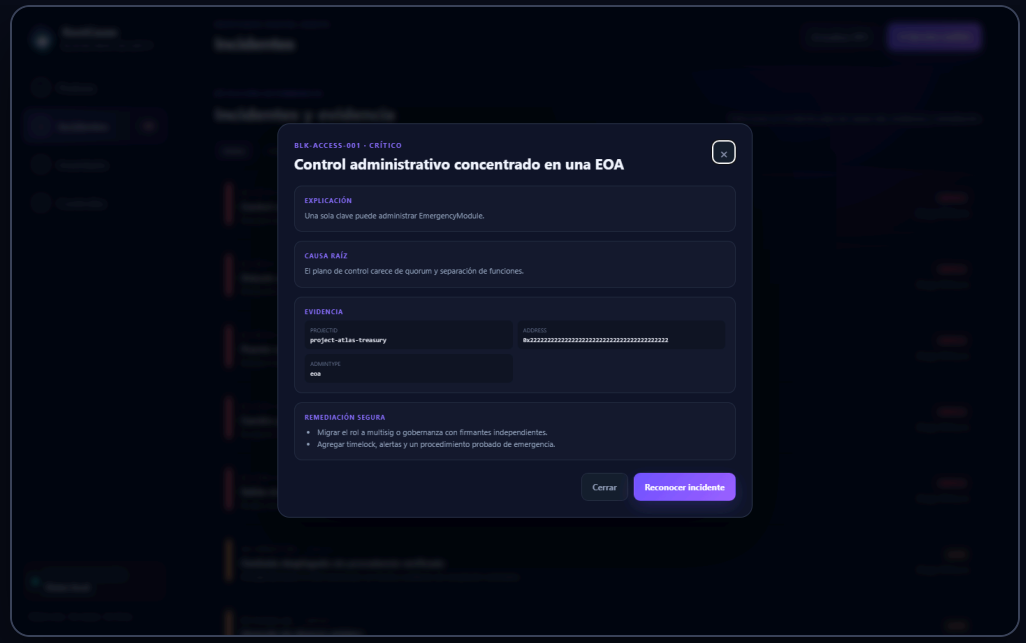
LO QUE TRAE DENTRO	CUÁNTO
Controles de defensa, del bytecode a la actividad de wallets	**13**
Detecciones deterministas `BLK-*`	**22**
Indicadores investigativos `INT-*`	**15**
Escenarios sintéticos con su resultado esperado	**10**
Pruebas automatizadas que CI ejecuta en cada cambio	**144**
Dependencias externas	**0**



De la alerta suelta a la causa raíz

Un incidente que no dice qué arreglar es ruido con formato.

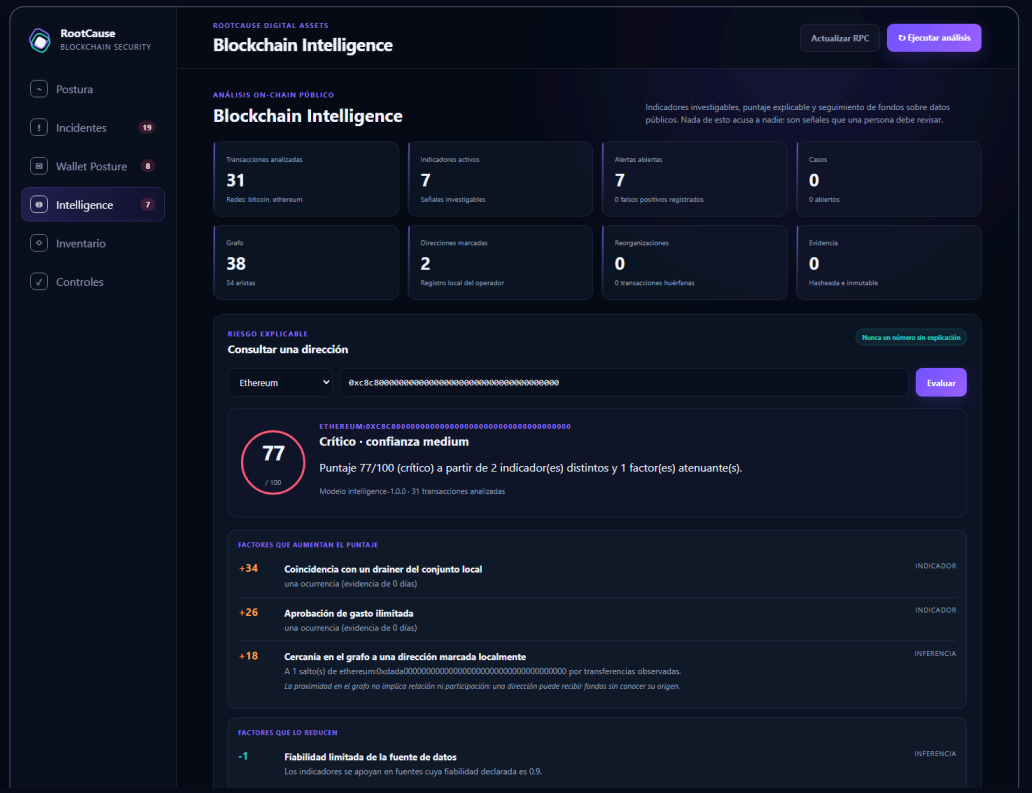
- Cada hallazgo abre con **explicación, causa raíz, evidencia y remediación**.
- Las reglas son **deterministas**: el mismo estado produce el mismo veredicto.
- Cada código `BLK-*` pertenece a un control del catálogo. Un gate lo obliga.
- La remediación es un **runbook para una persona**, no un botón.



Wallets e inteligencia: señalar no es acusar

Dos catálogos separados a propósito, y no se mezclan nunca.

	POSTURA `BLK-*`	INTELIGENCIA `INT-*`
Qué afirma	Un control **está mal puesto**	Hay algo **que conviene revisar**
Cómo se decide	Regla determinista contra política	Indicador con umbral calibrable
Qué produce	Un incidente con remediación	Una señal con su explicación
Lo que **nunca** hace	—	Atribuir identidad, inferir intención, bloquear fondos



Seis principios que se comprueban solos

Una promesa de seguridad que nadie verifica es material de marketing.

1. Nunca pide ni guarda semillas, claves, keystores ni transacciones sin publicar.
2. No firma, no construye y no transmite transacciones.
3. El JSON-RPC usa una **allowlist estricta de métodos de lectura**.
4. Acepta `localhost`; un destino remoto exige habilitación explícita.
5. Las reglas son deterministas. **La IA es opcional y no tiene autoridad.**
6. Toda acción privilegiada exige runbook y aprobación humana separada.

> Los seis se verifican **arrancando la aplicación real** y atacándola: **51 invariantes** en `scripts/check-security-claims.js`.

Demo en vivo: 90 segundos de terminal

Dos comandos. Sin red, sin instalar nada, sin datos reales.

COMANDO	LO QUE IMPRIME
`pnpm check:local-only`	Claim solo-local verificado: cero dependencias, cero orígenes remotos, cero proveedores alojados.
`pnpm check:security`	Invariantes de seguridad verificados: 51 comprobaciones.

> Esta tabla se rellena **ejecutando los dos comandos** al generar las diapositivas. Si la terminal falla delante del público, señálala y sigue.

Lo que nunca hará, y cómo se adopta

La lista de lo que no hace es parte del producto, no una nota al pie.

NUNCA	SÍ, HOY
Custodiar claves ni firmar	Vigilar controles y explicar por qué fallan
Sustituir una auditoría o una verificación formal	Detectar controles mal puestos, en continuo
Atribuir identidad ni bloquear fondos	Señalar lo que conviene investigar, con evidencia
Prometer soporte de toda cadena	EVM implementado; el resto, por eventos normalizados

- **Empezar:** descargar el instalador de Windows, o `pnpm start` para el modo demo.
- **Antes de datos reales:** la lista de comprobación de seguridad. En ese orden.
- **Licencia MIT.** Versión 0.3.0. Sin telemetría, sin cuenta, sin nube.