

Seminario Socrático #13 — temas para preparar

miércoles 26 de agosto de 2026 · resumen de los temas del temario · BitDevs Córdoba

Este resumen es para llegar al seminario sabiendo de qué se va a hablar. Cada tema abre en lenguaje natural —para cualquiera— y sigue con el detalle técnico en el recuadro. Cada uno lleva su fuente primaria, por si querés ir al original.

Explicaciones preparadas por Maxwell Labib (agente automatizado) y revisadas antes de publicar.

1. ¿Qué tan confiable es la heurística multi-input para agrupar direcciones de Bitcoin en contextos policiales y judiciales?

How Reliable Is the Multi-Input Heuristic for Bitcoin Address Clustering in Law Enforcement Contexts? — fuente

Cuando la policía o una empresa de análisis dice «estas 400 direcciones son del mismo dueño», casi siempre lo dice por una sola regla: si dos direcciones aparecen juntas gastando en la misma transacción, quien firmó tenía las llaves de las dos, así que son de la misma persona. Se llama heurística multi-input y es el cimiento de todo el forense de blockchain. Es razonable, es vieja, y casi nunca se midió contra la verdad, porque la verdad no existe públicamente: nadie sabe de quién es cada dirección.

Este paper consiguió esa verdad. Siete exchanges europeos entregaron, por obligación legal de reporte, el mapeo de qué direcciones son suyas, y ahí se pudo medir cuánto acierta la heurística. El resultado tiene dos caras. Mirada de una forma parece sólida. Mirada de la forma en que se usa realmente en una causa —«este cluster entero es del acusado»— acierta bastante menos de la mitad, y hay servicios donde falla casi por completo. Los autores lo dicen sin vueltas: cuando un cluster sostiene una sospecha, un embargo de bienes o una prueba en juicio, fiscales y jueces tienen que saber que la fiabilidad depende de qué métrica se mire y de qué entidad se trate. Es el tema del temario que más rápido baja a una sala de tribunales.

Framework reusable con nueve métricas: precision/recall/F1 pairwise, precision/recall/F1 per-wallet, NMI, aNMI y average error rate (AER). Ground truth: mapeos dirección→entidad de siete crypto asset service providers europeos entregados bajo obligación legal de reporte, con corte el 23/06/2023 (altura 795.357). El corpus de cadena son todas las transacciones hasta esa altura desde un full node: 854.144.149 transacciones con 1.105.300.685 direcciones de input distintas, media de 2,22 inputs por transacción. Restringiendo la evaluación a las direcciones reportadas, la MIH luce fuerte a nivel dataset: cero mergers entre servicios reportados y recall 0,71 en recuperar pares de direcciones del mismo servicio.

El problema es que ese número está traccionado por un solo servicio grande e ignora las direcciones no etiquetadas que el cluster absorbe. Las métricas sobre clusters completos dan precisión 0,36 y recall 0,44 – o sea que los servicios quedan parcialmente recuperados o embebidos dentro de clusters más grandes. A nivel entidad la dispersión es brutal: Servicio 1 (el dominante) da per-wallet precision 0,950 y recall 0,730; Servicio 2 da recall 0,004 y F1 0,008; Servicio 3, de tamaño moderado, da recall y F1 esencialmente cero con AER 0,999; Servicio 5 da precisión 0,256 y recall 0,042. La conclusión de los autores es que «MIH performance is highly entity-dependent», con reconstrucción fuerte de un servicio dominante conviviendo con

recuperación casi nula de otros – y la bajada explícita al foro judicial: «prosecutors and judges must account for the heuristic's metric-dependent and entity-dependent reliability».

2. Aviso de generación de semillas en COLDCARD — vulnerabilidad de entropía

COLDCARD seed generation warning — entropy vulnerability — [fuente](#)

Una hardware wallet sirve para una sola cosa: generar y guardar una llave que nadie más pueda adivinar. Durante cinco años, las COLDCARD generaron esa llave con mucha menos aleatoriedad de la que decían. El dispositivo tiene un generador de azar físico y no lo estaba usando: un cambio de código de marzo de 2021 hizo que la semilla saliera de un generador de software, arrancado desde valores predecibles del propio equipo. El resultado es que en vez de 128 bits de entropía —el número que hace imposible adivinar— quedaron unos 40 en los modelos más viejos. Cuarenta bits sí se puede recorrer entero con una computadora.

Y acá está lo que lo vuelve fatal, y es propio de Bitcoin: el atacante no necesita saber quién sos ni qué aparato tenés. Genera todas las semillas posibles, deriva las direcciones de cada una y las cruza contra la cadena, que le dice gratis cuál tenía plata. El 30 de julio de 2026, un día después de nuestro seminario #12, un análisis comunitario del incidente reporta que alguien hizo exactamente eso y barrió unos 594 BTC en 25 minutos, sin tocar un solo dispositivo. Lo más importante para quien tenga una en esta sala: actualizar el firmware NO repara una semilla ya generada. Si nació débil, nace débil para siempre — hay que generar una nueva y mudar los fondos.

Causa raíz según Optech #416: «a 2021 code change that unintentionally routed seed generation to a software PRNG, initialized from predictable device-unique values, instead of the device's hardware RNG». El detalle de compilación —dos símbolos ``rng_get()`` con la misma firma en el binario, el guard ``#ifndef MICROPY_HW_ENABLE_RNG`` que no disparó porque la macro estaba definida en ``0``, y el linker resolviendo a Yasmarang, el PRNG no criptográfico de MicroPython— está documentado en el backgrounder técnico de Coinkite. Alcance verificado contra Optech: Mk3 con firmware $\geq 4.0.1$ (marzo 2021), Mk2 con 4.x, Mk4/Mk5 pre-5.6.0 y Q pre-1.5.0Q. Cifras: Mk3 sin datos suplementarios queda en «roughly 40 bits of effective entropy instead of the intended 128»; para Mk4/Mk5/Q, «only 32 bits of that entropy reach the PRNG state». Block verificó además que el problema alcanza a otros secretos generados en el equipo: claves de paper wallet y ephemeral seeds.

La explotación es enumeración pura: recorrer los 2^{40} estados del PRNG → derivar la seed de cada estado → derivar direcciones por BIP32 → cruzar contra el UTX0 set. No hay componente de red ni de supply chain; la cadena es el oráculo de verificación, gratis y permanente. Es el mismo patrón que Randstorm (2011-2015, ``SecureRandom()`` de JSBN + ``Math.random()``/MWC1616, 2^{32} efectivos) y Milk Sad (2017-2023, ``bx seed`` de Libbitcoin Explorer 3.0.0-3.6.0, MT19937 sembrado con 32 bits de reloj, CVE-2023-39910): PRNG con estado chico sembrado desde reloj o estado del dispositivo. Mitigaciones, según el advisory vía guión: ≥ 50 tiradas de dados al crear la seed original la dejan fuera de riesgo ($50 \times \log_2 6 = 2,585 \text{ bits} \approx 129 \text{ bits}$); una passphrase BIP-39 fuerte es una capa independiente que sobrevive al bug, aunque Coinkite igual recomienda migrar. El write-up comunitario reporta que de los 1.324 UTX0s barridos ninguno era multisig.

3. The Four Horsemen — una especificación unificada para los protocolos coinjoin de Bitcoin

The Four Horsemen — a unified spec for Bitcoin coinjoin protocols — [fuente](#)

Un coinjoin es una transacción donde varias personas juntan sus monedas en una sola operación para que desde afuera no se pueda decir quién le pagó a quién. El problema no es que no existan formas de hacerlo: es que existen cuatro, cada una con su implementación, su gente y sus defectos, y no se hablan entre sí. El usuario tiene que elegir una y quedarse con sus limitaciones.

Esta propuesta dice: dejemos de elegir. Un solo cliente que tenga los cuatro adentro y use el que convenga según el caso —cuánta plata movés, si estás pagando o solo mezclando, cuánta liquidez hay disponible en ese momento. El autor lo llama «un sistema operativo para protocolos de coinjoin». Dos advertencias honestas antes de venir al seminario: es un documento de diseño, no hay código; y es de una sola persona, sin nadie que le haya contestado todavía. Que Londres y Belo Horizonte lo hayan puesto en su temario habla de la agenda de la red, no de que el diseño haya sido revisado.

Los cuatro «jinetes» y los costos que el propio documento les asigna: ****WabiSabi****, credenciales anónimas verificadas con clave (KVAC), que permiten consolidar y gastar montos arbitrarios pero escalan mal —600-700 vBytes por usuario por transacción y ~100 sats de «dust» por usuario—, con el tamaño máximo de ronda limitado por la estabilidad de Tor; ****ZeroLink****, firmas ciegas chaumianas sobre denominaciones fijas, muy eficiente para remixear (~100 vBytes por usuario) pero genera «untouchable toxic change», con la ronda limitada por la liquidez disponible; ****JoinMarket****, modelo maker/taker donde el maker aporta liquidez pasiva y cobra, el taker paga las fees de minería, cuesta ~4.000 vBytes con 10 makers, y usa fidelity bonds contra la dilución sybil; y ****Payjoin****, coinjoin oportunista entre pagador y cobrador, donde el pagador subsidia los ~210 vBytes del input del receptor, y donde ambas partes son single point of failure. El argumento de integración es que se complementan: el usuario de WabiSabi remixa barato con ZeroLink, JoinMarket le da liquidez al que mueve mucho, y Payjoin es la vía rápida cuando lo que hay es un pago real. Requisito transversal declarado: «Wallet synchronization must be performed using compact block filters or the user's full node by default».

El estado del hilo importa tanto como el contenido, y hay que decirlo al presentarlo: al día de hoy son tres posts, los tres del mismo autor. El segundo introduce una taxonomía de roles (Rangers, Knights, Peasants) y nombres de planetas para configuraciones de nodo, más «razas» que clasifican patrones de comportamiento según el historial de transacciones — es andamiaje conceptual, no especificación. El tercero es un placeholder reservado para specs de servidor. No hay implementación de referencia, no hay test vectors, y no hay una sola objeción registrada. Para formato socrático eso es material: la pregunta no es si el diseño cierra, sino por qué un intento de unificar cuatro comunidades que no se hablan no generó ni una respuesta.

4. Señalización obligatoria de BIP-110: un chain split sin apoyo minero

BIP-110 mandatory signaling: a chain split with zero miner support — [fuente](#)

BIP-110 era una propuesta para limitar por un año los datos arbitrarios que se guardan en la cadena. Para activarse necesitaba que el 55% de los mineros la señalizara, y nunca se acercó. Lo distinto de esta propuesta es cómo pensaba forzar la cosa: a partir de cierto bloque —el 961.632, el 8 de agosto de 2026— los nodos que la corrieran iban a rechazar como inválido todo bloque que no la señalizara. La idea era que los mineros no tuvieran más remedio que subirse.

No pasó. Los mineros ignoraron el bit, así que el primer bloque de la ventana ya fue inválido para los nodos que aplicaban la regla, y esos nodos se quedaron sin cadena a la que agarrarse. Alguien minó cuatro bloques del lado de BIP-110 y ahí se frenó, mientras el resto de la red siguió como si nada. Diez días después la rama que quedó del otro lado tenía 533 bloques y la propuesta se cerró formalmente. Lo

interesante para discutir no es si limitar datos está bien o mal: es que una propuesta que se hizo campaña entre operadores de nodos eligió un mecanismo donde solo cuentan los mineros. Se quedó sin el electorado al que le hablaba.

El despliegue era un BIP9 modificado: bit 4, umbral 1109/2016 = 55% en vez de 95%, `NO\_TIMEOUT`, ventana de señalización obligatoria 961.632-963.647 —exactamente un periodo de retarget, porque $961.632 = 2016 \times 477$ —, lock-in a más tardar en 963.648, ACTIVE en 965.664 y duración de 52.416 bloques (26 periodos $\approx$ un año). Durante la ventana, los bloques que no señalizan bit 4 son rechazados como inválidos. La señalización medida contra la cadena dio 0,0%: cero bloques con bit 4 en ~140 muestreados de forma distribuida, lo que descarta con ~95% de confianza cualquier tasa $\geq 2,1\%$. El desenlace lo confirma el BIP editor al mergear bips#2245: «There are currently two consensus valid chaintips that split after height 961,631. One has four blocks, the other has 533 blocks. The BIP110 deployment attempt was decisively rejected by the Bitcoin network».

Dos mecanismos explican por qué no había vuelta atrás. Primero, la irreversibilidad: para el enforcer, el bloque 961.632 quedó inválido para siempre, y con él todos sus descendientes; aunque el 100% del hashrate empezara a señalar hoy, reconverger exigiría un reorg de cientos de bloques o abandonar la regla. La decisión no se tomó a lo largo de la ventana de dos semanas: se tomó en un solo bloque. Segundo, la trampa de dificultad: como la ventana obligatoria es exactamente un periodo de retarget, la rama minoritaria tiene que producir 2.016 bloques a la dificultad completa de la red antes de poder ajustarla a la baja — con una fracción $*x*$ del hashrate eso tarda ~ 14 días $\div *x*$, o sea ~ 9 meses con el 5% y décadas con el 0,1%. Y una tercera cosa, de gobernanza: el BIP declara que «The FAILED state is never reached because timeout is disabled», así que el protocolo no tenía forma de declararse fracasado. Lo cerró una línea de changelog del autor el 09/08 y un PR mergeado el 10/08. La máquina de estados no supo decir que no; lo dijo una persona.

5. Cuantificar la resiliencia de la red Bitcoin: qué decide un fork contencioso

Quantifying Bitcoin Network Resilience: what decides a contentious fork — [fuente](#)

Cada vez que se discute un cambio en Bitcoin, la conversación termina en el mismo lugar: cuántos mineros lo apoyan. Este paper dice que esa es la variable equivocada. Armaron 1.330 escenarios distintos con nodos Bitcoin reales corriendo en un laboratorio, moviendo cinco perillas —cuánto peso económico tiene cada lado, cuán comprometidos están los pools, su ideología, el hashrate y cada cuánto se ajusta la dificultad— y buscaron qué combinaciones producen una cadena partida que no se arregla.

La respuesta es que lo que decide es dónde está la plata: quién custodia las monedas y por dónde pasa el volumen. El hashrate resuelve una capa, pero la adopción económica resuelve otra, y las dos pueden terminar en lados distintos. Dan números concretos de dónde están los bordes, y hasta un resultado paradójico: pasar el pool más grande al lado que quiere el cambio, en ciertas condiciones, da vuelta el resultado en contra. Otro hallazgo incómodo para la sala: en su modelo, los nodos de usuarios individuales no mueven la aguja. Ojo con una cosa al leerlo: el paper es del 5 de agosto, tres días antes de lo de BIP-110. Que lo que pasó después parezca darle la razón es una lectura nuestra, no algo que el paper diga.

Metodología: Scenario Discovery con PRIM (Patient Rule Induction Method) sobre una ensemble simulation de 1.330 escenarios válidos corridos en Warnet con nodos `bitcoind` reales, variando peso económico, mining-pool commitment, ideología de pool, hashrate y régimen de retarget. El peso económico se modela como $0.7 \times \text{custody_btc} + 0.3 \times \text{volume_btc}$, calibrado de modo que cuatro nodos de exchange concentran $\sim 2/3$ del

total. Los umbrales que reporta: piso de soporte económico 0,45-0,50, techo de override 0,78-0,82, y un Economic Self-Sustaining Point en 0,74 entre ambos. El régimen de retarget importa: con 2.016 bloques (mainnet) los escenarios son 1,6× más contenciosos en promedio que con 144, porque los pools comprometidos tienen que sostener pérdidas durante epochs enteros.

Los dos hallazgos con más filo para un socrático. Uno: el flip-point en 0,214 de hashrate comprometido, punto en el que reasignar el pool más grande (calibrado como Foundry, ~30% del hashrate total) al lado que actualiza invierte paradójicamente el resultado —«increasing pool commitment to the upgrading fork can reverse the outcome under intermediate economic conditions». Dos: los resultados se resuelven en dos capas independientes, hashrate (gobernada por la estructura de compromiso de pools) y adopción económica (gobernada por ideología de pool y tolerancia a la pérdida), que pueden resolver por separado. Y el que le pega a la mística: «Individual user nodes show no detectable influence on fork outcomes under the modeled economic weightings». Cierran con tres preguntas de monitoreo responsables con datos públicos —¿el soporte económico está arriba de ~0,50?, ¿de qué lado está el pool comprometido más grande?, ¿se cruzó el techo de override de ~0,82?— monitoreables «from exchange pricing, pool block attribution, and block-production rates». Limitaciones declaradas por los propios autores: el ratio 2.197:1 entre usuarios y peso económico es una calibración específica, el cap de ±20% de divergencia de precio subestima divergencias reales extremas, la topología de red es estática, y el switching de pools usa umbrales determinísticos en vez de juicio estratégico.

6. BTCPay Server 2.4.2: bypass de 2FA explotado activamente

BTCPay Server 2.4.2: actively-exploited 2FA bypass — [fuente](#)

BTCPay Server es el software que usa el que quiere cobrar en Bitcoin sin depender de un procesador de pagos: se lo instala en su propio servidor y la plata le llega directo. Como cualquier panel de administración, tiene login con contraseña y una segunda capa —el código de seis dígitos de la app del celular, o una llave física USB. El 7 de agosto de 2026 se publicó una actualización de emergencia: había una forma de entrar salteándose esa segunda capa, y ya se estaba usando contra instalaciones reales.

El agujero no estaba en la pantalla de login sino en la API, la puerta por donde entran los programas en vez de las personas. Ahí el chequeo estaba mal escrito: en vez de preguntar «¿este usuario tiene segundo factor activado?», preguntaba «¿este usuario tiene una llave física registrada?». Al que usaba el código del celular y no tenía llave física —que es la mayoría— la API lo dejaba pasar con solo email y contraseña. Es el tema más doméstico de todo el temario: si tenés un BTCPay corriendo, esto te toca a vos, hoy.

El fix es el PR #7491: la autenticación Basic de la Greenfield API evaluaba ``user.Fido2Credentials. Any()`` para decidir si bloquear el login, lo que dejaba afuera a los usuarios con el flag ``AuthenticatorEnabled`` (TOTP) y cero registros FIDO2. Se reemplazó por ``await signInManager.IsTwoFactorEnabledAsync(user)``, que cubre cualquier forma de 2FA y alinea el comportamiento de la API con el de la UI, que ya lo enforceaba correctamente. Es un error de lógica de autorización, no de criptografía: la primitiva funcionaba, la condición preguntaba otra cosa.

La release 2.4.2 abre con «This release contains fix of a critical vulnerability that is being actively exploited. You need to update as fast as you can», y agrega un breaking change de endurecimiento (#7492): la autenticación Basic queda deshabilitada por default cinco minutos después de crear la cuenta, con opt-in explícito desde settings y desde la API. Los integradores además tienen que subir NBXplorer a 2.6.10.

La misma release trae fixes no relacionados de firma y finalización de PSBT multisig (#7484, #7488) y de carga del stylesheet RTL (#7482). No hay GHSA ni CVE formal: el rastro público es la release más el PR.

7. Extracción y verificación de direcciones ilícitas de Bitcoin a partir de discusiones en foros clandestinos

Extracting and Verifying Illicit Bitcoin Addresses from Underground Forum Discussions — [fuente](#)

Cuando alguien dice «esta dirección de Bitcoin es de un delincuente», vale la pena preguntar cómo lo sabe. Los listados que usan la academia y las empresas de análisis salen de denuncias de víctimas, de heurísticas sobre la cadena, de recopilaciones hechas después de cada incidente, o directamente de etiquetado propietario que nadie puede revisar. Casi nunca se puede reproducir el procedimiento, y casi nunca hay prueba de que esa dirección haya estado realmente involucrada en algo.

Este trabajo da vuelta el ángulo. En vez de mirar la cadena para inferir quién es quién, va a un foro de cibercrimen con quince años de archivo y saca las direcciones de lo que los propios actores publicaron: la que pusieron para cobrar el malware, la del servicio de hackeo por encargo. Después las filtran con un modelo de lenguaje, las revisa un experto a mano, y se quedan solo con las que tienen actividad real en la cadena. Publican el dataset y el pipeline completo, que es exactamente lo que hoy no existe. Y engancha directo con el otro paper forense del temario: uno mide cuánto miente la herramienta de agrupar, este ataca de dónde salen las etiquetas.

Pipeline de tres etapas sobre un snapshot de HackForums con 42.474.325 posts en 212 subforos, cubriendo 2010-2024. Etapa 1: extracción por regex de direcciones de Bitcoin, Ethereum, Litecoin, Dash y Monero sobre posts y firmas – 146.875 strings candidatos, que validan a 27.131 direcciones, de las cuales 23.882 (88,0%) son Bitcoin; solo se retienen las que tienen historial de transacciones confirmado on-chain. Etapa 2: screening asistido por LLM con Qwen3-30B en zero-shot y temperatura 0,3, que reduce de 23.882 a 2.606 direcciones marcadas como potencialmente ilícitas y les asigna una de doce categorías de cibercrimen. Etapa 3: revisión experta manual de las 2.606, que confirma 2.438 ilícitas, reclasifica 98 como lícitas y excluye 70 por evidencia insuficiente.

Los números de calidad del screening, que son los que hay que mirar con desconfianza sana: precisión del LLM 93,6% (2.438 de 2.606), y recall estimado en 74,1% a partir del muestreo manual de 200 direcciones NO marcadas, donde encontraron 8 falsos negativos. La distribución por categoría está dominada por Fraud (34,5%), Malware (13,5%), Investment Schemes (12,8%) y Services/hacking-for-hire (12,1%), sobre una taxonomía de doce que incluye además Abuse, Account Selling, Card Dumps, Counterfeit, Drugs, Laundering, Membership, Weapons y Other. Los autores publican el dataset, la metadata temporal y el pipeline de extracción completo, que es el aporte que distingue esto de un listado propietario más.

Lanzamientos

Salieron hace poco. Se anuncian, no se debaten.

- **Stratum V2 minó su primer bloque** — En un pool tradicional el pool decide qué transacciones entran al bloque y el minero solo pone la electricidad. Stratum V2 le devuelve esa decisión al minero (job declaration), y el 25/06/2026 se minó el primer bloque conocido donde la plantilla la construyó el minero: el bloque 955.318, con el pool DMND validando y GoMining declarando su propia plantilla e incluyendo sus pagos GoBTC. Es un cambio de poder, no una mejora técnica, y conecta directo con el reverse engineering del Canaan. ([fuente](#))