

CRIPTO · SERIE 2 · PARTE 03 / 04

ARQUITECTURAS DE BLOCKCHAIN

Bitcoin Script frente a la EVM de Ethereum, el gas como solución económica al problema de la parada, el reloj criptográfico de Solana, los canales de pago de Lightning y el trilema como compromiso de diseño cuantificable.

Complemento técnico de la guía «No todo es Bitcoin»

<https://amstech.dev/guias/bitcoin-s2/03-no-todo-es-bitcoin/>

@ams_tech

ÍNDICE

1	Resumen
2	Bitcoin: un lenguaje limitado a propósito
3	Ethereum: un ordenador replicado
4	Solana: velocidad a cambio de hardware
5	Lightning: canales de pago (capa 2)
6	El trilema, cuantificado
7	Lo que el vídeo simplifica
8	Referencias

1. RESUMEN

Todas las blockchains comparten la estructura de la serie 1 (bloques encadenados por huellas), pero difieren en **qué se puede programar en ellas** y en **cuánto rendimiento sacrifican a la descentralización**.

Este dossier compara cuatro diseños:

- el lenguaje de Bitcoin, deliberadamente no Turing-completo (§2);
- la máquina virtual de Ethereum y el **gas**, que resuelve en la práctica el problema de la parada (§3);
- la **prueba de historia** de Solana y su coste en hardware (§4);
- los canales de pago de **Lightning**, una capa 2 (§5).

Por último, cuantifica el trilema con la capacidad de Bitcoin (de 6,7 a 11,8 transacciones por segundo según el tipo de transacción) y los requisitos de un nodo de cada red (§6).

2. BITCOIN: UN LENGUAJE LIMITADO A PROPÓSITO

Cada salida de una transacción lleva un pequeño programa, el **script de bloqueo**, que fija la condición para gastarla. Quien la gasta aporta un **script de desbloqueo**, y el par se ejecuta en una **máquina de pila** [1]. El caso más común (pago a una clave pública):

```
desbloqueo:  <firma> <clave_pública>
bloqueo:      OP_DUP OP_HASH160 <huella_de_la_clave> OP_EQUALVERIFY OP_CHECKSIG
```

El intérprete duplica la clave, calcula su huella, comprueba que coincide con la del bloqueo y verifica la firma. Si la pila termina con «verdadero», el gasto es válido.

El lenguaje **no tiene bucles ni saltos hacia atrás**: cada instrucción se ejecuta como mucho una vez. Por tanto, el tiempo de validación está acotado por la longitud del *script*, y ningún programa puede bloquear a los nodos que lo validan. Aun así permite condiciones útiles: multifirma («2 de 3 firmas»), bloqueos temporales (`OP_CHECKLOCKTIMEVERIFY`, BIP 65) y, desde Taproot, árboles de condiciones alternativas. La filosofía es que Bitcoin es dinero y el lenguaje no necesita más.

3. ETHEREUM: UN ORDENADOR REPLICADO

3.1 LA EVM

Ethereum ejecuta programas arbitrarios en la **Ethereum Virtual Machine** [2][3]: una máquina de pila con palabras de 256 bits, una pila de hasta 1.024 elementos, memoria temporal y un **almacenamiento persistente** por contrato (un diccionario de palabras de 256 bits). Es **Turing-completa**: admite bucles y, en principio, cualquier cálculo. Un **contrato inteligente** es un programa desplegado en una dirección. Cualquier nodo puede reproducir su ejecución, y todos llegan al mismo estado.

3.2 EL PROBLEMA DE LA PARADA Y EL GAS

Turing demostró en 1936 que no existe un algoritmo que decida si un programa arbitrario termina: es el **problema de la parada**. Para un sistema replicado es grave, porque un contrato con un bucle infinito colgaría a todos los nodos y no se puede detectar de antemano.

Ethereum lo resuelve con economía: cada instrucción tiene un precio en **gas**, y cada transacción fija un máximo que paga por adelantado. Si el gas se agota, la ejecución se aborta y los cambios se deshacen, pero el gas consumido se cobra igualmente. Algunos precios [3]:

OPERACIÓN	GAS
Transacción simple (base)	21.000
ADD (suma)	3
KECCAK256 (hash)	30 + 6 por palabra
SSTORE (escribir almacenamiento de cero a no cero)	20.000 (+2.100 si el acceso es «frío»)

Los precios reflejan el **coste para la red**: sumar es casi gratis, pero escribir en el almacenamiento obliga a todos los nodos a guardar ese dato para siempre. Desde EIP-1559, cada bloque tiene una **tarifa base** que sube o baja según la ocupación y que se **quema**; el usuario añade una propina para el proponente.

4. SOLANA: VELOCIDAD A CAMBIO DE HARDWARE

4.1 LA PRUEBA DE HISTORIA

Solana combina prueba de participación con un **reloj criptográfico**, la *proof of history* [4]: una cadena de huellas SHA-256 calculadas en serie,

$$h_{i+1} = \text{SHA256}(h_i), \quad (1)$$

en la que de vez en cuando se mezclan datos (transacciones). Como cada huella depende de la anterior, la cadena **no se puede paralelizar**: su longitud demuestra que ha pasado un tiempo mínimo. Verificarla, en cambio, sí se puede repartir entre muchos núcleos, comprobando tramos distintos a la vez. Los validadores obtienen así un orden de eventos acordado sin intercambiar mensajes para cada uno, y el líder de cada ranura (≈ 400 ms) puede producir bloques sin esperar.

4.2 EL PRECIO

	BITCOIN	ETHEREUM	SOLANA
Tiempo por bloque/ranura	≈ 600 s	12 s	$\approx 0,4$ s
Bloques/ranuras al día	144	7.200	≈ 216.000
Hardware de un nodo	un mini PC	un PC con varios TB de SSD	≈ 24 núcleos, 384–512 GB de RAM, 10 Gbit/s

Los requisitos de un validador de Solana [5] quedan fuera del alcance doméstico, y eso reduce el número de operadores posibles. Es la cara de la descentralización en el trilema.

5. LIGHTNING: CANALES DE PAGO (CAPA 2)

Lightning [6] mueve los pagos **fuera de la cadena** y usa Bitcoin solo como árbitro:

1. **Apertura.** Dos partes bloquean fondos en una salida que exige **las dos firmas** (multifirma 2 de 2). Es una transacción en la cadena.
2. **Pagos.** Para cada pago firman una nueva **transacción de compromiso** con el reparto actualizado, sin publicarla. Cada parte entrega a la otra un **secreto de revocación** del estado anterior: si alguien publicara un reparto antiguo, la otra parte podría usar ese secreto para quedarse con todos los fondos del canal. El castigo disuade de hacer trampa.
3. **Cierre.** Se publica el último reparto: una segunda transacción en la cadena.

Para pagar a alguien sin canal directo, el pago salta por canales intermedios con **HTLC** (*hash time-locked contracts*). Cada salto se libera solo si se revela la preimagen r de una huella $H(r)$ que conoce el destinatario, y caduca si no se revela a tiempo. O el pago llega entero, o no se mueve nada. Las rutas se cifran por capas (enrutamiento cebolla), así que cada nodo intermedio solo conoce el salto anterior y el siguiente.

Miles de pagos pueden quedar resueltos con **dos** transacciones en la cadena. La capacidad de Lightning no está limitada por los bloques de Bitcoin, sino por la liquidez de los canales.

6. EL TRILEMA, CUANTIFICADO

El **trilema de la escalabilidad**, popularizado por Vitalik Buterin, afirma que es difícil lograr a la vez escalabilidad, seguridad y descentralización. No es un teorema, sino un compromiso de ingeniería que se puede medir.

6.1 LA CAPACIDAD DE BITCOIN

El límite de bloque es de 4 millones de **unidades de peso** (BIP 141 [7]), equivalentes a 1.000.000 de *bytes virtuales*. Con un bloque cada 600 s:

$$tx/s \approx \frac{\text{capacidad del bloque}}{\text{tamaño medio de una tx} \times 600 \text{ s}} = \begin{cases} \frac{10^6}{250 \times 600} \approx 6,7 & (\text{tx antigua típica, 250 B}) \\ \frac{10^6}{141 \times 600} \approx 11,8 & (1 \text{ entrada SegWit, 2 salidas, 141 v}) \end{cases} \quad (2)$$

6.2 POR QUÉ NO BASTA CON AGRANDAR LOS BLOQUES

Multiplicar por k el tamaño o la frecuencia de los bloques multiplica por k el ancho de banda, el almacenamiento y el cálculo que necesita **cada nodo completo**. Además, los bloques tardan más en propagarse, y aumenta la probabilidad de que dos mineros encuentren bloques a la vez (bloques huérfanos), lo que favorece a los mineros mejor conectados. La escalabilidad se compra con requisitos de hardware, es decir, con menos nodos independientes. Las soluciones de capa 2 (como Lightning) intentan esquivar el trilema moviendo la mayoría de las operaciones fuera de la cadena.

7. LO QUE EL VÍDEO SIMPLIFICA

EN EL VÍDEO	CON PRECISIÓN
«Bitcoin es dinero, y poco más»	Admite multifirma, bloqueos temporales y condiciones con Taproot, pero sin bucles
«Ethereum ejecuta programas»	Turing-completo, con el gas como límite económico de cada ejecución
«Solana: nodos más caros»	≈ 24 núcleos y cientos de GB de RAM por validador (2026)
«Lightning va por encima de Bitcoin»	Canales 2 de 2 con transacciones de compromiso revocables y HTLC
«El trilema»	Compromiso de ingeniería, no un teorema

8. REFERENCIAS

- [1] Bitcoin Project. *Bitcoin Developer Guide: Transactions (Script)*.
<https://developer.bitcoin.org/devguide/transactions.html>
- [2] V. Buterin. *Ethereum Whitepaper*. 2014. <https://ethereum.org/es/whitepaper/>
- [3] G. Wood. *Ethereum: A Secure Decentralised Generalised Transaction Ledger* (Yellow Paper), y EIP-2929 / EIP-1559. <https://ethereum.github.io/yellowpaper/paper.pdf>
- [4] A. Yakovenko. *Solana: A new architecture for a high performance blockchain*. 2017.
<https://solana.com/solana-whitepaper.pdf>
- [5] Anza / Solana. *Validator requirements*. <https://docs.anza.xyz/operations/requirements>
- [6] J. Poon, T. Dryja. *The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments*. 2016.
<https://lightning.network/lightning-network-paper.pdf>
- [7] E. Lombrozo, J. Lau, P. Wuille. *BIP 141: Segregated Witness*.
<https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki>
- [8] A. M. Turing. «On Computable Numbers, with an Application to the Entscheidungsproblem». *Proceedings of the London Mathematical Society*, 1936.

Las capacidades de (2), los bloques por día y las cifras de gas se han calculado o tomado de las especificaciones citadas con un script propio.