

CRIPTO · SERIE 1 · PARTE 01 / 04

LA CADENA DE BLOQUES COMO ESTRUCTURA DE DATOS

Punteros hash, evidencia de manipulación, validación independiente por miles de nodos y por qué una cadena de huellas, por sí sola, no basta para que el pasado sea inmutable.

Complemento técnico de la guía «¿Qué es una blockchain?»
<https://amstech.dev/guias/bitcoin-s1/01-que-es-una-blockchain/>

@ams_tech

ÍNDICE

- 1 Resumen
- 2 El problema: dinero digital sin intermediario
- 3 Punteros hash y evidencia de manipulación
- 4 Qué comprueba un nodo completo
- 5 Por qué no basta con encadenar huellas
- 6 Qué garantiza y qué no
- 7 Lo que el vídeo simplifica
- 8 Referencias

1. RESUMEN

Una blockchain combina dos ideas: una **estructura de datos** en la que cada bloque contiene la huella criptográfica del anterior, y una **red de nodos** que guardan copias completas y aplican las mismas reglas de validación. La primera hace que cualquier modificación del pasado sea **detectable**; la segunda hace que nadie necesite fiarse de un servidor central.

Este dossier presenta el problema que resuelve (§2), formaliza la cadena de punteros hash y demuestra su propiedad de evidencia de manipulación (§3), describe qué comprueba un nodo completo (§4) y cuantifica por qué recalcular huellas, por sí solo, no protege nada: un ordenador doméstico rehace las huellas de los más de 970.000 bloques de Bitcoin en segundos (§5). Esa última observación es la pregunta que responde la prueba de trabajo, en el dossier 4 de la serie.

2. EL PROBLEMA: DINERO DIGITAL SIN INTERMEDIARIO

Un fichero digital se copia sin coste. Si una moneda digital fuera solo un fichero, podría gastarse dos veces: es el **problema del doble gasto**. La solución clásica es un intermediario (un banco) que lleva el registro único de quién tiene qué. Bitcoin [1] lo sustituye por un registro **público, replicado y ordenado** cuyas reglas cualquiera puede comprobar.

Dos líneas de trabajo previas confluyen en ese diseño:

- **Sellado de tiempo encadenado.** Haber y Stornetta propusieron en 1991 fechar documentos encadenando sus huellas, de modo que alterar un documento antiguo obligara a alterar todos los posteriores [2], y en 1993 añadieron árboles de Merkle para agrupar muchos documentos en cada sello [3]. El whitepaper de Bitcoin los cita expresamente.
- **Acuerdo entre partes que no se fían.** El problema de los **generales bizantinos** (Lamport, Shostak y Pease, 1982 [4]) formaliza cómo llegar a un acuerdo cuando algunos participantes pueden mentir. Los protocolos clásicos necesitan conocer de antemano quiénes son los participantes; Bitcoin funciona con participantes anónimos que entran y salen, y para eso necesita la prueba de trabajo.

3. PUNTEROS HASH Y EVIDENCIA DE MANIPULACIÓN

3.1 DEFINICIÓN

Sea H una función hash resistente a colisiones (SHA-256 aplicada dos veces en Bitcoin; ver el dossier 2). Una cadena de bloques es una sucesión

$$B_i = (h_{i-1}, \text{datos}_i), \quad h_i = H(B_i), \quad h_0 = \text{valor fijo.} \quad (1)$$

Cada h_{i-1} es un **puntero hash**: indica dónde está el bloque anterior y, a la vez, fija su contenido exacto [5]. Basta con conocer el último valor, h_n , para comprometerse con toda la historia.

3.2 PROPIEDAD

Proposición. Si alguien conoce el valor auténtico de h_n y le presentan una cadena bien enlazada (cada puntero es la huella del bloque anterior) $B'_1, \dots, B'_n$ con $B'_j \neq B_j$ para algún j que, aun así, termina en el mismo h_n , entonces se ha encontrado una **colisión** de H .

Idea de la demostración. Comparando las dos cadenas desde el final, sea $k \geq j$ el último índice en el que $B'_k \neq B_k$. Si $k = n$, tenemos $H(B'_n) = h_n = H(B_n)$ con $B'_n \neq B_n$: colisión. Si $k < n$, el bloque $k + 1$ coincide en las dos cadenas, así que contiene el mismo puntero: $H(B'_k) = h_k = H(B_k)$, de nuevo con entradas distintas. En ambos casos hay dos entradas distintas con la misma huella. $\square$

Como encontrar colisiones de SHA-256 cuesta del orden de 2^{128} operaciones (dossier 2), en la práctica **no se puede modificar el pasado sin que cambie la última huella**.

3.3 EJEMPLO CON HUELLAS REALES

La cadena de juguete del vídeo usa $h_i = \text{SHA256}(\text{«bloque } i \mid \text{datos} \mid h_{i-1}\text{»})$, mostrando solo los 4 primeros caracteres:

BLOQUE	DATOS	PUNTERO GUARDADO	HUELLA
1	Ana→Luis: 5	0000	d726...
2	Luis→Eva: 2	d726	bf2a...
3	Eva→Leo: 1	bf2a	6554...

Si se cambia el bloque 2 a Luis→Eva: 200, su huella pasa a 88c6... . El bloque 3 sigue guardando bf2a, que ya no coincide con la huella real del bloque 2: la manipulación es evidente.

4. QUÉ COMPRUEBA UN NODO COMPLETO

Un **nodo completo** descarga todos los bloques desde el primero (el bloque 0, de 2009) y valida cada uno de forma independiente [1][6]. Entre otras reglas, comprueba:

1. **Formato y enlace:** la cabecera está bien formada y su puntero coincide con la huella del bloque anterior.
2. **Prueba de trabajo:** la huella de la cabecera queda por debajo del objetivo vigente, y ese objetivo es el que manda el ajuste de dificultad (serie 2).
3. **Raíz de Merkle:** la raíz de la cabecera es la de las transacciones que contiene el bloque (dossier 3).
4. **Transacciones:** cada una gasta salidas que existen y no se han gastado antes, lleva firmas válidas y no crea más valor del que consume.
5. **Emisión:** la recompensa del minero no supera la permitida a esa altura más las comisiones.

4.1 EL MODELO UTXO

Bitcoin no lleva saldos por cuenta. El estado es el conjunto de **salidas no gastadas** (UTXO, *unspent transaction outputs*): cada transacción consume salidas anteriores y crea salidas nuevas. Comprobar un doble gasto se reduce a preguntar si una salida sigue en ese conjunto. Cada nodo reconstruye el conjunto procesando la cadena entera, sin fiarse de nadie.

4.2 QUÉ CADENA SE SIGUE

Si dos versiones compiten, los nodos siguen la cadena **válida con más trabajo acumulado**, la suma de la dificultad de sus bloques, no simplemente la más larga [1][6]. No hay votación: cada nodo aplica la misma regla a la misma información y llega a la misma conclusión.

5. POR QUÉ NO BASTA CON ENCADENAR HUELLAS

La proposición del §3 dice que el pasado no se puede cambiar **sin que cambie** h_n . Pero nada impide a un falsificador cambiar el bloque y **recalcular todas las huellas siguientes**, obteniendo una cadena coherente con otro h'_n . Si los nodos aceptaran cualquier cadena bien enlazada, bastaría con difundirla.

¿Cuánto cuesta ese recálculo? Un solo hilo de JavaScript en un ordenador doméstico calcula unas 500.000 huellas SHA-256 por segundo (dossier 4). Recalcular la doble huella de

$$971.712 \text{ cabeceras} \times 2 \text{ SHA-256} \approx 1,9 \cdot 10^6 \text{ operaciones} \Rightarrow \approx 4 \text{ segundos.} \quad (2)$$

Es decir, **la cadena por sí sola no protege nada**: solo garantiza que cualquier cambio es visible. Para que reescribir la historia sea inviable hace falta que cada huella sea **cara** de producir y que los nodos prefieran la cadena que más ha costado. Eso es la prueba de trabajo, y es la pregunta con la que acaba el vídeo.

6. QUÉ GARANTIZA Y QUÉ NO

GARANTIZA	NO GARANTIZA
Integridad: el registro no cambia sin que se note	Veracidad: un dato falso anotado correctamente queda como falso (serie 2, dossier 4)
Orden: las transacciones tienen un orden acordado	Anonimato: las direcciones son seudónimos; el historial es público y analizable
Disponibilidad: miles de copias independientes	Rapidez: un bloque cada 10 minutos de media
Verificabilidad: cualquiera puede comprobarlo todo	Reversibilidad: un pago enviado por error no se puede anular

7. LO QUE EL VÍDEO SIMPLIFICA

EN EL VÍDEO	CON PRECISIÓN
«Ningún banco ni gobierno puede borrar un pago»	Borrarlo exigiría rehacer la prueba de trabajo de todos los bloques posteriores más rápido que el resto de la red (dossier 4)
«Cada bloque guarda la huella del anterior»	La guarda en su cabecera; la huella se calcula sobre la cabecera, no sobre el bloque entero
«Miles de ordenadores lo comprueban todo»	Los nodos completos validan cada regla por su cuenta; no hay votación
Huellas de 4 caracteres	Simplificación visual; las reales tienen 64

8. REFERENCIAS

- [1] S. Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. 2008. <https://bitcoin.org/bitcoin.pdf>
- [2] S. Haber, W. S. Stornetta. «How to Time-Stamp a Digital Document». *Journal of Cryptology*, 3(2), 1991.
- [3] D. Bayer, S. Haber, W. S. Stornetta. «Improving the Efficiency and Reliability of Digital Time-Stamping». *Sequences II*, Springer, 1993.
- [4] L. Lamport, R. Shostak, M. Pease. «The Byzantine Generals Problem». *ACM Transactions on Programming Languages and Systems*, 4(3), 1982.
- [5] A. Narayanan, J. Bonneau, E. Felten, A. Miller, S. Goldfeder. *Bitcoin and Cryptocurrency Technologies*. Princeton University Press, 2016.
- [6] A. M. Antonopoulos, D. A. Harding. *Mastering Bitcoin*, 3.ª ed. O'Reilly, 2023.
- [7] Bitcoin Project. *Bitcoin Developer Guide: Block Chain*. https://developer.bitcoin.org/devguide/block_chain.html

Las huellas de §3 y la estimación de §5 se han calculado con un script propio. La altura de la cadena (bloque 971.712) corresponde al 16/10/2026.