



Pauta del expositor

Blockchain Learning Path · muestra del programa

66 clases · 91 prácticas · de cero a la infraestructura financiera

v0.14.0

7 diapositivas

≈ 35 min

4 anexos

Cómo se usa. Proyecta [PRESENTACION.pdf](#) a pantalla completa y ten esta pauta impresa o en un segundo monitor. Cada diapositiva trae **tres bloques separados a propósito**, y no se leen igual:

- **En pantalla** — lo que ve el público. Es contexto: no se lee.
-  **Lo que dices** — el guion, palabra por palabra, en intervenciones numeradas. **Esto se pronuncia tal cual está escrito.** Si te pierdes, el número te devuelve al sitio exacto.
-  **Lo que haces** — las acotaciones: qué abrir, dónde detenerte, qué recortar. **Nada de esto se dice en voz alta.**

Los tiempos suman 35 minutos sobre 7 láminas, así que la charla entera cabe en una franja de tres cuartos de hora dejando margen para preguntas. **Si expones hoy, empieza por el primer anexo:** es la comprobación de los diez minutos previos. Y si el hueco que tienes no son 35 minutos, el anexo siguiente dice exactamente qué proyectar y qué sacrificar.

Guion en un vistazo

#	DIAPPOSITIVA	TIEMPO
1	Blockchain Learning Path	2 min

#	DIPOSITIVA	TIEMPO
2	Qué enseña, y qué no	6 min
3	Las 66 clases en diez etapas	6 min
4	Cómo se aprende: una clase y 91 prácticas	6 min
5	Demo en vivo: 90 segundos de terminal	4 min
6	Seguridad, casos reales y por qué creerle al material	6 min
7	Para quién es, y cómo se empieza	5 min
TOTAL		35 MIN

Diapositiva a diapositiva

1 Blockchain Learning Path

 2 min

EN PANTALLA — lo que ve el público

De cero a la infraestructura financiera programable, en español.

- 66 clases secuenciales · 91 prácticas · un proyecto final con caso custodial.
- Criptografía, Bitcoin, Ethereum, contratos, seguridad, producción, dinero y regulación.
- Web, manual en PDF y apps que **funcionan sin conexión**.
- Material público: código MIT y contenido educativo original de esta distribución bajo CC BY-NC-SA 4.0; se conservan los permisos históricos.

 LO QUE DICES — léelo tal cual

- 1 Buenos días, y gracias por el rato. Soy Vladimir Acuña, y les voy a mostrar un programa de formación en blockchain, en español, que está publicado y que se puede empezar a usar hoy mismo.
- 2 La frase que lo resume es esta: lleva a una persona desde no saber qué es un hash hasta poder sentarse en una reunión y discutir cómo se liquida un bono tokenizado.
- 3 Y antes de que nadie lo pregunte, lo aclaro yo: esto no es un curso de inversión ni de trading. No hay una sola recomendación de compra en todo el material. Esto es ingeniería.

- 4 Son sesenta y seis clases que se estudian en orden, noventa y una prácticas que se ejecutan, y un proyecto final.
- 5 Está disponible en la web, en un manual en PDF, y en dos aplicaciones —una de Windows y una de Android— que funcionan sin conexión.
- 6 El código sigue bajo MIT. El contenido educativo original de esta distribución se ofrece bajo CC BY-NC-SA 4.0; los permisos históricos continúan según el historial de licencias. Pueden usar cada componente en su aula o equipo conforme a la licencia de la versión elegida, con la atribución y las demás condiciones que correspondan.
- 7 Vamos a estar treinta y cinco minutos.
- 8 **Si el grupo es pequeño:** Interrúmpame cuando quieran, no hace falta esperar al final.
- 9 **Si es un comité o una sala grande:** Les pido que guardemos las preguntas para el final, así llegamos con tiempo a todo.

LO QUE HACES — no se dice en voz alta

- Si quieres añadir una línea de credenciales tuyas, va después de la intervención 1 — una sola frase, y sigue. Nadie vino a oír tu currículum.
- Elige **una** de las dos versiones finales —la 8 o la 9— y cúmplela el resto de la charla. Prometer preguntas al final y luego aceptarlas a mitad rompe el tiempo.
- Si el público es institucional, haz una pausa después de explicar que cada versión declara condiciones de uso claras: es el dato que les cambia la conversación.
- No entres todavía en el mapa de clases. Aquí solo se abre la puerta.

2 Qué enseña, y qué no

 6 min

EN PANTALLA — lo que ve el público

Blockchain no es sinónimo de criptomoneda: esto enseña a decidir, no a especular.

- Cuándo una cadena de bloques **aporta valor** y cuándo una base de datos es mejor.
- Cómo **construirla** con pruebas automatizadas, no con capturas de pantalla.
- Cómo **llevarla a una empresa**: infraestructura, costos, riesgo y cumplimiento.
- Qué cambia —y qué **no**— cuando el dinero y los valores se vuelven programables.

LO QUE HAY DENTRO	CUÁNTO
Clases secuenciales (1 → 66), en diez etapas	66
Prácticas guiadas con evidencia y criterio de aceptación	91
Preguntas de autoevaluación, corregidas al instante	117
Pruebas automatizadas que la CI ejecuta en cada cambio	345



LO QUE DICES — léelo tal cual

- 1 Empecemos por lo que más ruido hace. Cuando uno dice "blockchain", la mayoría de la gente piensa en el precio de una moneda. Este programa no va de eso, y quiero sacarlo del medio en el primer minuto.
- 2 La primera pregunta que hace el programa no es "cómo se hace". Es: "¿de verdad lo necesito?".
- 3 Y la respuesta honesta, muchas veces, es que no. Si un sistema tiene un dueño claro, si las partes confían entre ellas y si nadie necesita verificar nada por su cuenta, una base de datos de toda la vida es más barata, más rápida y más fácil de operar.
- 4 Eso no es una opinión mía para quedar bien: es literalmente la clase 1, la primera que hace todo el mundo.
- 5 A partir de ahí, el programa hace cuatro cosas.
- 6 La primera es enseñar a decidir: cuándo una cadena de bloques aporta valor y cuándo no.
- 7 La segunda es enseñar a construirla, con pruebas automatizadas. No con una captura de pantalla de que un día funcionó.
- 8 La tercera es enseñar a llevarla a una empresa: infraestructura, costos, riesgo y cumplimiento. Esa parte casi nunca está en los cursos.
- 9 Y la cuarta es entender qué cambia —y sobre todo qué no cambia— cuando el dinero y los valores se vuelven programables.
- 10 Sobre los números de la tabla, no los voy a leer todos. Me quedo con dos.

- 11 Noventa y una prácticas. Eso significa que nadie termina este programa habiendo solamente leído. Cada práctica dice qué hay que hacer, qué evidencia hay que producir y con qué criterio se acepta.
- 12 Y trescientas cuarenta y cinco pruebas automatizadas. Eso significa que el material se comprueba ejecutándolo, no afirmándolo.
- 13 Y hay un detalle que para mí es lo más importante de esta lámina: esas cifras las verifica la propia integración continua. Si alguien añade una clase y no actualiza este número, la comprobación falla y el material no se publica. El número que están viendo no es un número de folleto.

LO QUE HACES — no se dice en voz alta

- Esta lámina fija el tono de toda la charla. Es la más importante de las siete y la única que **nunca** se recorta.
- No leas la tabla entera. Señala solo las dos filas que vas a comentar.
- Si alguien interrumpe preguntando por precios o por invertir, contesta con la respuesta del anexo de preguntas y vuelve inmediatamente a la intervención donde estabas.
- Marca con la voz el contraste "qué cambia / qué no cambia": es la idea que sostiene la segunda mitad del programa.

3 Las 66 clases en diez etapas

 6 min

EN PANTALLA — lo que ve el público

Cada clase prepara la siguiente. Se estudian en orden.

ETAPA	CLASES	FOCO
Orientación	1–2	¿Necesito blockchain?
Fundamentos	3–8	Criptografía, redes P2P, consenso
Desarrollo	9–16	Bitcoin, EVM, Solidity, dApps
Profesional	17–24	Tokens, seguridad, oráculos, DAO
Avanzado	25–32	L2, interoperabilidad, ZK, arquitectura
Producción	33–38	Infraestructura, empresa, implementación
Finanzas on-chain	39–52	DeFi, dinero, stablecoins, MDBC, pagos, tokenización
Institucional	53–56	Custodia, identidad digital, regulación
Analítica de datos	57–58	Minería de datos on-chain, grafos, patrones, anomalías

ETAPA	CLASES	FOCO
Custodia y auditoría	59–66	Exchanges, conciliación, PoR/PoL, forensics y gobernanza

Entre las clases 10 y 11 se cruza **Wallets desde cero**: uso, seguridad y recuperación.

LO QUE DICES — léelo tal cual

- 1 Este es el mapa completo. Denle unos segundos, porque es todo lo que vamos a recorrer.
- 2 Sesenta y seis clases, agrupadas en diez etapas, y se estudian en orden.
- 3 El orden no es un capricho editorial: es una cadena de dependencias. No se puede entender una stablecoin sin entender antes un token, ni un token sin entender la máquina virtual de Ethereum, ni esa máquina sin entender firmas y hashes.
- 4 Fundamentos, del uno al tres: qué es lo que hace verificable a una cadena. Hash, firmas, árboles de Merkle, propagación en la red y consenso. Una blockchain no es magia: es la composición de tres cosas que ya existían por separado.
- 5 Desarrollo, del cuatro al siete: aquí se pasa de entender a construir. Bitcoin y su modelo de UTXO, cuentas y gas en Ethereum, contratos en Solidity probados con Foundry, y una aplicación que firma, envía y sabe manejar los errores.
- 6 Profesional, del ocho al once: escribir contratos que otra persona pueda auditar. Estándares de token, seguridad, oráculos y gobernanza con retardo temporal.
- 7 Avanzado, del doce al quince: qué se hace cuando una cadena sola no alcanza. Y la idea que me interesa que se lleven de esta etapa es que escalar no sale gratis: se paga con supuestos de confianza.
- 8 Producción, del dieciséis al dieciocho. Esta es la etapa que casi ningún curso tiene, y es la que más pesa en una entrevista de trabajo. Porque la pregunta que hunde proyectos no es técnica: es "¿por qué esto justifica su costo y su mantenimiento durante los próximos cinco años?".

- 9 Finanzas on-chain, del diecinueve al veinticinco: DeFi, dinero y liquidación, stablecoins, depósitos tokenizados y monedas digitales de banco central, pagos y cambio de divisas, tokenización de activos reales y mercados de capitales. Aquí es donde este programa se separa de la oferta habitual.
- 10 Institucional, veintiséis y veintisiete: custodia, identidad digital y regulación comparada —Chile, Europa con MiCA, Estados Unidos, América Latina y los estándares internacionales—. Y una idea que atraviesa toda esta etapa: el cumplimiento no es un trámite que se añade al final, es una restricción de diseño.
- 11 Y la novena etapa, las clases 57 y 58: leer la propia cadena como fuente de datos. Grafos, patrones, detección de anomalías. Y sobre todo, saber qué no se puede concluir: una dirección no es una persona, y un patrón no es una prueba.
- 12 Me falta una línea, la de abajo, que parece menor y no lo es. Entre las clases 10 y el cinco hay una unidad sobre wallets: qué son, cómo se usan sin perder los fondos y qué hacer cuando algo sale mal.
- 13 La puse ahí porque el punto donde más gente abandona no es la criptografía. Es la primera vez que tiene que firmar algo de verdad.

LO QUE HACES — no se dice en voz alta

- Después de la intervención 1, **cállate tres o cuatro segundos** y deja que lean el mapa. Es la única pausa larga de la charla.
- Una intervención por etapa y avanza. No te enredas en ninguna: cada una tiene su propia clase si alguien pregunta.
- **Este es el punto por donde se recorta si vas corto.** Di las diez etapas en dos frases —"de fundamentos a producción, y después toda la parte financiera y regulatoria"— y salta a la lámina 4.
- Si el público es de banca o de sector público, alarga la etapa de finanzas on-chain y acorta la de desarrollo. Si es técnico, al revés.

4 Cómo se aprende: una clase y 91 prácticas

🕒 6 min

EN PANTALLA — lo que ve el público

Las 66 clases comparten una anatomía clara, pero cada una usa una estrategia didáctica propia y ninguna se aprueba solo leyendo.

- 🎯 **Objetivos** medibles · 🗺️ **temas** con su porqué · 🧩 **esquema visual**.
- 🧠 **Modelo mental** con su analogía y los **límites** de la analogía.
- 🔬 **Profundización** con ejemplos numéricos · ⚠️ **errores frecuentes** (síntoma → causa).
- 🧪 **Laboratorio guiado** y ✂️ **reto verificable** con criterio de aceptación.
- 🔗 **Referencias** a fuentes primarias, con enlaces vivos y comprobados cada semana.
- **91 prácticas**: 62 con verificación ejecutable (`pnpm lab:*`), el resto con rúbrica.
- Contratos con **Foundry** (fuzzing e invariantes), dApp con **viem** e indexador.

🗣️ LO QUE DICES — léelo tal cual

- 1 Ya vieron qué se enseña. Ahora les quiero mostrar cómo.
- 2 Las sesenta y seis clases conservan señales visuales comunes para orientarse, pero no repiten la misma dinámica: unas demuestran, otras simulan, investigan, debaten o auditan. Cuando el estudiante aprende a navegar una, reconoce dónde están la evidencia y las fuentes en las demás sin confundir sus actividades.
- 3 Objetivos medibles, los temas con su porqué, un esquema visual, y un modelo mental con su analogía.
- 4 Y aquí me quiero detener, porque es la sección de la que estoy más orgulloso: cada analogía viene acompañada de otra que dice dónde deja de servir.
- 5 Las analogías mal cerradas son la principal fuente de malentendidos en este campo. "Blockchain es como un libro contable compartido" — hasta que deja de serlo, y conviene saber exactamente en qué punto.
- 6 Después viene la profundización, con ejemplos numéricos, y los errores frecuentes, escritos como síntoma y causa. No como una lista de consejos: síntoma, y causa. Que es como se depura de verdad.

- 7 Cada clase declara además de dónde saca lo que afirma. Y hay una comprobación automática que exige un mínimo de tres enlaces a fuente primaria, porque "según Antonopoulos", sin enlace, no significa nada.
- 8 Y como varias de esas obras tienen edición legalmente gratuita, el programa se puede seguir entero sin comprar un solo libro.
- 9 Luego están las noventa y una prácticas. Sesenta y dos se verifican solas, con un comando. Las demás producen una evidencia que se revisa con una rúbrica.
- 10 Esa mezcla es deliberada. Hay cosas que una máquina verifica mejor que una persona: un hash, una invariante de un contrato. Y hay otras que exigen criterio humano, como justificar una decisión de arquitectura. Este programa no finge que todo se puede automatizar.
- 11 Fíjense también en que todas las prácticas tienen una columna que dice "evidencia". Sin un artefacto concreto, "hice el laboratorio" no es comprobable, ni para quien estudia ni para quien evalúa.
- 12 Y para que esto no se quede en una promesa mía, vamos a ejecutar dos de esas prácticas ahora mismo.

LO QUE HACES — no se dice en voz alta

- Abre una clase real en el sitio —la 3 de criptografía sirve— y recórrela con el cursor mientras hablas las intervenciones 3 a 7.
- Detente físicamente en dos sitios de la pantalla: **límites de la analogía y errores frecuentes**. Son las dos secciones que venden el material.
- Después de la intervención 12, **pasa a la lámina 5 antes de cambiar a la terminal**. Así el público se queda con los números delante mientras tú buscas la ventana.
- Si vas justo de tiempo, esta lámina admite perder dos minutos: quita las intervenciones 7 y 8 (bibliografía) y no se nota.

5 Demo en vivo: 90 segundos de terminal

 4 min

EN PANTALLA — lo que ve el público

Nada de esto es una promesa: se ejecuta delante del público.

- `pnpm lab:hash` — encadena bloques, altera un dato y **todos los siguientes quedan inválidos**.
- `pnpm lab:remesa` — desglosa comisión a comisión el costo real de enviar 200 USD.

REMESA DE 200 USD	COSTO	SOBRE EL ENVÍO
Vía tradicional (cuatro comisiones)	12.00	6.0 %
Corredor on-chain, última milla barata	2.80	1.4 %
Corredor on-chain, última milla cara	13.30	6.7 %

- De esos 12.00, **6.50 no se anuncian como comisión**: van en el margen de cambio.
- **El tramo que la cadena no toca —la última milla— decide el resultado.**

LO QUE DICES — léelo tal cual

- 1 Esto no lo tienen que creer porque yo lo diga. Lo vamos a ejecutar aquí, ahora.

DEMO A · PÚBLICO TÉCNICO O MIXTO — `PNPM LAB:HASH`

- 2 Lo que ven aquí es una cadena de dos bloques. Cada bloque guarda el hash del bloque anterior; ese es todo el truco.
- 3 Ahora voy a cambiar un dato de en medio... y fíjense en la última línea: la verificación acaba de pasar a "falso".
- 4 Esto es lo que quiere decir "inmutable", y no es que no se pueda cambiar. Es que falsificar un bloque obliga a rehacer todos los que vienen después.

DEMO B · PÚBLICO FINANCIERO O DE NEGOCIO — `PNPM LAB:REMESA`

- 5 Vamos a contar cuánto cuesta de verdad enviar doscientos dólares a otro país.
- 6 Por la vía tradicional: cinco dólares de comisión de envío, cinco de margen de cambio, uno con cincuenta de bancos intermedios y cincuenta centavos de retiro en destino. Doce dólares. Un seis por ciento del envío.
- 7 Y de esos doce dólares, seis con cincuenta no se anuncian nunca como comisión: van escondidos dentro del tipo de cambio.
- 8 Ahora el mismo envío por un corredor on-chain, con una última milla barata: dos con ochenta. Gana la cadena, y gana por mucho.

- 9 Pero el mismo envió otra vez, con una última milla cara —efectivo, poca competencia en el país de destino—: trece con treinta. Pierde la cadena.

CIERRE COMÚN A LAS DOS DEMOS

- 10 Y ese es el argumento honesto de todo el programa. El tramo que la cadena no toca es el que decide el resultado.
- 11 Aquí no les estoy vendiendo que la tecnología arregle el problema. Les estoy enseñando a calcular qué parte arregla. Un vendedor les habría enseñado solo la fila de los dos con ochenta.
- 12 Y esto es determinista y sin red: mismos datos, mismo resultado, sin claves y sin fondos reales. Lo pueden reproducir ustedes con dos comandos cuando salgan de aquí.

LO QUE HACES — no se dice en voz alta

- **Elige una sola demo**, la A o la B, según el público. No hagas las dos: no hay tiempo y se diluye el remate.
- Los números están impresos en la lámina. **Si la terminal falla, no la arregles delante del público**: señala la tabla, di el cierre común y sigue. La demo es el adorno; el argumento es la tabla.
- En la demo B, apunta con el cursor a la fila de los 6.50 antes de decirlo: es la cifra que la gente no había visto nunca.
- Después del contraste 2.80 / 13.30, **haz una pausa** antes del cierre. Es el remate de toda la charla.
- Ejecuta las dos órdenes una vez **antes de que entre el público**: la primera ejecución tarda más y ese silencio se hace largo.

6 Seguridad, casos reales y por qué creerle al material

 6 min

EN PANTALLA — lo que ve el público

Primero se rompe el contrato, después se arregla. Y todo se verifica solo.

- Contratos **vulnerables a propósito**, con su exploit ejecutable y su corrección.
- Reentrada, control de acceso, desbordamiento y dependencia de oráculos, con **Slither** en la CI.

- Cuatro casos reales con las cuentas hechas: **Terra/UST, FTX, puente Ronin y El Salvador**.
- **345 pruebas** en cada cambio, más comprobaciones de coherencia y de enlaces vivos.
- Los binarios se verifican **abriéndolos y contando el contenido**: un build en verde no prueba que la app lleve el curso dentro.

LO QUE DICES — léelo tal cual

- 1 Hay dos razones para creerle a un material como este, y las dos están en esta lámina.
- 2 La primera tiene que ver con cómo se enseña seguridad aquí: primero se rompe el contrato, y después se arregla. En ese orden.
- 3 El programa trae contratos vulnerables a propósito, cada uno con su ataque ejecutable y con su corrección al lado.
- 4 El ejemplo clásico es la reentrada. El contrato envía los fondos antes de anotar que ya los envió. Y quien los recibe vuelve a entrar en la misma función antes de que el saldo se haya actualizado. Y otra vez. Y otra.
- 5 Quien solamente leyó qué es la reentrada la reconoce en un examen. Quien la ejecutó la reconoce en una revisión de código a las once de la noche. No es lo mismo.
- 6 Y el encuadre ético está escrito en el propio material: todo corre en local o en una red de pruebas. Nunca con fondos ni con claves reales.
- 7 Después están los casos reales. Son cuatro, con las cuentas hechas: Terra, FTX, el puente de Ronin y El Salvador.
- 8 Los cuatro están escritos con la misma estructura: qué se prometió, qué mecanismo falló, qué señales había antes, y qué decisión de diseño lo habría evitado.
- 9 FTX es el que más me sirve para cerrar el argumento de la parte institucional: no cayó por un fallo criptográfico. Cayó por custodia y por controles internos.
- 10 Y la segunda razón para creerle a este material es que se verifica solo.

- 11 Trescientas cuarenta y cinco pruebas en cada cambio, comprobaciones de coherencia entre lo que el material dice y lo que el material tiene, y los enlaces externos revisados cada semana.
- 12 Hay además una regla escrita en el repositorio que me gusta especialmente: una compilación exitosa no demuestra que el archivo contenga el material.
- 13 Por eso la integración continua abre el instalador de Windows y abre el APK de Android, y cuenta las clases, las páginas y las preguntas que llevan dentro. Si sale vacío, la publicación falla — en vez de entregarles a ustedes una aplicación que se instala, se abre, y está en blanco.

LO QUE HACES — no se dice en voz alta

- Con público de negocio, los casos reales son el momento en que más te van a escuchar. No los aceleres aunque vayas justo.
- Si te preguntan si los contratos del curso están auditados, la respuesta está en el anexo de lo que no hay que prometer. **No improvises esa.**
- Las intervenciones 3 a 5 admiten abrir el contrato vulnerable en pantalla si el público es técnico y tienes el sitio a mano.
- Si recortas, quita la intervención 11 y encadena la 10 con la 12 y la 13: la de los binarios verificados es la que se recuerda.

7 Para quién es, y cómo se empieza

 5 min

EN PANTALLA — lo que ve el público

Nueve rutas, un proyecto final y cuatro formatos que salen del mismo build.

- **Rutas por perfil:** desarrollo, arquitectura, auditoría, producto, investigación, empresa, DeFi, banca y cumplimiento.
- Cada ruta termina en un **entregable de portafolio**: dApp probada, ADR, informe de auditoría, ficha de riesgo.
- **Proyecto final:** protocolo probado, dApp, datos, arquitectura, modelo de amenazas y caso de negocio.
- **Llévatelo entero:** sitio web, manual PDF de ~650 páginas, app de Windows y APK de Android, sin conexión.
- **Empieza hoy:** abre *Empieza aquí* → haz el diagnóstico → clase 1 y `pnpm lab:hash`.



LO QUE DICES — léelo tal cual

- 1 Última lámina: para quién es esto, y cómo se empieza.
- 2 Hay nueve rutas por perfil: desarrollo, arquitectura, auditoría, producto, investigación, empresa, DeFi, banca y cumplimiento.
- 3 Nadie tiene que hacerlo entero. Cada ruta dice qué clases priorizar y cuáles se pueden leer en diagonal.
- 4 Lo que sí tienen todas las rutas es un entregable de portafolio: una aplicación probada, un documento de decisión de arquitectura, un informe de auditoría, una ficha de riesgo.
- 5 Porque estudiar sin producir nada demostrable es tiempo que después no se puede acreditar delante de nadie.
- 6 El proyecto final, el que cierra el programa, no pide originalidad. Pide criterio: decisiones justificadas y verificadas.
- 7 Y tiene un requisito que incomoda a todo el mundo, que es justamente el que más enseña: un documento que responda "¿por qué blockchain, y qué alternativa descarté?".
- 8 Si la respuesta honesta es que bastaba una base de datos, el proyecto sigue siendo válido. Lo que se evalúa es el razonamiento, no la tecnología elegida.
- 9 Y llévenselo entero. Está el sitio web, un manual en PDF de unas seiscientas cincuenta páginas, la aplicación de Windows y el APK de Android.
- 10 Las aplicaciones funcionan sin conexión, que es lo que importa en un aula sin wifi o en el metro. Y las cuatro versiones salen de la misma compilación, así que ninguna se queda atrás cuando el material cambia.

- 11 Una advertencia honesta antes de terminar: los binarios no están firmados con certificado de código, porque ese certificado cuesta cientos de dólares al año. Windows y Android les van a avisar de que el origen es desconocido. Por eso se publica el SHA256 de cada archivo, para que puedan comprobar que descargaron lo que se publicó.
- 12 Y cómo se empieza hoy: abren la página "Empieza aquí", hacen el diagnóstico, y entran al clase 1. El primer laboratorio son dos comandos.
- 13 El enlace está en pantalla y lo dejo ahí mientras respondemos preguntas.

LO QUE HACES — no se dice en voz alta

- **No enumeres las nueve rutas de corrido.** Pregunta al público a qué se dedica y comenta solo las dos o tres que aparezcan.
- Cierra con la acción concreta de la intervención 12, **no con un agradecimiento.**
- Si están frente a un computador, pídeles que abran la página y ejecuten el primer laboratorio antes de irse: quien ve funcionar algo suyo el primer día vuelve el segundo.
- Deja la lámina en pantalla durante todas las preguntas, y ten abierto el anexo de preguntas frecuentes: casi todo lo que te van a preguntar está ahí con la respuesta ya redactada.

Anexos: lo que no se proyecta

Antes de empezar: la comprobación de los diez minutos previos

Nada de esto es opcional si la charla es hoy. En orden, con el proyector ya conectado:

1. **Proyector en 16:9** y **PRESENTACION.pdf** a pantalla completa. Las láminas miden 1280×720 px exactos: en una pantalla 4:3 saldrán con banda negra arriba y abajo, que es preferible a dejar que el visor las recorte.

2. **La pauta en otro sitio.** Impresa o en el segundo monitor, nunca en la pantalla que proyectas. Si solo tienes una pantalla, imprímela: son pocas hojas.
3. **Terminal abierta en el repositorio**, con letra grande (18 pt o más) y `pnpm install` ya ejecutado. Nadie lee una fuente de 11 pt desde la quinta fila.
4. **Ensaya la demo una vez**, en silencio, antes de que entre el público: `pnpm lab:hash` y `pnpm lab:remesa`. La primera ejecución de Node siempre tarda más, y ese silencio se hace eterno en público.
5. **Pestañas abiertas y en este orden:** el sitio del programa, la clase 3, el catálogo de laboratorios y el caso de FTX. Cierra todo lo demás.
6. **Plan B sin wifi:** ten descargados `PRESENTACION.pdf` y `MANUAL.pdf`, y la app de escritorio o el APK instalados. Si la sala se queda sin red y tú sigues, acabas de demostrar el argumento de la lámina 7 sin decir una palabra.
7. **Silencia las notificaciones** del sistema y del móvil, y cierra el correo si vas a compartir pantalla.

Un último detalle que se olvida siempre: **ten el enlace del repositorio en un formato que se pueda copiar** —un mensaje al chat de la sala, un código QR en el móvil—, porque cuando termines te lo van a pedir y dictarlo letra a letra queda mal.

Si tienes más o menos tiempo

La charla completa son las siete láminas, unos **35 minutos** sin contar preguntas. Cuando el hueco es otro, esto es lo que se hace:

TIENES	QUÉ PROYECTAS	QUÉ SACRIFICAS
10 min (comité, pasillo)	1 · 2 · 7	La demo y el mapa; queda un solo dato: 66 clases, 91 prácticas y todo verificado en CI
20 min	1 · 2 · 3 (en dos frases) · 5 (solo la remesa) · 7	La anatomía de las clases y media lámina de seguridad

TIENES	QUÉ PROYECTAS	QUÉ SACRIFICAS
35 min (completa)	Las siete, con los tiempos de esta pauta	Nada
50 min (clase)	Las siete, una clase real abierta en el sitio y las dos demos	Nada; sobra tiempo para preguntas largas

Reglas de recorte, por si hay que decidir en caliente:

- **La lámina 2 no se toca nunca.** Es la que fija el encuadre; sin ella la charla parece una promoción de criptomonedas.
- **La lámina 3 es la que mejor se comprime:** diez etapas en dos frases y adelante.
- **Antes de quitar la demo, quita minutos de la 4.** Quien ve ejecutarse algo recuerda la charla; quien solo ve viñetas, no.
- Si te pasas de tiempo, **no aceleres: corta.** Salta a la lámina 7, di la acción concreta y deja los enlaces en pantalla. Terminar puntual es parte del argumento.

Las preguntas que te van a hacer

Casi todas las preguntas de una sala caen en esta lista. Las respuestas están pensadas para decirse en voz alta, no para leerse:

TE PREGUNTAN	QUÉ RESPONDES
¿Esto enseña a invertir o a hacer trading?	No, y es deliberado. Enseña a decidir si la tecnología aplica, a construirla con pruebas y a operarla. No hay una sola recomendación de compra en todo el material
¿No estará obsoleto en seis meses?	Los fundamentos —hash, firmas, consenso, UTXO, EVM— llevan años estables. Lo que cambia rápido está fechado y con fuente, y hay un historial de cambios: el Merge, Dencun y Pectra ya están incorporados
¿Cuánto tiempo lleva?	El plan es de 26 semanas a ritmo de curso, y hay una ruta rápida para quien ya programa. Nadie tiene que hacerlo entero: las rutas por perfil recortan a lo que necesitas

TE PREGUNTAN	QUÉ RESPONDES
¿Necesito saber programar?	Para las clases 1–12, no. Desde la 13 hace falta JavaScript básico. Y hay rutas —producto, cumplimiento— que llegan al final sin escribir contratos
¿Por qué en español?	Porque casi todo el material serio del área está en inglés, y eso deja fuera a mucha gente que sí puede hacer este trabajo. Los términos técnicos se mantienen en inglés y se traducen la primera vez que aparecen
¿Quién garantiza que esto sigue vivo?	La integración continua: pruebas en cada cambio, enlaces externos revisados cada semana y las cifras del material comprobadas contra los archivos. Si algo se rompe, no se publica
¿Puedo usarlo en mi institución o en mi empresa?	El código MIT permite uso comercial; el contenido educativo original de esta distribución sigue CC BY-NC-SA 4.0. Estudiarlo internamente no equivale por sí solo a explotarlo comercialmente. Para redistribución comercial del contenido, revisa la licencia de la versión usada y el historial de permisos. Hay guía del instructor, syllabus, rúbricas y checklist de laboratorios.
¿Y la regulación de mi país?	Hay una carpeta de regulación con Chile, MiCA, Estados Unidos, LatAm y estándares internacionales, y cada afirmación declara su rango, su fuente oficial y su fecha. No sustituye a un abogado, y así está escrito
¿Da certificado?	El propio repositorio genera un certificado de progreso a partir de la evidencia de los laboratorios. No es una acreditación oficial y no lo presentamos como tal
¿Cuánto cuesta?	Nada. Y las obras de referencia clave tienen edición legalmente gratuita, así que se puede seguir entero sin comprar un libro

Si te preguntan algo que no sabes, la respuesta correcta es **"no lo sé, está en la clase tal, te lo mando"**. En un tema con tanto vendedor, admitir un límite suma credibilidad; improvisar una cifra la destruye para el resto de la sesión.

Lo que no hay que prometer

Este material se sostiene sobre lo que **no** afirma. Cuatro líneas que no se cruzan, aunque la pregunta invite a cruzarlas:

- **Nada de rentabilidad, precios ni recomendaciones.** Ni siquiera en broma, ni siquiera respondiendo a "¿y tú qué comprarías?". El programa no es asesoría financiera, y decirlo en voz alta es parte de la charla.
- **No llames "auditados" a los contratos del curso.** Son material didáctico probado con Foundry y revisado con Slither; algunos son vulnerables **a propósito**. Auditar es otra cosa y cuesta dinero.
- **No des asesoría legal ni tributaria.** La carpeta de regulación cita norma, rango y fecha para que alguien informado pueda buscar; ahí termina su alcance.
- **No inventes cifras.** Las del programa están en la lámina 2 y en esta pauta. Si te piden una que no tienes —cuántos alumnos, cuántas horas exactas, qué salario— di que no la tienes.

Y una tentación que no es una promesa pero se le parece: **no vendas blockchain a quien no la necesita**. Si alguien del público describe un problema que se resuelve con una base de datos, díselo. Es literalmente la primera lección de la clase 1, y hacerlo en directo vale más que las siete láminas juntas.