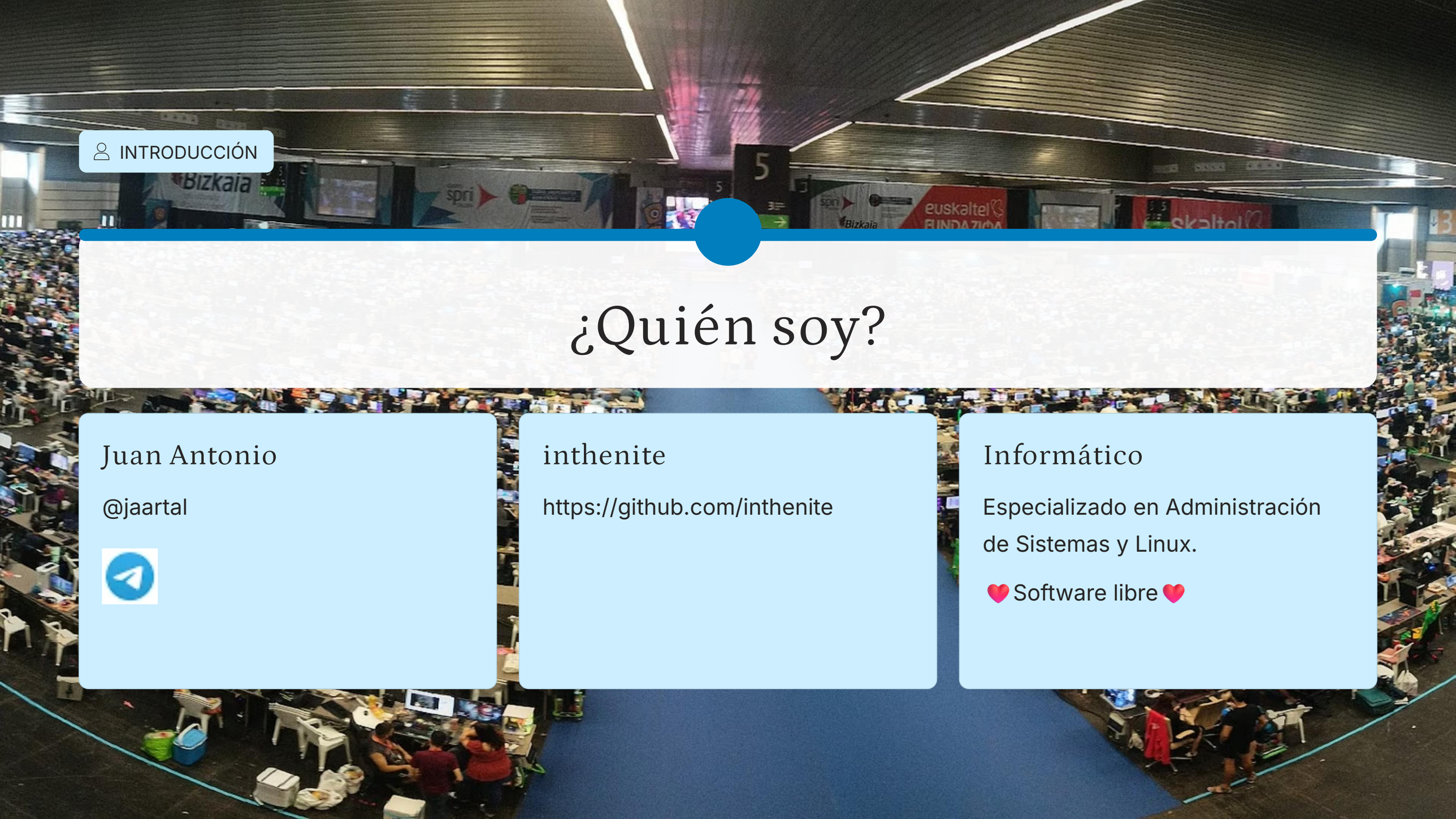
MEETUP 05/02/2026

Bitcoin desde el nodo: explorando la cadena de bloques

Primero contacto con el funcionamiento real de Bitcoin a través de un nodo propio. Herramientas que permiten consultar y analizar la blockchain desde tu nodo.





INTRODUCCIÓN

¿Quién soy?

Juan Antonio

@jaartal

inthenite

https://github.com/inthenite

Informático

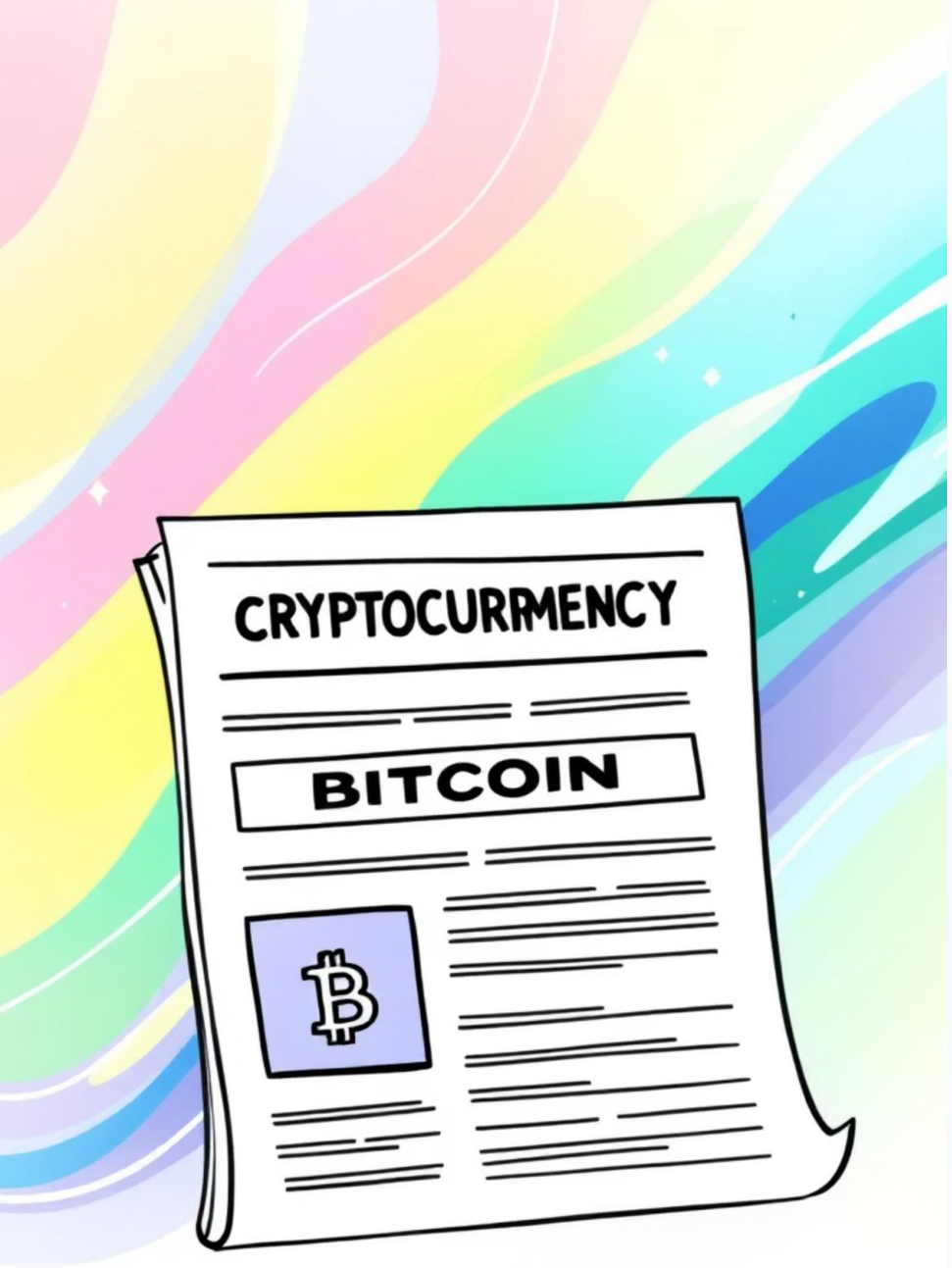
Especializado en Administración de Sistemas y Linux.

❤️Software libre❤️

Acto 1:

La mentira cómoda

APIs, exploradores, JSON, límites, privacidad falsa.



MI VIAJE

¿Cómo he llegado hasta aquí?

La cadena de bloques es pública

Esta característica fundamental de Bitcoin despertó mi curiosidad: toda la información está ahí, esperando ser explorada y analizada por quien sepa cómo hacerlo.

Noticias amarillistas que des-inspiran

- "No pararé hasta recuperar el disco duro con US$783 millones en bitcoins que tiró mi exnovia a la basura"
- "Tiene miles de bitcoins pero no recuerda la contraseña: si falla lo pierde todo"
- "¿Quién es la persona que posee más bitcoin en el mundo en este 2025?"

DESARROLLO

Vamos a programar

Existen múltiples plataformas que ofrecen acceso a datos blockchain mediante APIs o interfaces web. Estas herramientas son el punto de partida para cualquier desarrollador que quiera trabajar con Bitcoin. Fácil y rápido.



Blockstream.info

Explorador completo con API robusta



Mempool.space

Visualización avanzada del mempool



Blockcypher.com

APIs multi-blockchain flexibles



Blockchain.info

Pionero en exploradores Bitcoin



Chain.so

Soporte para múltiples altcoins



Blockchair.com

Motor de búsqueda blockchain universal

7

Linux y Python

Y con unos sencillos conocimientos de programación ya se pueden obtener datos rápidamente.

EJEMPLOS PRÁCTICOS

Vamos a programar

Veamos ejemplos reales de cómo estas APIs funcionan en la práctica. Se hacen peticiones a las APIs de diferentes webs y después observamos las respuestas JSON que obtenemos al consultar información de transacciones y bloques.

```python
def saldo_blockstream(address):
    url = f"https://blockstream.info/api/address/{address}"
    r = requests.get(url)
    r.raise_for_status()
    data = r.json()
    recibido = data["chain_stats"]["funded_txo_sum"] / 1e8
    gastado = data["chain_stats"]["spent_txo_sum"] / 1e8
    return recibido - gastado

1 usage
def saldo_mempool(address):
    url = f"https://mempool.space/api/address/{address}"
    r = requests.get(url)
    r.raise_for_status()
    data = r.json()
    recibido = data["chain_stats"]["funded_txo_sum"] / 1e8
    gastado = data["chain_stats"]["spent_txo_sum"] / 1e8
    return recibido - gastado

1 usage
def saldo_blockcypher(address):
    url = f"https://api.blockcypher.com/v1/btc/main/addrs/{address}"
    r = requests.get(url)
    r.raise_for_status()
    data = r.json()
    return data["final_balance"] / 1e8
```

Programando en python usando las apis de blockstream.info, mempool.space y blockcypher.com

```python
1 usage
def saldo_blockchain(address):
    url = f"https://blockchain.info/rawaddr/{address}"
    r = requests.get(url)
    r.raise_for_status()
    data = r.json()
    return data["final_balance"] / 1e8

1 usage
def saldo_sochain(address):
    return None # Es un api de pago
    url = f"https://sochain.com/api/v2/get_address_balance/BTC/{address}"
    r = requests.get(url)
    r.raise_for_status()
    data = r.json()
    return float(data["data"]["confirmed_balance"])

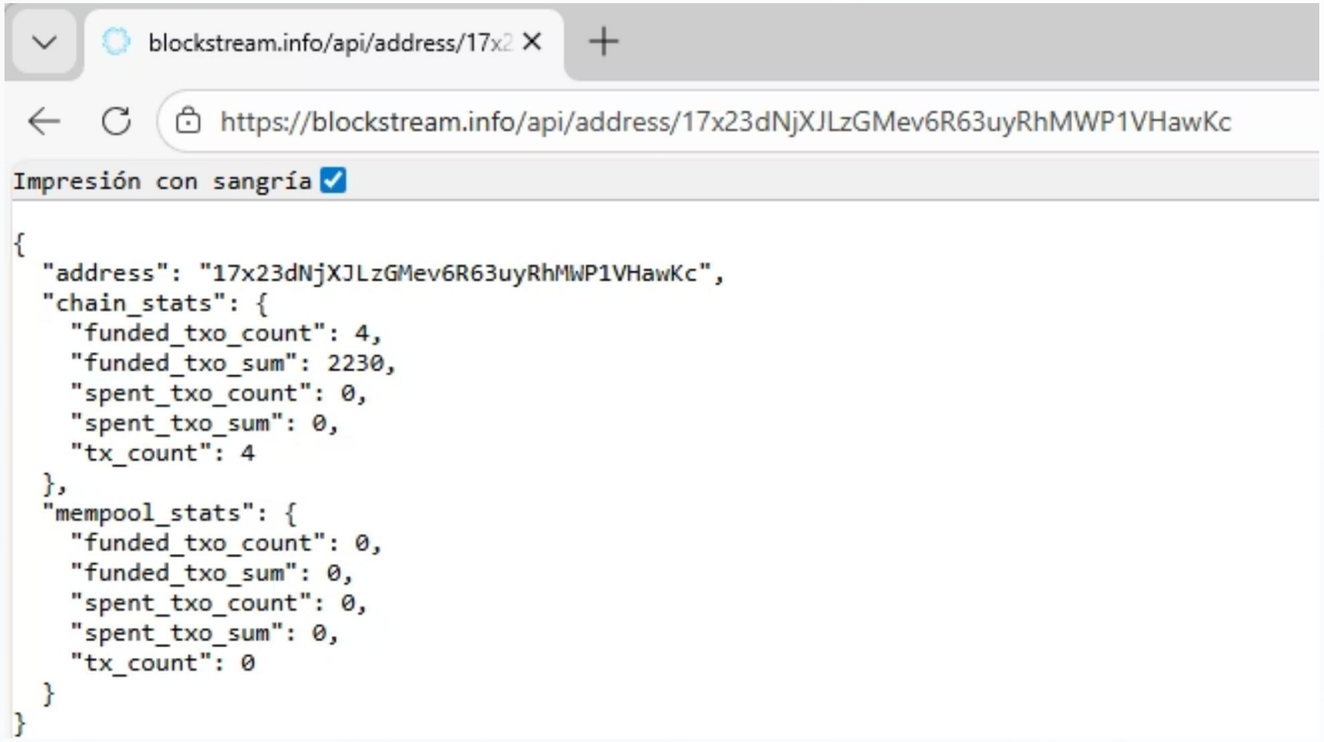
def saldo_blockchair(address):
    url=f"https://api.blockchair.com/bitcoin/dashboards/address/{address}"
    # No funciona bien
    return None
```

Aquí con blockchain.info, sochain.com y blochchair.com.

EJEMPLOS PRÁCTICOS

¿Qué es JSON?

JSON: texto estructurado, fácil de leer por humanos y máquinas. Este es el ejemplo de lo que nos devolvería una de las APIS previas



Empiezan los problemas porque el saldo debería de ser 5000002230 sats.

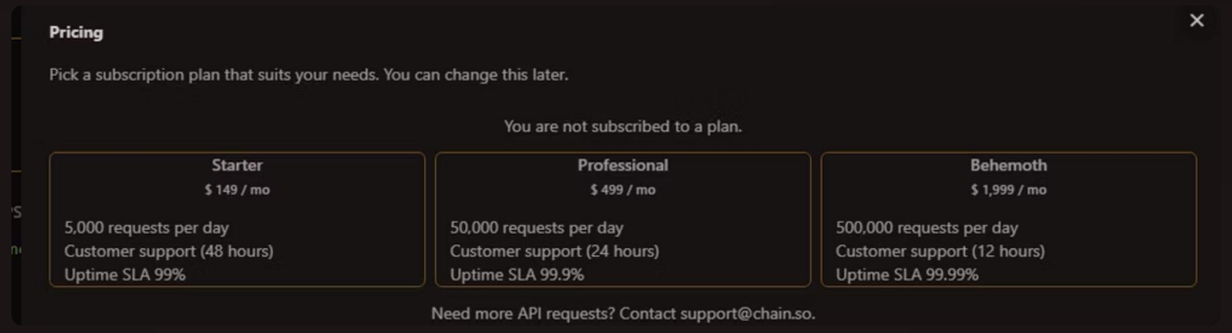
Pros y contras de estas webs

Beneficios

- ¿Es gratis? Te enganchan con lo gratis. Hay varias sin identificación, consulta "anónima"
- Sin infraestructura propia: No necesitas tener un nodo en casa encendido 24 horas al día, 7 días a la semana
- Acceso inmediato: Comienza a desarrollar en minutos sin sincronización previa

Limitaciones

- Modelos de pago: Algunos son de pago directamente desde el inicio (como sochain.com)
- Restricciones de uso: Los servicios gratuitos tienen limitaciones en número de peticiones y funcionalidades
- Dependencia externa: Confías en la disponibilidad y fiabilidad de un tercero
- Privacidad: Ninguna, desconocida, ¿Qué harán con tus datos? ¿Por qué consultaste esta dirección?

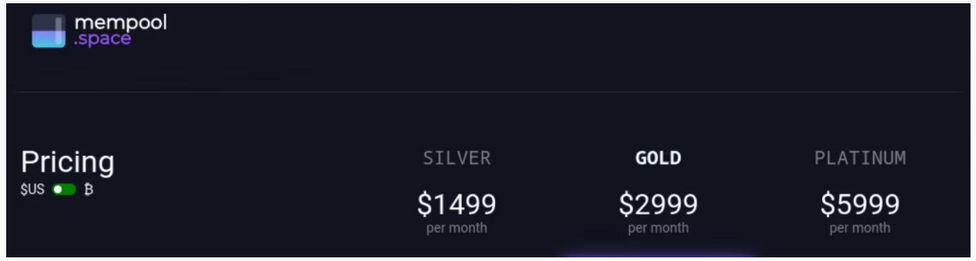


A PAGAR

Al final toca pagar

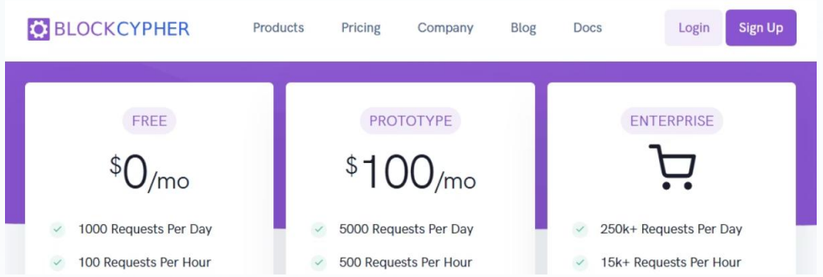
Todos tienen buena documentación

Pero te piden mucha pasta



Consideraciones

Para uso personal son caros y muy pocas peticiones al día.



Acto 2:

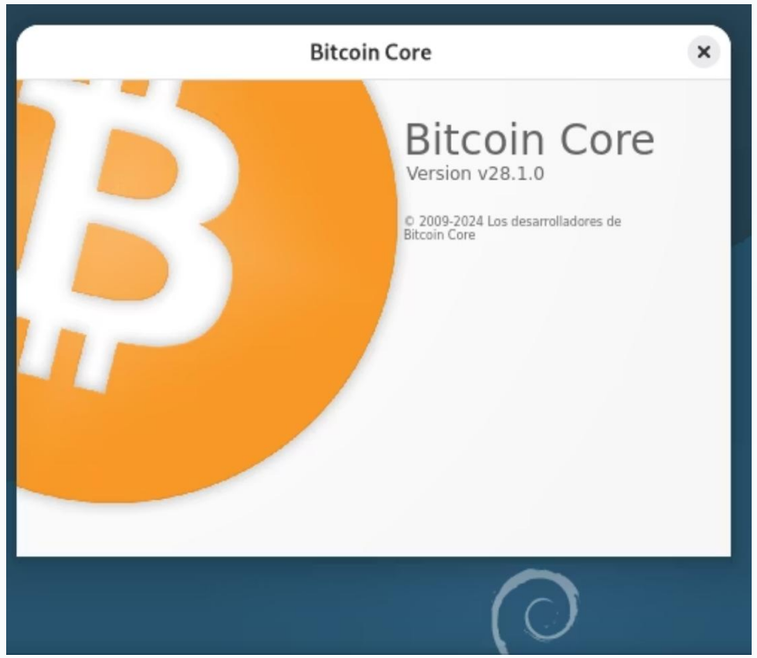
La soberanía duele
Bitcoin Core, UTXOs, indexación, consultas eternas.

t 49598812 dic 13 18:43 bitcoin-28.1-x86_64-linux-gnu.tar.qz

SOLUCIÓN CASI DEFINITIVA

Bitcoin Core

El día 13 de diciembre de 2025 decidí dar un giro a mi forma de acceder a los datos. Accederé a MIS datos.



Ventajas de tu propio nodo

- Soberanía total: Verificas todas las transacciones tú mismo, nadie puede bloquear tus transacciones si las emites desde tu nodo
- Contribución a la red: Ayudas a descentralizar Bitcoin

Ventajas como desarrollador

- Sin límites: Consultas ilimitadas sin restricciones de terceros
- Velocidad: acceso "rápido" a cualquier dato de la blockchain
- Privacidad máxima: Servidores de terceros pueden almacenar tu dirección IP, direcciones consultadas e historial de uso.

Doc de desarrollo: https://bitcoincore.org/en/doc/28.0.0/rpc/blockchain/getblock/

Bitcoin Core: Consideraciones técnicas

Antes de lanzarte a montar tu nodo, es crucial entender los requisitos y opciones de configuración que determinarán cómo funcionará tu instalación.

Pruned o Full Node (Podado o completo)

Nodo completo: Almacena toda la blockchain histórica (~832GB). Nodo podado: Solo mantiene los datos más recientes necesarios para validar. Para mi caso en particular sólo me servía un nodo completo. "Full Archival Node"

Tiempo de descarga

La sincronización inicial puede tardar desde horas hasta días dependiendo de tu conexión, hardware y si optas por un nodo completo. Paciencia y una buena conexión a internet son esenciales.

Tipo de disco

Planifica con antelación (yo no lo hice): un SSD NVME de al menos 2TB es recomendable para nodo completo. Para nodos podados, 20GB pueden ser suficientes, pero no tendrás datos antiguos. El mínimo para un nodo podado son 288 bloques (unos 2 días).

Recomendación: Si eres un impaciente nodo podado. Pero si quieres programar y privacidad Nodo Completo.

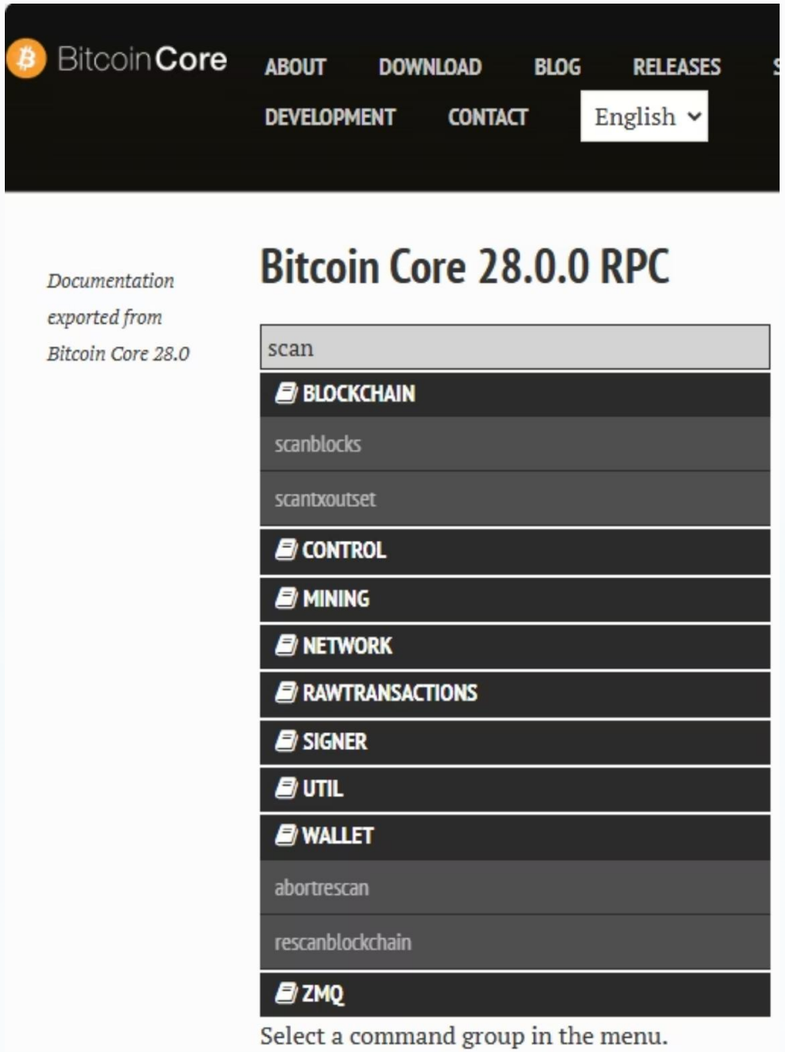
Bitcoin Core: Consideraciones técnicas

Configuración del nodo:

Fichero ~/.bitcoin/bitcoin.conf

```
server=1
rpcallowip=127.0.0.1
txindex=1
zmqpubhashblock=tcp://127.0.0.1:8433
zmqpubhashtx=tcp://127.0.0.1:8433
```

Bitcoin Core: Consulta de saldo de direcciones



getbalance (28.0.0 RPC)

```
getbalance ( "dummy" minconf include_watchonly avoid_reuse )

Returns the total available balance.
The available balance is what the wallet considers currently spendable, and is
thus affected by options which limit spendability such as -spendzeroconfchange.

Arguments:
1. dummy                (string, optional) Remains for backward compatibility. Must be excluded
or set to "*".
2. minconf              (numeric, optional, default=0) Only include transactions confirmed at
least this many times.
3. include_watchonly    (boolean, optional, default=true for watch-only wallets, otherwise
false) Also include balance in watch-only addresses (see 'importaddress')
4. avoid_reuse          (boolean, optional, default=true) (only available if avoid_reuse wallet
flag is set) Do not include balance in dirty outputs; addresses are considered dirty if they
have previously been used in a transaction.

Result:
n    (numeric) The total amount in BTC received for this wallet.

Examples:

The total amount in the wallet with 0 or more confirmations
> bitcoin-cli getbalance

The total amount in the wallet with at least 6 confirmations
> bitcoin-cli getbalance "*" 6

As a JSON-RPC call
> curl --user myusername --data-binary '{"jsonrpc": "2.0", "id": "curltest", "method":
"getbalance", "params": ["*", 6]}' -H 'content-type: application/json' http://127.0.0.1:8332/
```

Tenemos una documentación muy buena sobre cómo utilizar RPC para Bitcoin Core

Bitcoin Core: Consulta de saldo de direcciones

No, en Bitcoin una dirección no tiene saldo. Es la suma de lo que te han enviado y todavía no has gastado. Esto no es como el saldo de la cuenta de un banco.

¿Qué es lo que te han enviado? Un UTXO - Unspent transaction output

El conjunto completo de todos los Bitcoin que se pueden gastar son el UTXO set.

Bitcoin Core: Consulta de "saldo" de direcciones

“Consultar saldo en Bitcoin Core NO es trivial”

Método	Comando	¿Requiere importaddres?	¿muestra saldo neto actual?	
getreceivedbyaddress (WALLET)	getreceivedbyaddress "dirección" [minconf]	Sí	No (muestra recibido total)	Devuelve la cantidad total recibida por una dirección
listunspent (WALLET)	listunspent 1 9999999 "[\"dirección\"]"	Sí	Sí (sumar amounts)	Devuelve una lista de unspent transaction outputs (UTXO)
getaddressinfo (WALLET)	getaddressinfo "dirección"	Sí	No	Da información sobre la dirección, algunos detalles sólo si está en la wallet activa
scantxoutset (BLOCKCHAIN)	scantxoutset start "[{\"desc\": \"addr(dirección)\"}]"	No	Sí	Escanea el "unspent transaction output set" para las direcciones que se pongan
listreceivedbyaddress (WALLET)	listreceivedbyaddress 1 true true	Sí	No	Muestra el saldo de las direcciones de tu wallet

Limitaciones de indexación en Bitcoin Core

Estructura de indexación nativa

- Bloques: Cabeceras de bloque y relaciona la altura con el hash de ese bloque
- UTXOs: Todas las salidas no gastadas (UTXO set)
- Transacciones: Todas las transacciones por su txid incluyendo transacciones ya gastadas (txindex=1)

Importante: Bitcoin Core no indexa direcciones de forma nativa, lo que dificulta las consultas directas por dirección.

Comando bitcoin-cli

```
$ bitcoin-cli scantxoutset start
'[{"desc":"addr(17x23dNjXJLzGMev6R63uyRhMWP1VHawKc)"}]'
```

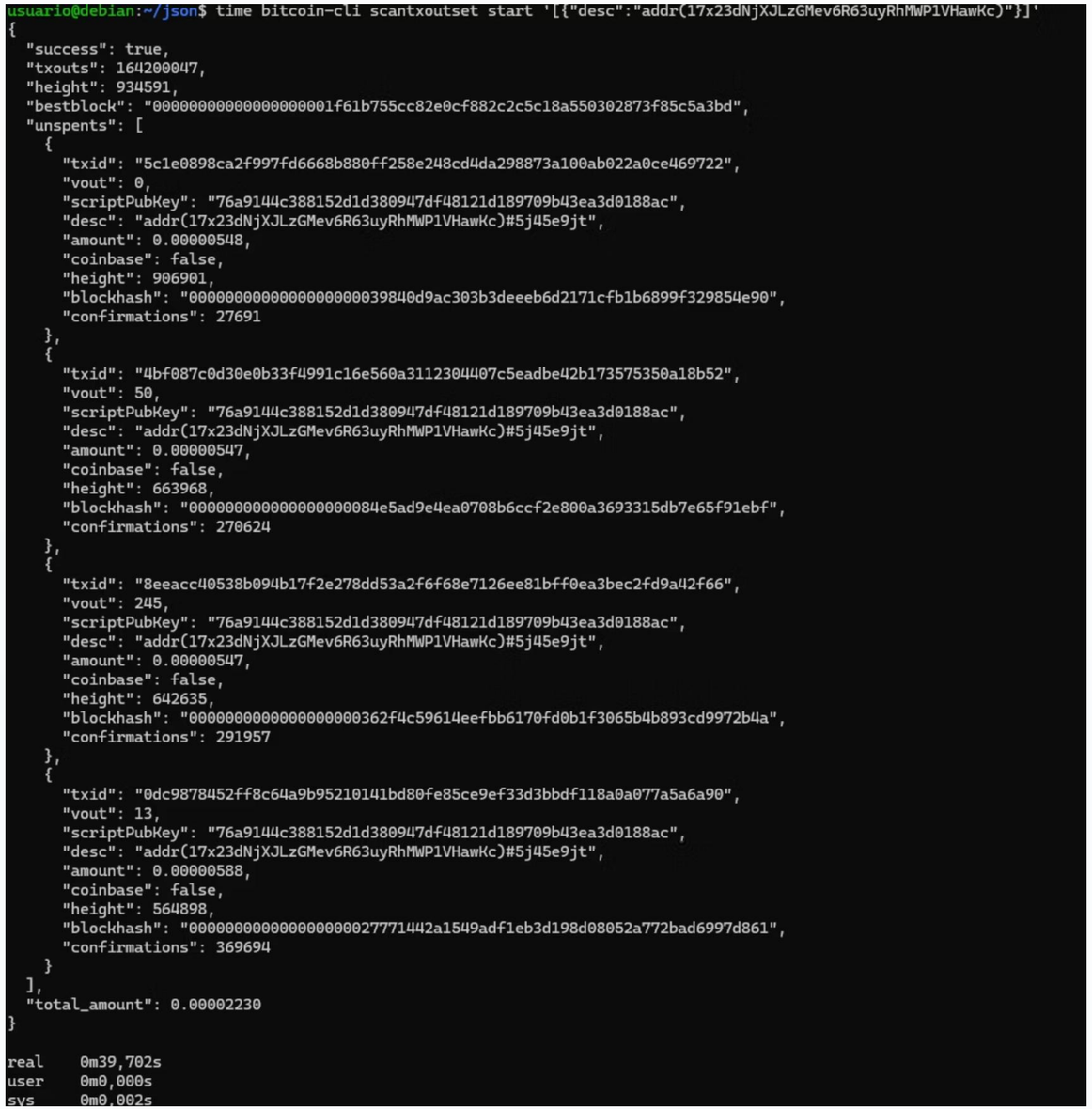
Es el único comando que permite obtener información sin necesidad de importación previa. Te permite escanear el UTXO set para encontrar saldos asociados a una dirección específica, pero es un proceso computacionalmente costoso.

Indexar es como crear un índice o catálogo que te permite encontrar cosas rápidamente, sin tener que revisar todo desde el principio. En un diccionario las palabras están en orden alfabético → eso es un índice. En bitcoin gracias a los índices se puede encontrar un bloque concreto, localizar una transacción o comprobar si un dinero ya se usó, sin tener que revisar toda la historia cada vez.

Resultados de consulta con Bitcoin Core

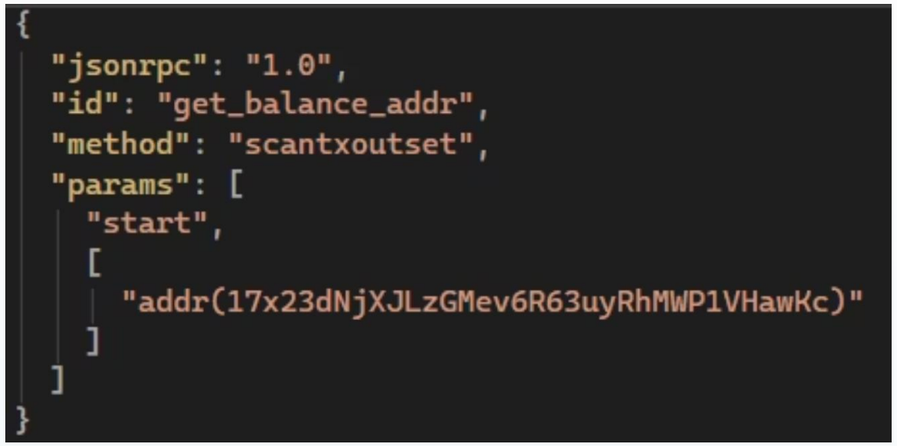
Cuando ejecutamos `scantxoutset`, Bitcoin Core analiza todo el conjunto de UTXOs disponibles. Este proceso puede tardar varios minutos dependiendo del hardware.

La respuesta incluye información detallada sobre el saldo total, los UTXOs específicos encontrados y metadatos sobre el proceso de escaneo realizado.



EJEMPLOS PRÁCTICOS

¿Y con rpc?



Ejemplo de petición de información a un servidor bitcoin-core mediante json-rpc

Este ejemplo lo ejecutaríamos así:

curl --user cookie:$(cat ~/.bitcoin/.cookie) --data-binary '@get_address_balance.json' -H 'content-type: application/json' http://127.0.0.1:8332/

Y después de unos 35 segundos nos daría esta información

```
{
  "result": {
    "success": true,
    "txouts": 164208493,
    "height": 934512,
    "bestblock": "00000000000000000012404de7f36d97c40ae649e859ce26fdfa8df3dc95f64",
    "unspents": [
      {
        "txid": "5c1e0898ca2f997fd6668b880ff258e248cd4da298873a100ab022a0ce469722",
        "vout": 0,
        "scriptPubKey": "76a9144c388152d1d380947df48121d189709b43ea3d0188ac",
        "desc": "addr(17x23dNjXJLzGMev6R63uyRhMWP1VHawKc)#5j45e9jt",
        "amount": 0.00000548,
        "coinbase": false,
        "height": 906901,
        "blockhash": "00000000000000000000039840d9ac303b3deeeb6d2171cfb1b6899f329854e90",
        "confirmations": 27612 },
      {
        "txid": "4bf087c0d30e0b33f4991c16e560a3112304407c5eadbe42b173575350a18b52",
        "vout": 50,
        "scriptPubKey": "76a9144c388152d1d380947df48121d189709b43ea3d0188ac",
        "desc": "addr(17x23dNjXJLzGMev6R63uyRhMWP1VHawKc)#5j45e9jt",
        "amount": 0.00000547,
        "coinbase": false,
        "height": 663968,
        "blockhash": "000000000000000000084e5ad9e4ea0708b6ccf2e800a3693315db7e65f91ebf",
        "confirmations": 270545 },
      {
        "txid": "8eeacc40538b094b17f2e278dd53a2f6f68e7126ee81bff0ea3bec2fd9a42f66",
        "vout": 245,
        "scriptPubKey": "76a9144c388152d1d380947df48121d189709b43ea3d0188ac",
        "desc": "addr(17x23dNjXJLzGMev6R63uyRhMWP1VHawKc)#5j45e9jt",
        "amount": 0.00000547,
        "coinbase": false,
        "height": 642635,
        "blockhash": "000000000000000000362f4c59614eefbb6170fd0b1f3065b4b893cd9972b4a",
        "confirmations": 291878 },
      {
        "txid": "0dc9878452ff8c64a9b95210141bd80fe85ce9ef33d3bbdf118a0a077a5a6a90",
        "vout": 13,
        "scriptPubKey": "76a9144c388152d1d380947df48121d189709b43ea3d0188ac",
        "desc": "addr(17x23dNjXJLzGMev6R63uyRhMWP1VHawKc)#5j45e9jt",
        "amount": 0.00000588,
        "coinbase": false,
        "height": 564898,
        "blockhash": "000000000000000000027771442a1549adf1eb3d198d08052a772bad6997d861",
        "confirmations": 369615}
    ],
    "total_amount": 0.0000223 },
  "error": null,
  "id": "get_balance_addr"
}
```

ME FALTAN BTC

Problemas para obtener el saldo

La dirección que hemos utilizado aparece inicialmente en el bloque 30, pero no aparece en ese formato. Está derivada de su clave pública, ya que entonces el formato que se utilizaba era P2PK (Pay to public key) pk(042cf59fafd089a348c5834283573608e89a305c60a034604c7d22dde50998f1b9bb74681986ca1884a6b1df8ce7f1b79a2277057de855a634626e7a5851c1e716) Esta dirección minó 50 BTC, pero los movimientos que hemos obtenido corresponden a cuando se usa su clave derivada Legacy P2PKH (Pay to public key hash)

Los exploradores te engañan. Convierten de P2PK a P2PKH y luego unen los datos.

```
  "tx": [
    {
      "txid": "f5e26c8b82401c585235c572ba8265f16f7d9304ed8e31c198eab571754f5331",
      "hash": "f5e26c8b82401c585235c572ba8265f16f7d9304ed8e31c198eab571754f5331",
      "version": 1,
      "size": 134,
      "vsize": 134,
      "weight": 536,
      "locktime": 0,
      "vin": [
        {
          "coinbase": "04ffff001d0121",
          "sequence": 4294967295
        }
      ],
      "vout": [
        {
          "value": 50.00000000,
          "n": 0,
          "scriptPubKey": {
            "asm": "042cf59fafd089a348c5834283573608e89a305c60a034604c7d22dde50998f1b9bb74681986ca1884a6b1df8ce7f1b79a2277057de855a634626e7a5851c1e716 OP_CHECKSIG",
            "desc": "pk(042cf59fafd089a348c5834283573608e89a305c60a034604c7d22dde50998f1b9bb74681986ca1884a6b1df8ce7f1b79a2277057de855a634626e7a5851c1e716)#tafyhwq2",
            "hex": "41042cf59fafd089a348c5834283573608e89a305c60a034604c7d22dde50998f1b9bb74681986ca1884a6b1df8ce7f1b79a2277057de855a634626e7a5851c1e716ac",
            "type": "pubkey"
          }
        }
      ],
      "hex": "0100000001000000000000000000000000000000000000000000000000000000000000000000ffffffff0704ffff001d0121ffffffff0100f2052a010000004341042cf59fafd089a348c5834283573608e89a305c60a034604c7d2
b1df8ce7f1b79a2277057de855a634626e7a5851c1e716ac00000000"
    }
  ]
}
usuario@debian:~$ time bitcoin-cli scantxoutset start '["pk(042cf59fafd089a348c5834283573608e89a305c60a034604c7d22dde50998f1b9bb74681986ca1884a6b1df8ce7f1b79a2277057de855a634626e7a5851c1e716)"]'
{
  "success": true,
  "txouts": 164232209,
  "height": 934749,
  "bestblock": "000000000000000001e0567cb6204efe2735ff27237beeb9ee2b2612f38559",
  "unspents": [
    {
      "txid": "f5e26c8b82401c585235c572ba8265f16f7d9304ed8e31c198eab571754f5331",
      "vout": 0,
      "scriptPubKey": "41042cf59fafd089a348c5834283573608e89a305c60a034604c7d22dde50998f1b9bb74681986ca1884a6b1df8ce7f1b79a2277057de855a634626e7a5851c1e716ac",
      "desc": "pk([4c388152]042cf59fafd089a348c5834283573608e89a305c60a034604c7d22dde50998f1b9bb74681986ca1884a6b1df8ce7f1b79a2277057de855a634626e7a5851c1e716)#607z6nx6",
      "amount": 50.00000000,
      "coinbase": true,
      "height": 30,
      "blockhash": "00000000bc919cfb64f62de736d55cf79e3d535b474ace256b4fbb56073f64db",
      "confirmations": 934720
    }
  ],
  "total_amount": 50.00000000
}

real    0m37.904s
```

RESUMEN DE BITCOIN CORE

¿Pero puedo obtener el saldo?

Sí, pero no.

Puedo obtener el saldo, pero aún activando txindex=1 le cuesta unos 40 segundos obtener el saldo.

Y además si esa dirección tiene UTXOs de P2PK y P2PKH no se muestran.

Siguiente paso - Electrum

Acto 3:

El salto técnico

Fulcrum, direcciones, NVME, scripthash, rendimiento real.

Fulcrum: servidor Electrum

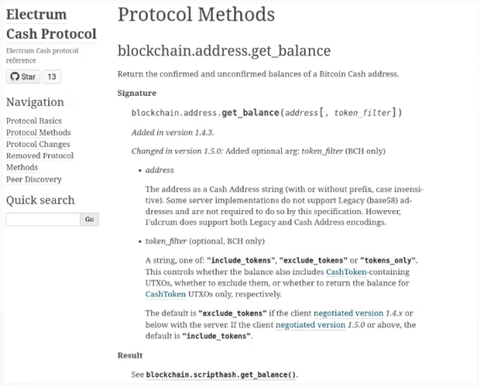
Fulcrum es un servidor Electrum de alto rendimiento. Es una alternativa a Bitcoin Core para consultar el saldo de una dirección. Las consultas se hacen por script (hash del script)

Ventaja principal de Fulcrum

Con Fulcrum, el saldo sí se puede consultar para cualquier dirección, porque Electrum indexa por scripthash, no depende del UTXO set como Bitcoin Core.

Esta indexación adicional permite consultas instantáneas y eficientes sobre cualquier dirección de la red.

Mempool.space utiliza internamente un servidor Electrum. Da 3 alternativas y Fulcrum es una de ellas.



Electrum Cash Protocol

Electrum Cash protocol reference

Star 13

Navigation

Protocol Basics
Protocol Methods
Protocol Changes
Removed Protocol Methods
Peer Discovery

Quick search

Go

Protocol Methods

blockchain.address.get_balance

Return the confirmed and unconfirmed balances of a Bitcoin Cash address.

Signature

blockchain.address.get_balance(address[, token_filter])

Added in version 1.4.3.

Changed in version 1.5.0: Added optional arg: token_filter (BCH only)

- address

The address as a Cash Address string (with or without prefix, case insensitive). Some server implementations do not support Legacy (base58) addresses and are not required to do so by this specification. However, Fulcrum does support both Legacy and Cash Address encodings.

- token_filter (optional, BCH only)

A string, one of: "include_tokens", "exclude_tokens" or "tokens_only". This controls whether the balance also includes CashToken-containing UTXOs, whether to exclude them, or whether to return the balance for CashToken UTXOs only, respectively.

The default is "exclude_tokens" if the client negotiated version 1.4.x or below with the server. If the client negotiated version 1.5.0 or above, the default is "include_tokens".

Result

See blockchain.scripthash.get_balance().

Por dirección no funciona, que es sólo para BCH

blockchain.scripthash.get_balance

Return the confirmed and unconfirmed balances of a script hash.

Signature

```
blockchain.scripthash.get_balance(scripthash[, token_filter])
```

Added in version 1.1.

Changed in version 1.5.0: Added optional arg: *token_filter* (BCH

scripthash

The script hash as a hexadecimal string.

token_filter (optional, BCH only)

A string, one of: **"include_tokens"**, **"exclude_tokens"** or "
This controls whether the balance also includes CashToken
UTXOs, whether to exclude them, or whether to return the
CashToken UTXOs only, respectively.

The default is **"exclude_tokens"** if the client negotiated ve
below with the server. If the client negotiated version *1.5.0*
default is **"include_tokens"**.

Result

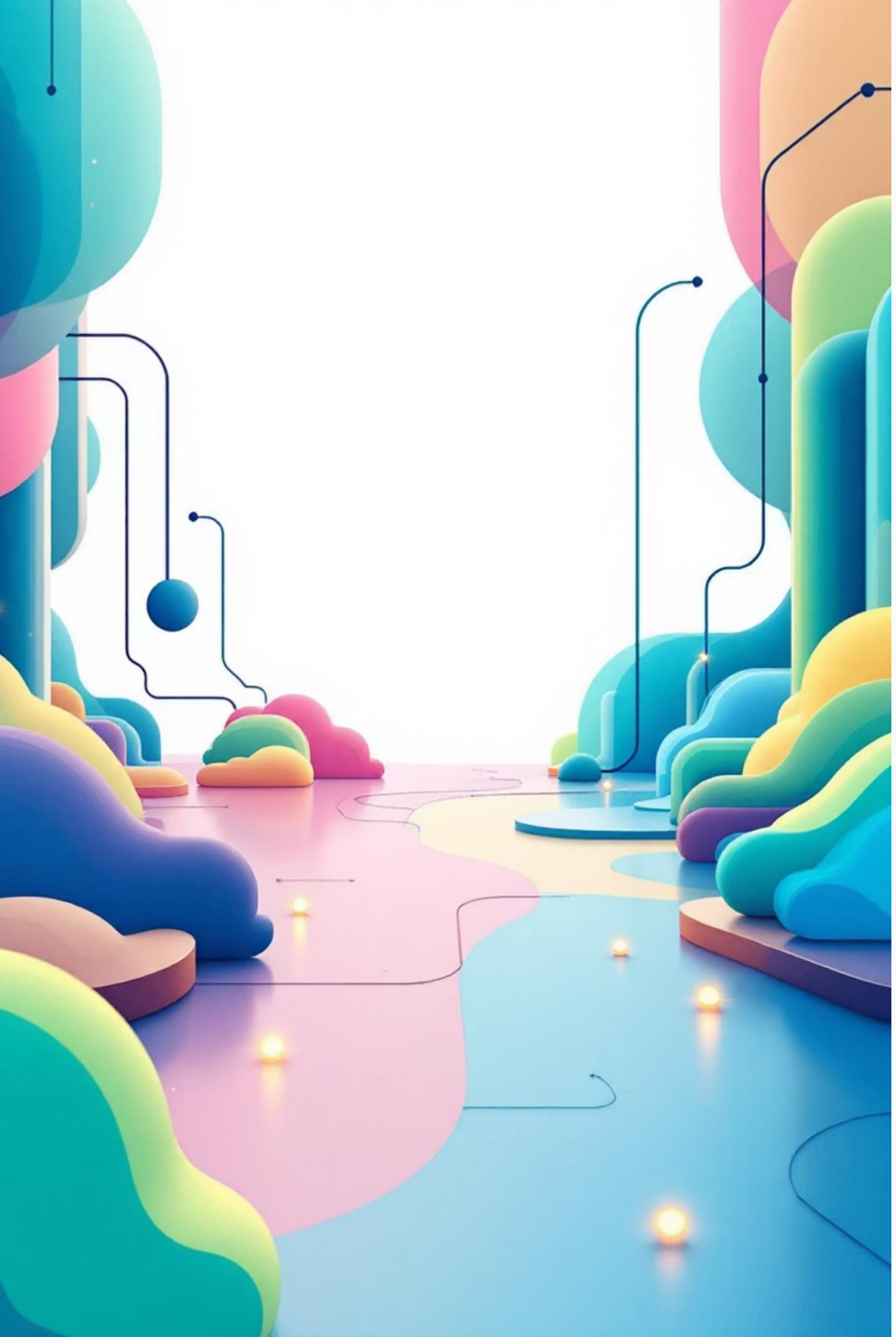
A dictionary with keys *confirmed* and *unconfirmed*. The value
appropriate balance in satoshis. Note that on BCH, be sure to s
priate *token_filter* argument to retrieve the balance either incl
ting tokens or for tokens only.

Result Example

Desde clave pública a dirección/script/hash

Y os cuento qué es una clave privada, pública, hash, dirección y script. [PELIGRO NO LEER, ¡QUÉ LÍO!] Formas distintas hay de gastar bitcoin

Paso / Método	P2PK (2009)	P2PKH (2009)	P2SH (2012)	P2WPKH (2017)	P2WSH (2017)	P2TR (2021)
Parte de clave pública	✔	✔	⚠ No siempre	✔	⚠ No siempre	✔ (x-only)
Formato de pubkey	Completa (65B)	Comprimida (33B)	Una o varias	Comprimida (33B)	Una o varias	X-only (32B)
SHA256(pubkey)	❌	✔	✔ (redeemScript)	✔	❌	❌
RIPEMD160	❌	✔	✔ (redeemScript)	✔	❌	❌
HASH160(pubkey)	❌	✔	❌	✔	❌	❌
Script intermedio	❌	❌	✔ (redeemScript)	❌	✔ (witnessScript)	Opcional
Hash del script	❌	❌	✔ HASH160(redeemScript)	❌	✔ SHA256(witnessScript)	✔ (tweak)
Versión SegWit	❌	❌	❌	v0	v0	v1
ScriptPubKey final	`<pubkey> OP_CHECKSIG`	`OP_DUP OP_HASH160 <h160> OP_EQV OP_CHECKSIG`	`OP_HASH160 <hRS> OP_EQUAL`	`0 <h160>`	`0 <sha256>`	`OP_1 <xonly-pubkey>`
Codificación dirección	❌	Base58Check	Base58Check	Bech32	Bech32	Bech32m
Prefijo típico	—	`1...`	`3...`	`bc1q...`	`bc1q...`	`bc1p...`
Eficiencia / fees	Muy baja	Baja	Baja	Alta	Media	Muy alta
Privacidad	Muy baja	Media	Baja	Media	Baja	Alta



Consultas eficientes con Fulcrum

01

Opción básica: JSON + nc

Hacemos un json a mano y lo envíamos por netcat a nuestro servidor Fulcrum.

02

Opción python

También se creará un json, pero tenemos ventajas de automatización y reutilización de la conexión.

03

Recepción de respuesta

El servidor responde casi instantáneamente con el saldo confirmado y no confirmado de la dirección.

El tiempo de ejecución de consultas con Fulcrum es inferior a 1 segundo, comparado con minutos usando Bitcoin Core sin indexación adicional. (yo tengo txindex=1)

Fulcrum: ejemplo de consulta

Convertimos la clave pública que se llevó el bloque 30 y no la dirección 17×23dNjXJLzGMev6R63uyRhMWP1VHawKc

echo -n 41042cf59fafd089a348c5834283573608e89a305c60a034604c7d22dde50998f1b9bb74681986ca1884a6b1df8ce7f1b79a2277057de855a634626e7a5851c1e716ac | xxd -r -p | sha256sum | awk '{print $1}' | sed 's/../& /g' | awk '{for(i=NF;i>0;i--) printf $i; print ""}'

gapk.json

{ "method": "blockchain.scripthash.get_balance",

"params": ["8d013257dc730fc1b96e26c3da52bff65ac7171d87f294f080eedcadc2bed687"],

"id": 1 }

usuario@debian:/json$ cat gapk.json | nc localhost 50001

{"id":1,"jsonrpc":"2.0","result":{"confirmed":5000000000,"unconfirmed":0}}

- Aquí seguimos comprobando que no es lo mismo la dirección P2PK que la P2PKH
- Y lo más importante, rápido como un pepino.

Fulcrum: conversión de dirección a scripthash

```python
def address_to_scriptpubkey(address):
    """ Convierto la dirección a scriptpubKey """
    # P2PKH
    if address.startswith("1"):
        decoded = base58.b58decode_check(address)
        pubkey_hash = decoded[1:]
        return b"\x76\xa9\x14" + pubkey_hash + b"\x88\xac"

    # P2SH
    if address.startswith("3"):
        decoded = base58.b58decode_check(address)
        script_hash = decoded[1:]
        return b"\xa9\x14" + script_hash + b"\x87"

    # Bech32
    if address.startswith("bc1q"):
        hrp, data = bech32_decode(address)
        if hrp != "bc":
            raise ValueError(f'HRP inválido {hrp} - {address}')
        witver = data[0]
        witprog = bytes(convertbits(data[1:], 5, 8, False))
        if witver == 0:
            return b"\x00" + bytes([len(witprog)]) + witprog
    # P2TR
    if address.startswith("bc1p"):
        witver, witprog = bech32m.decode("bc",address)
        if witver == 1:
            # P2TR (Taproot)
            if len(witprog) != 32:
                raise ValueError("Taproot (bc1p) debe tener exactamente 32 bytes en witness program")
            # scriptPubKey = OP_1 PUSH32 <32-byte-tweaked-key>
            return b"\x51\x20" + witprog
        else:
            raise ValueError(f"Versión witness no soportada para bc1p: {witver}")
    raise ValueError("Tipo de dirección no soportado")


def address_to_scripthash(address):
    """ Convierto de dirección a Hash invertido para consulta a Electrum """
    script_pubkey = address_to_scriptpubkey(address) # Primero obtengo el scriptpubKey
    sha = hashlib.sha256(script_pubkey).digest()
    hash = sha[::-1].hex()  # little-endian Electrum
    return hash
```

Fulcrum: ventajas e inconvenientes

Ventajas

Velocidad de consulta

Respuestas prácticamente instantáneas para cualquier dirección. Capacidad de procesar múltiples transacciones por segundo sin degradación del rendimiento.

Indexación completa

Acceso directo a todo el historial de transacciones por dirección(hash), sin necesidad de escanear toda la blockchain.

Inconvenientes

Requisito txindex=1 y nodo completo

Necesita activar el índice completo de transacciones en Bitcoin Core, aumentando significativamente el uso de disco, y días en indexar

Tiempo de indexación

Sin un SSD, el proceso inicial de indexación supera el mes de duración. Incluso con SSD puede tardar varios días (3)

P2PK y P2PKH

Los trata de forma diferente, no son uno

Fulcrum: tipos de discos duros

Discos mecánicos



Tiempo de indexado > 1 mes

Velocidad máxima en lectura 180 MB/s

Velocidad máxima escritura 160 MB/s

Discos Sata - SSD





Tiempo de indexado: semanas

Velocidad máxima en lectura 520 MB/s

Velocidad máxima en escritura 480 MB/s

Discos NVME - SSD



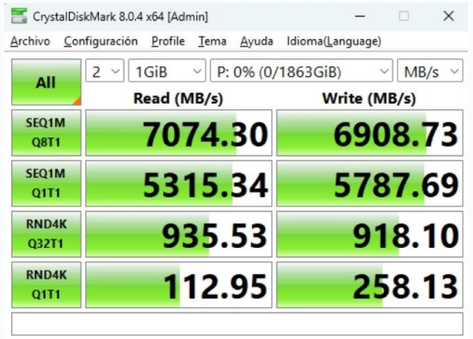
Precios de locura

Mi disco es un WD Black SN7100 PCIex 4.0. Pagué 180€ el 31/12/2025.

Precio actual 240€. Precio el 1/7/2025 130€

Tiempo de indexado 3 días

Estuve 10 días indexando con el disco mecánico y llegué al 60% de los bloques. Con el cambio de disco acabó en menos de 2 días.



Acto 4:

El poder real

Análisis histórico, disco duro en la basura, consultas en un plisplás, conclusiones.

Casos de uso: Bitcoin Core

Extracción de transacciones coinbase

Me los saco todos, hay 1 por bloque y recorro toda la blockchain. Me los guardo en un fichero csv. Antes pensaba que había algún problema con los bloques minados, pero era sólo para los que se convierte de clave pública a dirección.

1

Recorrer bloques

Hay exactamente una transacción coinbase por cada bloque minado desde el génesis. Y es la primera de cada bloque (tx=0)

2

Almacenar datos

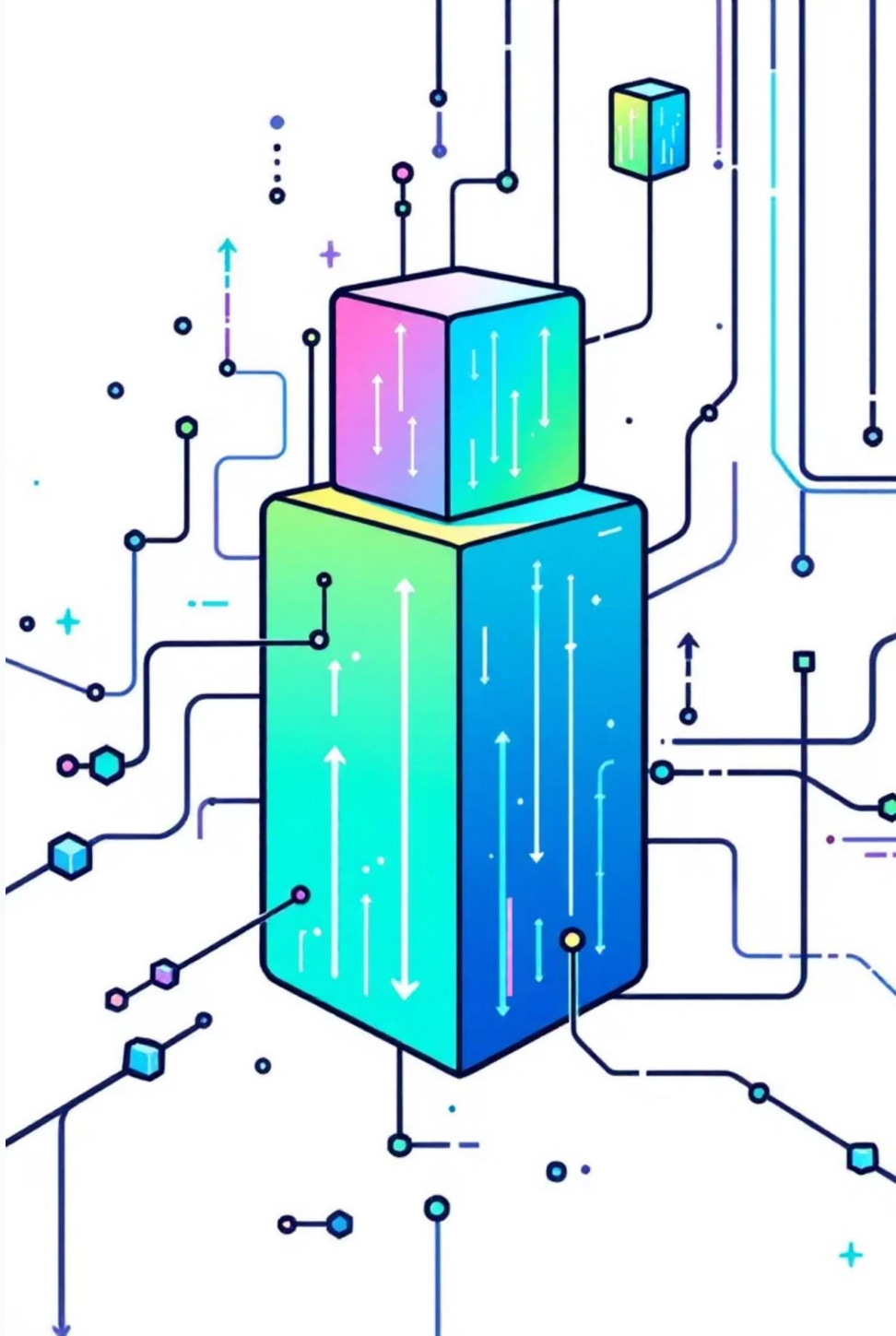
Guarda la información relevante de cada coinbase en una base de datos local. (Fichero CSV)

3

Análisis posterior

Salen datos muy curiosos después de procesar toda la blockchain.

Esto es un "análisis histórico" de minería y se puede ver la distribución de recompensas desde los primeros años de Bitcoin hasta la actualidad.



Resultados obtenidos: 000000-930340.csv

Una vez procesada toda la blockchain (varias horas, menos de 12) (me falta optimizar) los resultados se guardan en formato CSV para facilitar el análisis posterior con herramientas de hoja de cálculo, scripts personalizados u otra aplicación que vaya a utilizar estos datos. Desde el bloque 0 génesis al 930340.

1

Estructura del CSV

Por cada recompensa minada habrá al menos una fila, si se reparte entre varios habrá más, estarán las columnas para altura de bloque, número de orden, txid hash de transacción, cantidad, dirección, tipo de dirección, timestamp (fecha y hora del minado) y si está gastado o no.

2

Ventajas del formato

Fácil de filtrar, ordenar y visualizar. Compatible con las siguientes aplicaciones para aprovechar estos datos y no tener que calcularlos de nuevo

3

Aplicaciones prácticas

Análisis de patrones de minado de bloque, porcentaje de bloques minados sin gastar, fechas de minado, número de de destinatarios de las recompensas. Usaré Fulcrum para consultar el saldo de una dirección y sumaré si esa dirección tenía un bloque minado con dirección convertida.

Trozo del CSV

Bloque, Nº receptor, txid, recompensa, dirección, tipo de dirección, timestamp (fecha y hora del minado), ¿Gastado?

```
901028,1,b62b9ce445f38ac866edb86f8e2418ca59326b092bdaa080b64216479e3b9fb1,3.15837074,bc1qwzrryqr3ja8w7hnja2spmkgfdcgvqwp5swz4af4ngsjecfz0w0pqud7k38,P2WSH,2025-06-13T05:46:40+00:00,True
901029,0,3fc3afb85df431137ccc3615f2e8ecfb632d586cdf1f622da20b412f7db1428b,0.00000546,1K6KoYC69NnafWJ7YgtrpwJxBLiijWqwa6,P2PKH,2025-06-13T05:49:18+00:00,False
901029,1,3fc3afb85df431137ccc3615f2e8ecfb632d586cdf1f622da20b412f7db1428b,3.14041393,1AfCc4F9c4VTYSE31PUe2kUEKs6ZxiDjxm,P2PKH,2025-06-13T05:49:18+00:00,True
901030,0,48f369d0a88ef1b0c3a0a51aab3bef70c2997fd3776d85bc451a94428ac53c74,0.00000546,bc1pp7w6kxnj7lzgm29pmuhezwl0vjdlcrthqukll5gn9xuqfq5n673smy4m63,P2WSH,2025-06-13T05:51:15+00:00,False
901030,1,48f369d0a88ef1b0c3a0a51aab3bef70c2997fd3776d85bc451a94428ac53c74,3.13864498,bc1qwzrryqr3ja8w7hnja2spmkgfdcgvqwp5swz4af4ngsjecfz0w0pqud7k38,P2WSH,2025-06-13T05:51:15+00:00,True
901031,0,3093f23010337ff7635cbf24e09c14f66536cb24b1e16a9fa131750096f65b6d,3.13176656,bc1qrpp7g75sx3ejclvsfdw2uahzchtyu7vumkuadu,P2WPKH,2025-06-13T05:54:06+00:00,True
901032,0,d80ec72849da7eabbd7fcadb72c615622f3d46bdafc4921fe2a2db9bd4d33270,3.14936766,1BM1sAcrfV6d4zPKytzziu4McLQDsFC2Qc,P2PKH,2025-06-13T06:01:44+00:00,True
901033,0,e413cd097580b5804d339232683a4154d8b7be170d6e1929ef22866254a0d793,0.00000546,bc1pp7w6kxnj7lzgm29pmuhezwl0vjdlcrthqukll5gn9xuqfq5n673smy4m63,P2WSH,2025-06-13T06:05:13+00:00,False
901033,1,e413cd097580b5804d339232683a4154d8b7be170d6e1929ef22866254a0d793,3.13419064,bc1qwzrryqr3ja8w7hnja2spmkgfdcgvqwp5swz4af4ngsjecfz0w0pqud7k38,P2WSH,2025-06-13T06:05:13+00:00,True
901034,2,7b68bff92a539c392eb96ddb688c29685b637e3ca32490ae1edfa7a9a4292ee6,0.48560526,bc1qpthfpk8aj45363s4gm4j4d5g67c7gsfaljuyshyj8f5cr0kww6wsqaf7qg,P2WSH,2025-06-13T06:05:27+00:00,True
901034,3,7b68bff92a539c392eb96ddb688c29685b637e3ca32490ae1edfa7a9a4292ee6,0.41096932,bc1q6f3ged3f74sga3z2cgeyehv5f9lu9r6p5arqvf44yzsy4gtjxtlsmnhn8j,P2WSH,2025-06-13T06:05:27+00:00,True
901034,4,7b68bff92a539c392eb96ddb688c29685b637e3ca32490ae1edfa7a9a4292ee6,0.35550899,bc1qdxpl0l5c86tdck6rvalmzjpz5p64jcmyeg2ggv,P2WPKH,2025-06-13T06:05:27+00:00,True
901034,5,7b68bff92a539c392eb96ddb688c29685b637e3ca32490ae1edfa7a9a4292ee6,0.26791946,bc1q6xf8r8zmap5qdrcz8wfc8v7tajm2lwckp28ldl,P2WPKH,2025-06-13T06:05:27+00:00,True
901034,6,7b68bff92a539c392eb96ddb688c29685b637e3ca32490ae1edfa7a9a4292ee6,0.15247865,35eS5Lsqw8NCjFJ8zhp9JaEmyvLDwg6XtS,P2SH,2025-06-13T06:05:27+00:00,True
901034,7,7b68bff92a539c392eb96ddb688c29685b637e3ca32490ae1edfa7a9a4292ee6,0.07724597,bc1q4sg69xjhy85kqt3edtxhu5skeyvkkldmzn84wwfpc3hxl63c4syshh5lfa,P2WSH,2025-06-13T06:05:27+00:00,False
901034,8,7b68bff92a539c392eb96ddb688c29685b637e3ca32490ae1edfa7a9a4292ee6,0.06068587,bc1qkvnltjlxxemxe562xmd2g9aqh5ln7tpsyynpg0,P2WPKH,2025-06-13T06:05:27+00:00,True
901034,9,7b68bff92a539c392eb96ddb688c29685b637e3ca32490ae1edfa7a9a4292ee6,0.05819686,12i9DnPv6r9KoJHnVQsS8Hs1kSk6Bb1qF5,P2PKH,2025-06-13T06:05:27+00:00,True
901034,10,7b68bff92a539c392eb96ddb688c29685b637e3ca32490ae1edfa7a9a4292ee6,0.05748695,bc1qauy6m9mgjtwqjrwllwszgzc4xzp3p9mnd75p5y,P2WPKH,2025-06-13T06:05:27+00:00,True
901034,11,7b68bff92a539c392eb96ddb688c29685b637e3ca32490ae1edfa7a9a4292ee6,0.05567495,3Dkn56s9MVsN7tc6dm8rD4s5rqRyRDHBWC,P2SH,2025-06-13T06:05:27+00:00,True
901034,12,7b68bff92a539c392eb96ddb688c29685b637e3ca32490ae1edfa7a9a4292ee6,0.05522693,bc1qqjhanahrdwnu4kmzzv7ytzjnrr6p5v77lj3xqy,P2WPKH,2025-06-13T06:05:27+00:00,True
901034,13,7b68bff92a539c392eb96ddb688c29685b637e3ca32490ae1edfa7a9a4292ee6,0.05473116,3A4ND5Q2muSpwhn5gw8QpFcQpFdFgBmBnY,P2SH,2025-06-13T06:05:27+00:00,True
901034,14,7b68bff92a539c392eb96ddb688c29685b637e3ca32490ae1edfa7a9a4292ee6,0.05372278,bc1qm3qxs472cxua8afudxr2eduwwa6vvppgzpmdzrfler588cvd7mgqzc8n9t,P2WSH,2025-06-13T06:05:27+00:00,False
901034,15,7b68bff92a539c392eb96ddb688c29685b637e3ca32490ae1edfa7a9a4292ee6,0.04620410,31qrH4wLeyPefihYfru3nikdCF8WoSBAJq,P2SH,2025-06-13T06:05:27+00:00,True
```

Curiosidades obtenidas de mineros ★

% UTXOs gastados o sin gastar: El 8,12% de las recompensas minadas no se han movido. 1.763.002 BTC

Número máximo de destinatarios de un minado:

El bloque 133471 (27/6/2011) tiene 919 destinatarios, hay todavía 328 destinatarios que no han movido su recompensa. (minado por el pool Eligius)

En 2025 el número máximo de destinatarios fue 75 en el bloque 902024 (Minado por el pool Ocean)

Tipos de direcciones usadas:

P2PKH:2121509 P2PK:196479 P2SH:137798 P2WPKH:73381 P2WSH:23436 P2TR: 19647

Primer uso en minado:

P2PKH: 14-09-2010 P2SH: 20-01-2015 P2WPKH:23-03-2018 P2WSH: 03-12-2023 P2TR: 11-04-2024

Última vez que se usó un P2PK en minado: Bloque 886688 07-03-2025 (Pool p2p-spb.xyz)

Caso real para probar el nodo: búsqueda en la basura

> Not your keys, not your coins

La historia

Entre 7.500 y 8.000 Bitcoin minados durante 2009, almacenados en un disco duro que fue desechado en agosto de 2013. ¿A quién se le ocurre tirar un disco duro? Yo aún tengo un disco IDE de 80GB en una caja.

Detalles técnicos del último bloque de 2009

- Bloque: 32489
- Recompensa: 50.00000000 BTC
- Fecha de minado: 31 de diciembre de 2009, 23:44:29 UTC
- Estado: Sin gastar
- Transaction ID: ec1f8104b94a9d5ef46d8b65a3d232ba8a69b03ffb0e6effb2560dc84d451f48
- Dirección: 1Mv3d7GTXdsvEFMbM9zujDP2e5DcTcJTYr

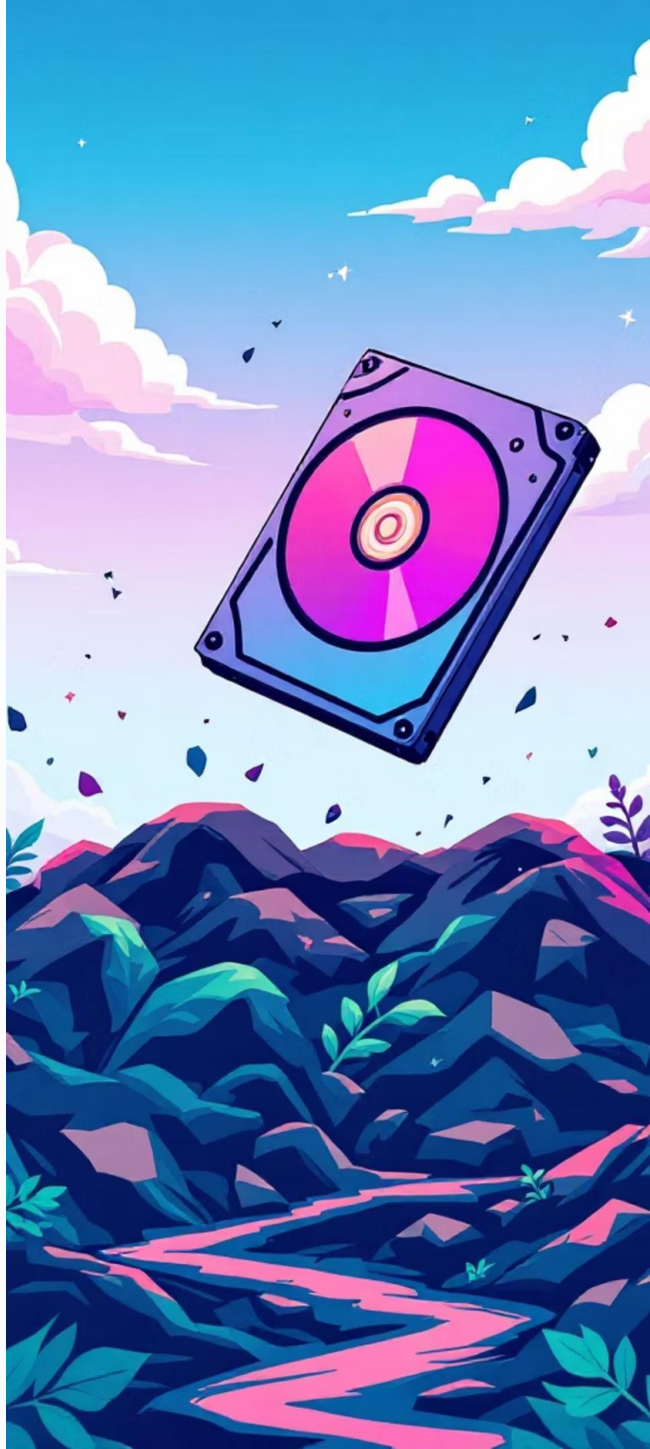
¿Qué vamos a buscar?

```
Direcciones usadas en 2009. No usadas posteriormente a 2013.
```

Posbile dirección Bitcoin

```
198aMn6ZYAczwrE5NvNTUMyJ5qkfy4g3Hi
```

Este caso ilustra la importancia de mantener copias de seguridad seguras de tus claves privadas y la inmutabilidad de la blockchain: esos Bitcoin permanecen visibles pero inaccesibles para siempre.



Caso real: búsqueda en la basura

> Modificaré mi aplicación previa. Esta ya recorre todos los bloques, y sólo usa la primera transacción (coinbase), que use las siguientes también.

¿Qué hará la aplicación?

- Recorrer los bloques del año 2009
- Crear una contabilidad por cada dirección que aparezca en las transacciones de cada bloque
  - Sumo los coinbase
  - Resto las tx salientes
  - Sumo las tx entrantes.
- Con la contabilidad hecha: consultar las direcciones >7000 BTC
- Con el filtro de tamaño, consultar qué direcciones no han enviado fondos desde agosto de 2013

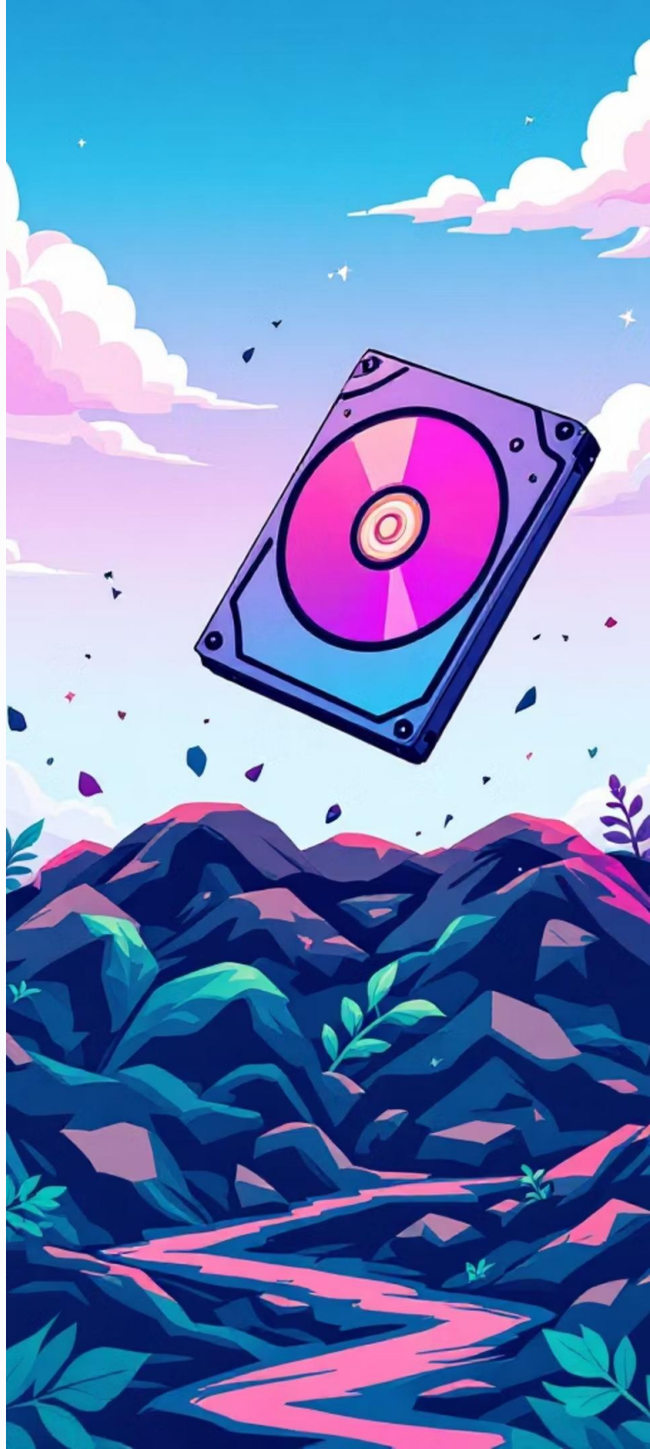
Tiempo esperado de ejecución

```
Poco: en 2009 los bloques solían tener
1 tx: la coinbase
```

Direcciones mineros

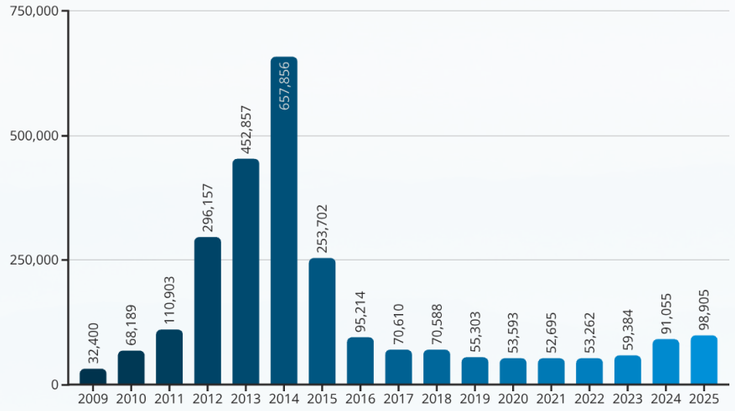
```
La búsqueda no es trivial, porque no
se repite ni una dirección de destino
de los bitcoin minados en 2009
```

Actualmente cada pool mina siempre a la misma dirección. Y ganan los mismos. La sorpresa es que alguien diferente se lo lleve

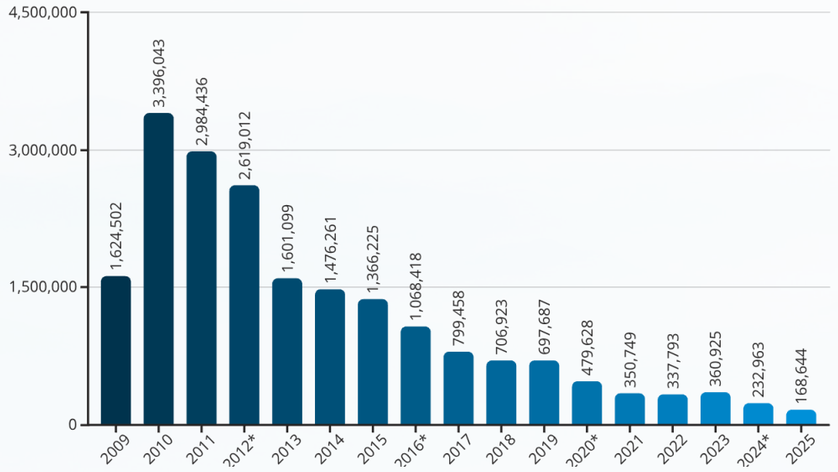


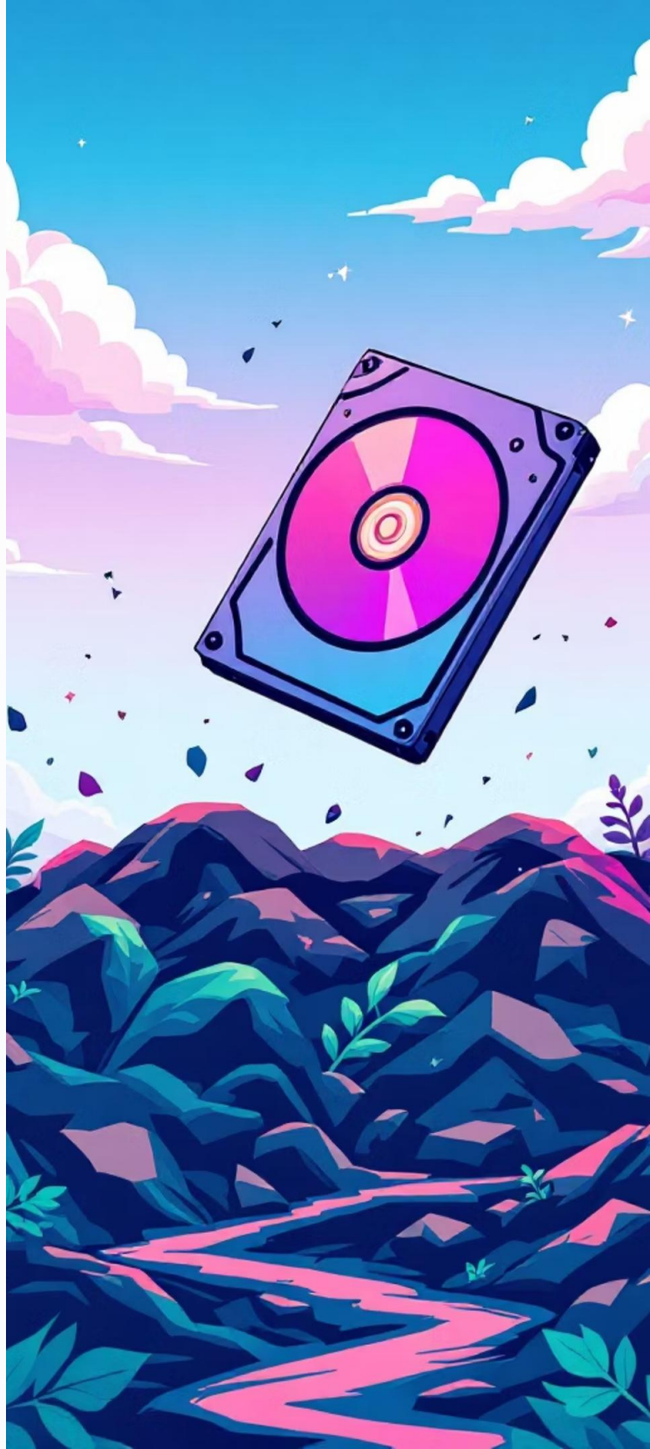
Caso real: búsqueda en la basura

Nº direcciones de destino



Bitcoin minados por año

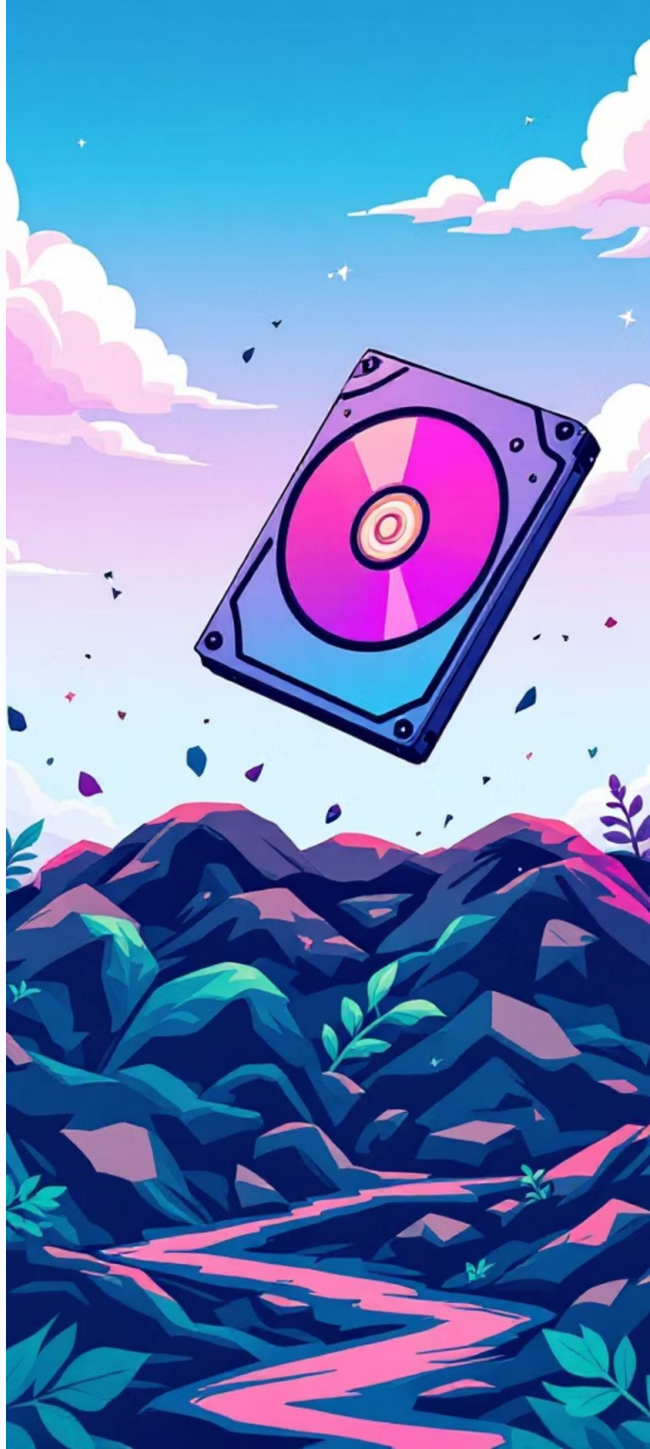




Caso real: búsqueda en la basura

> Resultado previo de la aplicación

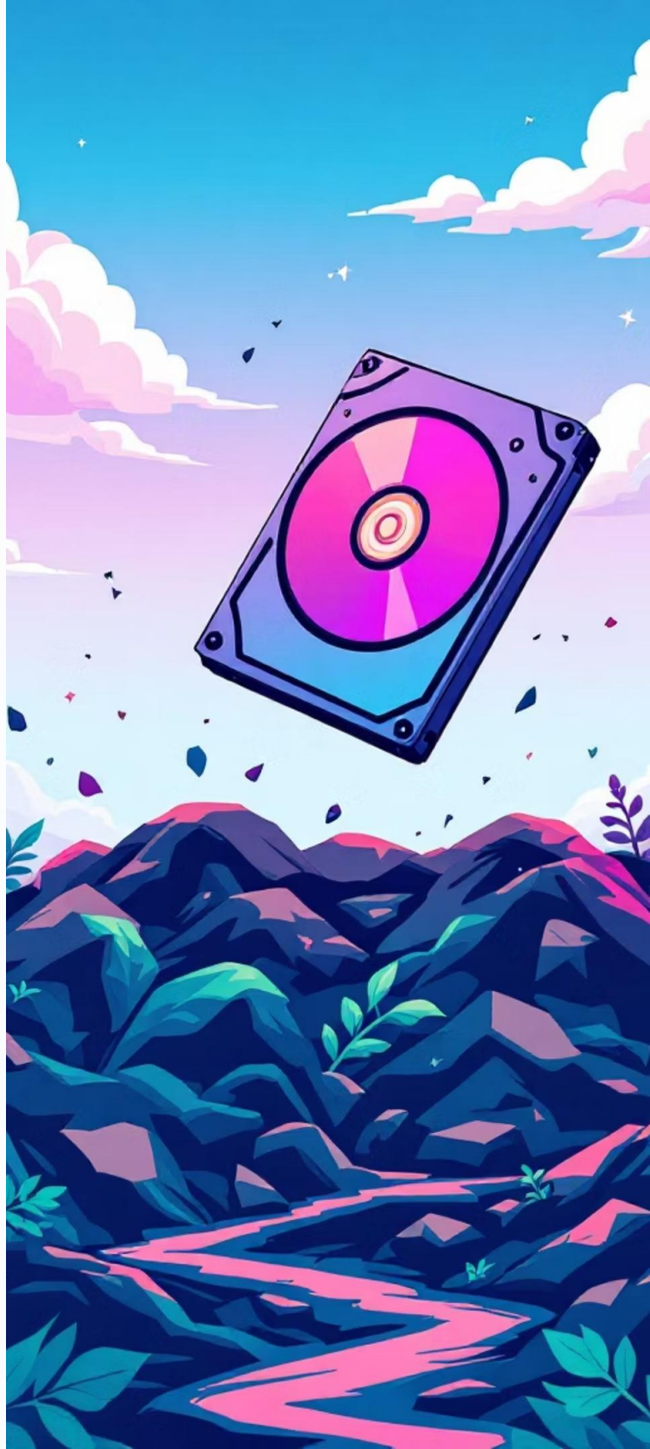
12higDjoCCNXSA95xZMWUdPvXNmkAduhWv	61424.62
198aMn6ZYAczwrE5NvNTUMyJ5qkfy4g3Hi	8000.0
1Ff2xcJtfWWiNRCTUuTng6TJQm3g8h61Y2	22500.0
1MxR5zEtu4cq3Sm7AtfpvpLyJwf4f4kJBi	10000.2
1Ph1WL8gHPW1VShh55yJpRQE57wth6zSBm	15954.35



Caso real búsqueda en la basura ★

Curiosidades

1. En 2009 aparecieron 32610 direcciones diferentes, sólo se crearon 210 direcciones
2. La ejecución ha durado de 80 segundos (aplicación sin optimizar) Casi no hay transacciones en 2009
3. Había 38 direcciones que tenían más de 61 BTC, están vacías o casi vacías todas menos tres. La que hemos visto antes y
   a. 137zjnSXZs7Wdhg8zCoAJHz3NPgX8WtPPv con 895 BTC (sin ninguna salida)
   b. 1JuEjh9znXwqsy5RrnKqgzqY4Ldg7rnj5n con 82 BTC (sin ninguna salida)
4. 21487 direcciones recibieron 50 BTC de minado y nunca se gastaron (1074350 BTC)



Consulta de saldo con Fulcrum

App que a partir de un fichero con direcciones te devuelve el "saldo" de esa dirección

```
usuario@debian:~$ python3 saldo_fulcrum_MasMinado_fichero.py prueba.2009
Saldo confirmado: 12higDjoCCNXSA95xZMWUdPvXNmkAduhWv 35711 + 0 = 35711 sats ; No confirmado 0 sats
Saldo confirmado: 1Ff2xcJtfWWiNRCTUuTng6TJQm3g8h61Y2 0 + 0 = 0 sats ; No confirmado 0 sats
Saldo confirmado: 1Ph1WL8gHPW1VShh55yJpRQE57wth6zSBm 0 + 0 = 0 sats ; No confirmado 0 sats
Saldo confirmado: 1MxR5zEtu4cq3Sm7AtfpvpLyJwf4f4kJBi 0 + 0 = 0 sats ; No confirmado 0 sats
Saldo confirmado: 198aMn6ZYAczwrE5NvNTUMyJ5qkfy4g3Hi 800000707240 + 0 = 800000707240 sats ; No confirmado 0 sats
Saldo confirmado: 15J5MZxBkbo88hfHbY96UEDdWWtTJKTBgj 890 + 0 = 890 sats ; No confirmado 0 sats
Saldo confirmado: 1N6cJEdLm7KSg1jRXTYKv7iYkRtzFLKNDH 0 + 0 = 0 sats ; No confirmado 0 sats
Saldo confirmado: 1NAauy83Upy9yYv5zjcjk4hwPRJWnzvX52 0 + 0 = 0 sats ; No confirmado 0 sats
Saldo confirmado: 1PZwWEJe5YGx9bzJbEpgXiWLuWnyD2rjLA 0 + 0 = 0 sats ; No confirmado 0 sats
Saldo confirmado: 137zjnSXZs7Wdhg8zCoAJHz3NPgX8WtPPv 89565172293 + 0 = 89565172293 sats ; No confirmado 0 sats
```

Prueba con aplicación python

Reaprovechando la conexión y automatizando con un fichero de 208837 direcciones, lo realiza en 147 segundos. Obteniendo 1420 consultas por segundo. A toda leche. (antes eran 40 segundos para 1 consulta). Ejecutando a la vez 2 programas, consigo comprobar 417674 direcciones en 155 segundos: 2694 consultas por segundo. Ha merecido la pena.

¿FIN?

- Bitcoin no es fácil, pero es verificable
- La soberanía tiene coste técnico (y económico)
- Los datos reales cambian tu percepción de la historia

> “La blockchain no guarda secretos, pero solo habla con quien sabe escucharla.”