

CRIPTO · SERIE 1 · PARTE 03 / 04

ANATOMÍA DE UN BLOQUE

La cabecera de 80 bytes campo a campo, el orden de bytes, la codificación compacta del objetivo, el árbol de Merkle y sus pruebas de inclusión, y la verificación completa del bloque génesis y de su titular de periódico.

Complemento técnico de la guía «Bloque 0: un secreto»

<https://amstech.dev/guias/bitcoin-s1/03-bloque-0-un-secreto/>

@ams_tech

ÍNDICE

1 Resumen

2 La cabecera

3 El objetivo en formato compacto

4 El árbol de Merkle

5 Verificación del bloque génesis

6 Lo que el vídeo simplifica

7 Referencias

1. RESUMEN

Un bloque de Bitcoin tiene dos partes: una **cabecera** de 80 bytes y una **lista de transacciones**. La huella del bloque se calcula solo sobre la cabecera, que resume las transacciones mediante la **raíz de un árbol de Merkle**.

Este dossier:

- describe los seis campos de la cabecera y su codificación (§2);
- explica cómo se descomprime el campo `bits` en el objetivo de la red (§3);
- formaliza el árbol de Merkle, sus pruebas de inclusión de tamaño logarítmico y una debilidad conocida de su construcción (§4);
- reconstruye el bloque génesis byte a byte. Su doble SHA-256 da la huella publicada, `00000000000019d6...`, y la doble SHA-256 de su única transacción coincide con la raíz de Merkle. Así se demuestra que el titular de *The Times* que lleva dentro es auténtico (§5).

2. LA CABECERA

2.1 LOS CAMPOS

DESPLAZAMIENTO	CAMPO	TAMAÑO	CONTENIDO
0	<code>version</code>	4 B	versión y señales de actualizaciones de protocolo
4	<code>prev_block</code>	32 B	doble SHA-256 de la cabecera anterior
36	<code>merkle_root</code>	32 B	raíz del árbol de Merkle de las transacciones
68	<code>time</code>	4 B	segundos desde el 1/1/1970 UTC (tiempo Unix)
72	<code>bits</code>	4 B	objetivo de la prueba de trabajo, en formato compacto (§3)
76	<code>nonce</code>	4 B	número libre que el minero varía

En total, $4 + 32 + 32 + 4 + 4 + 4 = 80$ bytes [1][2]. La huella del bloque es

$$h = \text{SHA256}(\text{SHA256}(\text{cabecera})), \quad (1)$$

y es la que el siguiente bloque guarda en `prev_block`.

2.2 ORDEN DE BYTES

Los enteros se codifican en **little-endian** (byte menos significativo primero) y las huellas se guardan **invertidas** respecto a cómo se muestran. La huella del génesis se escribe `000000000019d6...`, pero en los bytes del bloque siguiente aparece como `...d619000000000000`. Es una convención heredada del código original. Por eso los ceros de la huella, que «van delante» al mostrarla, son en realidad los bytes más significativos del número que se compara con el objetivo.

2.3 LA HORA

El campo `time` no tiene que ser exacto. Las reglas exigen que sea mayor que la **mediana de los 11 bloques anteriores** y que no esté más de 2 horas por delante del reloj del nodo que lo valida [2]. Esa holgura permite a los mineros usarlo como un segundo «nonce» (dossier 4).

3. EL OBJETIVO EN FORMATO COMPACTO

Comparar una huella de 256 bits con el objetivo exige el objetivo completo, también de 256 bits. Para ahorrar espacio, la cabecera lo guarda en 4 bytes con un formato de coma flotante propio: el primer byte es un **exponente** E y los otros tres, una **mantisa** M :

$$\text{objetivo} = M \cdot 256^{E-3}. \quad (2)$$

Para el génesis, `bits = 1d00ffff`: $E = 0x1d = 29$ y $M = 0x00ffff$, así que

$$\text{objetivo} = 0xffff \cdot 256^{26} = 0xffff \cdot 2^{208} = 00000000ffff0000 \dots 0000. \quad (3)$$

Es el objetivo **máximo** (dificultad 1). Exige que la huella empiece por 8 ceros hexadecimales y, por tanto, en torno a 2^{32} intentos por bloque. La **dificultad** de cualquier otro bloque se define respecto a él:

$$D = \frac{\text{objetivo máximo}}{\text{objetivo actual}}. \quad (4)$$

4. EL ÁRBOL DE MERKLE

4.1 CONSTRUCCIÓN

Las hojas son las huellas (doble SHA-256) de las transacciones, en el orden del bloque. Cada nivel se obtiene concatenando parejas y aplicando la doble SHA-256:

$$n_{\text{padre}} = \text{SHA256d}(n_{\text{izq}} \parallel n_{\text{der}}). \quad (5)$$

Si en un nivel hay un número impar de nodos, el último se empareja **consigo mismo**. La raíz es el único nodo del último nivel. Con una sola transacción, como en el bloque génesis, la raíz es la propia huella de esa transacción. La estructura la describió Merkle en el contexto de las firmas digitales [3].

4.2 PRUEBAS DE INCLUSIÓN

Para demostrar que una transacción t está en un bloque basta con dar las huellas **hermanas** del camino de su hoja a la raíz. El verificador recalcula el camino y lo compara con la `merkle_root` de la cabecera. Con n transacciones el árbol tiene $\lceil \log_2 n \rceil$ niveles:

TRANSACCIONES	HUELLAS EN LA PRUEBA	TAMAÑO
4	2	64 B
1.000	10	320 B
4.000	12	384 B
1.000.000	20	640 B

Es la base de la **verificación simplificada de pagos** (SPV) que describe el whitepaper [4]: una cartera ligera descarga solo las cabeceras (80 bytes por bloque, unos 78 MB para toda la cadena) y pide pruebas de inclusión de sus propias transacciones.

4.3 UNA DEBILIDAD CONOCIDA

Duplicar el último nodo de un nivel impar tiene un efecto secundario: una lista de transacciones $[a, b, c]$ y la lista $[a, b, c, c]$ producen **la misma raíz**. Esto dio lugar a una vulnerabilidad de denegación de servicio (CVE-2012-2459). Un atacante podía enviar una versión de un bloque válido con transacciones repetidas; el nodo la rechazaba por inválida y marcaba como inválida su huella, que era la del bloque bueno. Se corrigió detectando esos bloques «mutados» y descartándolos sin marcar su huella como inválida [5]. Es un buen ejemplo de que una estructura criptográficamente sólida puede tener problemas en los detalles de construcción.

5. VERIFICACIÓN DEL BLOQUE GÉNESIS

5.1 LA CABECERA

Con los valores publicados [1], la cabecera de 80 bytes del bloque 0, en hexadecimal y tal como se transmite, es:

Por campos: versión `1`; huella anterior, todo ceros (no hay bloque anterior); raíz de Merkle `4a5e1e4b...` (invertida); hora `0x495fab29` = 1.231.006.505 = **3/1/2009 18:15:05 UTC**; `bits` = `1d00ffff`; nonce `0x7c2bac1d` = 2.083.236.893. Su doble SHA-256, invertida para mostrarla, es

que coincide con la huella publicada del bloque 0. Empieza por **10 ceros** hexadecimales, dos más de los 8 que exigía el objetivo de (3).

El bloque contiene una sola transacción, la ***coinbase***: la que crea las monedas nuevas y paga al minero. Ocupa 204 bytes y su campo de entrada (que en las transacciones normales lleva la firma) es libre. En el génesis contiene, en ASCII:

¿Cómo sabemos que no se ha añadido después? Porque la doble SHA-256 de esos 204 bytes es exactamente `4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b`, la raíz de Merkle de la cabecera, y la cabecera tiene la prueba de trabajo de 2009. Cambiar una sola letra del titular cambiaría la raíz, la cabecera y su huella.

El titular cumple una función de **sello de tiempo**: demuestra que el bloque no pudo crearse antes del 3 de enero de 2009, y descarta que se minara en secreto una cadena por adelantado. También se lee como una declaración de intenciones en plena crisis bancaria.

La salida de la coinbase vale 5.000.000.000 satoshis, es decir, **50 BTC**. Sin embargo, el código original no añadió esa transacción a la base de datos de salidas disponibles (UTXO; dossier 1), así que los nodos rechazan cualquier intento de gastarla. Corregirlo exigiría cambiar las reglas de consenso, y nunca se ha hecho.

6. LO QUE EL VÍDEO SIMPLIFICA

EN EL VÍDEO	CON PRECISIÓN
«La cabecera: huella anterior, hora, objetivo y nonce»	Más la versión y la raíz de Merkle: seis campos, 80 bytes
«El objetivo de la red»	Un entero de 256 bits codificado en 4 bytes con exponente y mantisa (§3)
«Cambias un pago y cambia todo»	Cambia la hoja, todo su camino hasta la raíz, la cabecera y la huella del bloque
«Un titular del mismo día»	Demuestra que el bloque es posterior al 3/1/2009, no la fecha exacta
Árbol del vídeo	Transacciones de ejemplo, no las de un bloque real

7. REFERENCIAS

- [1] Bitcoin Project. *Bitcoin Developer Reference: Block Chain (block headers, compact nBits)*. https://developer.bitcoin.org/reference/block_chain.html
- [2] A. M. Antonopoulos, D. A. Harding. *Mastering Bitcoin*, 3.ª ed., caps. 11–12. O'Reilly, 2023.
- [3] R. C. Merkle. «A Digital Signature Based on a Conventional Encryption Function». *CRYPTO '87*, LNCS 293, Springer, 1988.
- [4] S. Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*, sección 8. 2008. <https://bitcoin.org/bitcoin.pdf>
- [5] National Vulnerability Database. *CVE-2012-2459*. <https://nvd.nist.gov/vuln/detail/CVE-2012-2459>
- [6] National Institute of Standards and Technology. *FIPS PUB 180-4: Secure Hash Standard (SHS)*. 2015. <https://doi.org/10.6028/NIST.FIPS.180-4>

La cabecera, la huella del génesis y la comprobación de la coinbase se han calculado con un script propio a partir de los valores publicados del bloque 0.