
# Expertos del Protocolo

Clase 2

Librería de Satoshi

**₿4OS**

</bitcoin for open source>

CLASS OVERVIEW

1. Mas números:
- Big Endian / Little Endian
- Compact Size

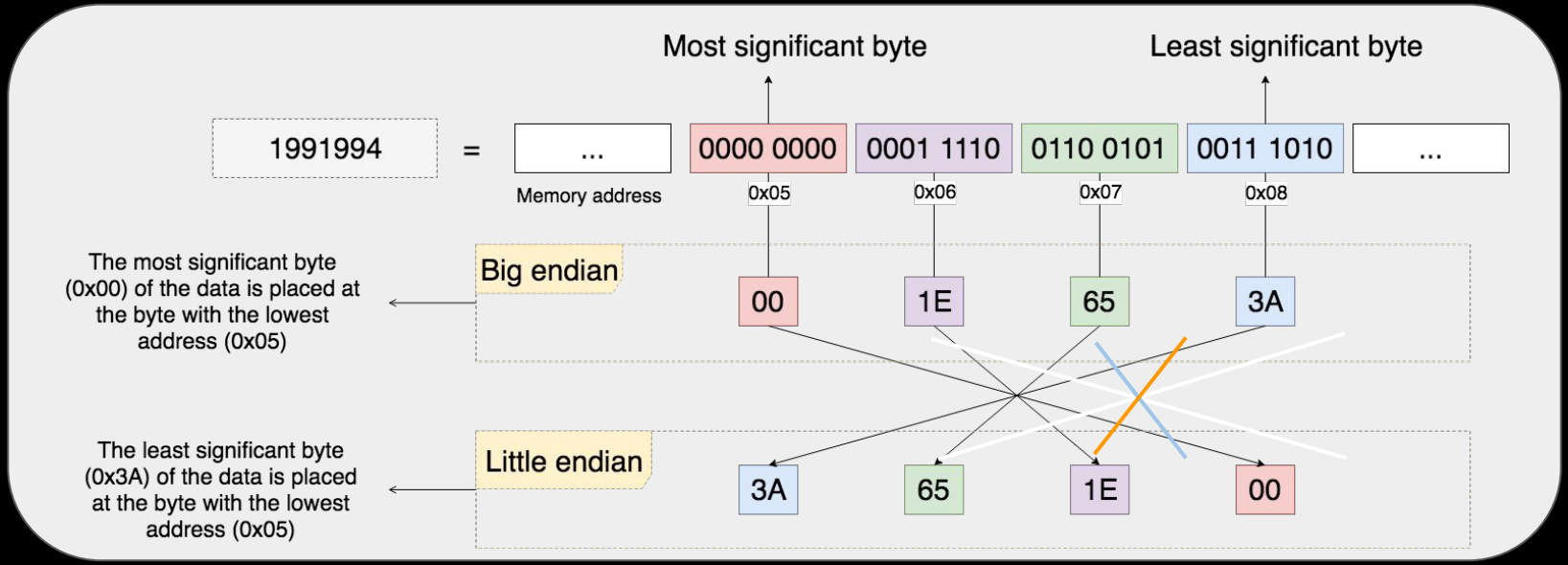
2. Transacciones:
- Campo por campo

# Big Endian - Little Endian

B4OS

# Big Endian - Little endian

Usualmente usamos el Big Endian



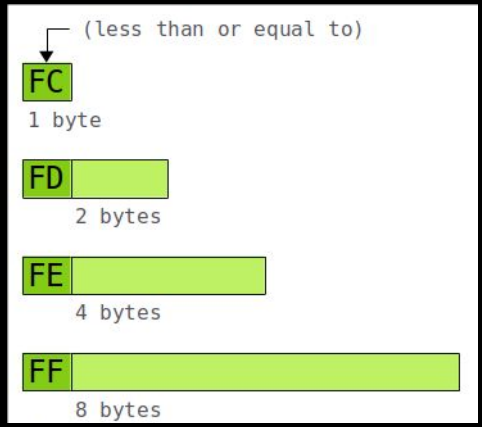

B4OS

# Compact Size

B4OS

# Compact Size

La cantidad de bytes que ocupa la variable depende del valor del primer byte



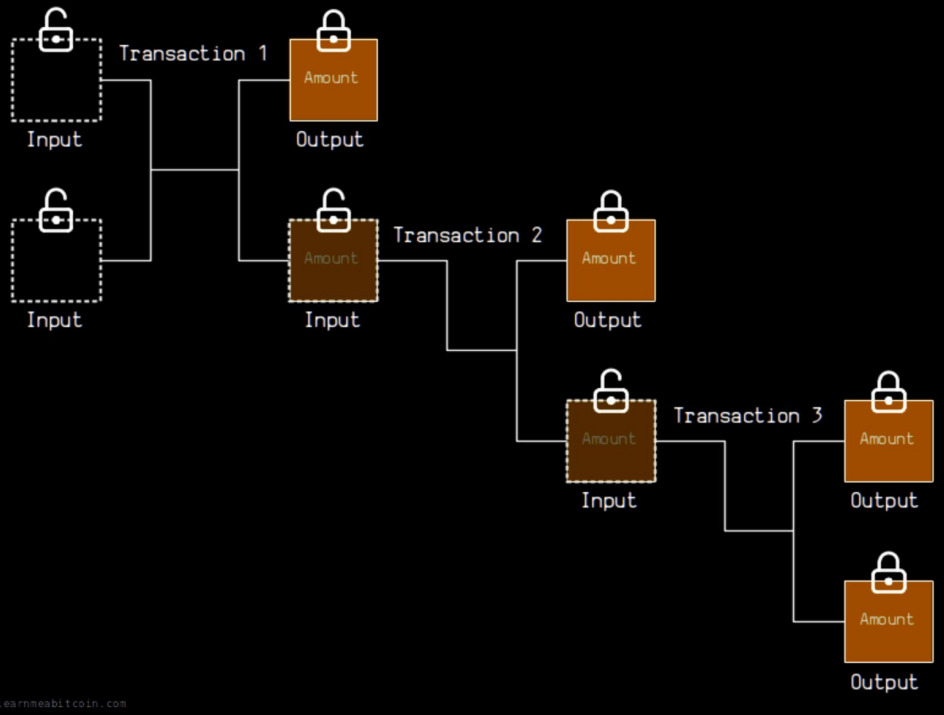

| Leading Byte | Number | Range | Field Size | Example |
|---|---|---|---|---|
| FC (and below) | Current byte | 0 - 252 | 1 byte | 64 (100) |
| FD | Next 2 bytes | 253 - 65535 | 3 bytes | FDE803 (1,000) |
| FE | Next 4 bytes | 65536 - 4294967295 | 5 bytes | FEA0860100 (100,000) |
| FF | Next 8 bytes | 4294967296 - 18446744073709551615 | 9 bytes | FF00E40B5402000000 (10,000,000,000) |

**Note:** The bytes containing the number are in little-endian.

# Transacciones

B4OS

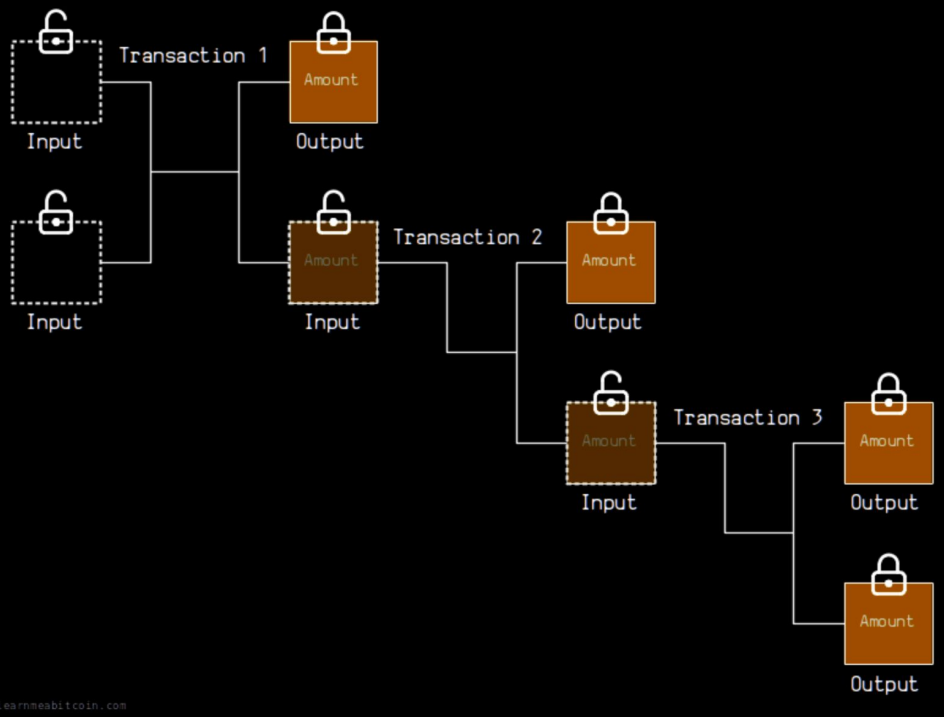
# Transacciones de Bitcoin




¿Qué se mueve durante una transacción?

B4OS

# Transacciones de Bitcoin




## ¿Qué mueve una transacción: Inputs y Outputs?

Una transacción **no transfiere bitcoins entre direcciones**.

En cambio:

- **Consume salidas anteriores (UTXOs)** como inputs.
- **Crea nuevas salidas (outputs)** con un valor y una condición de gasto.

*"Es más bien como romper una moneda para crear nuevas con diferente denominación."*

B4OS

Como se ve una transacción en la blockchain:

```
  "hex":

"01000000000101913f1adc3278713e2dcf37ec68d28dc39b4128415cb1d4f101113ca55516b6
800100000000fdffffff022347000000000000001600142bc513dd08e0cb0d14f3d9a2406afd5fa
a76eeb5e739000000000000160014358fccaa9abbeee4bbde4ecb0f29fbcc87c6d15a02473044
022066af37296e2d6041922eed156f16fdc6c25aad16c06f733e0f63991a7dbb59b402204fe0e
4565e82c267f3ec2cb575361b8788bf0a3c8f10db81a8b2c19bd91f82b70121023e715bbdeae3
b90a61f34dd08f5c21e8f8ae801890cedaf14dcbd2b241ec2ea300000000",
```

Otro ejemplo:

```
  "hex":

"02000000fdf40197a690863ffe8c24d486cca5970a5f0940e84e82ad4d86ce133d1c018f3358
2a000000006a47304402206b4548aa41c93e49654bec102ec1050804f49f248356cb7918d9268
2ed04e91b02206a9eb32c9a05f29dc9bf4bb1ae5ff2c99613c103e12732fcac198b79e4220d8b
012102c1face5e7ed97aa53d2eaf922c89c87c6c24c890f597d39d8507c03d563a3c95fffffff
ff6925a3ffa1b73a176b1aa731683ce6e7887ad62ca458bb00b05c2ff2079e5f7060000006a47
304402206647c329051e0f8eac1aa273acab453c8c297a58604f4e788d69e3a457e77cb202206
e3f3e85fb816c8ab8193b6684096692afce6fe0f8a576611cb398b3a80e5b49012102500eebfe
6d1b238084df0981b4fcb7880e854512fd86bbce81b733cf40cc6778ffffffff8562ddad404d2
95ebcaf2683cdeeca31c53f6aa6323854b42c06b2f57ac11b030b0000006b483045022100b2f4
c9764b053a894f73b227f3a00d17d665ef57b1dc7da2ab91b29bb7c88ad302201f49de03554f9
e8023a82ec0c125373873c9a2c4e0f57845562b4db1d8cdce060121029bd5b2ce8fddbe17296a
fc69882a53376a79f501cab5cf4efa2fc8ec4aa88d07ffffffffeaedf2b2095453f58d7ed355e
fb650dc06d210bdb26dff425d5cde5b3e3a21af030000006a47304402202d1a6bb3264b60ba1f
bce337b19ad9468adc1bf5cb0e3bee7dc3e0224d26405f0220293cd46771233a73fbf14fdc9b4
ba47be17dddf8da511beb56f1cc02898a411a012102500eebfe6d1b238084df0981b4fcb7880e
854512fd86bbce81b733cf40cc6778ffffffff4271a3ff489188eb447581489e556b9dca8d",
```

Bitcoin **no almacena transacciones en formato JSON o texto**, sino en un **formato binario compacto** para ahorrar espacio y hacer que las transacciones sean eficientes en la red.

Un ejemplo es:

- Una transacción en **JSON** ocupa **500 bytes**.
- La misma transacción en **binario** ocupa **250 bytes** (aproximadamente la mitad).

Bitcoin **no almacena transacciones en formato JSON o texto**, sino en un **formato binario compacto** para ahorrar espacio y hacer que las transacciones sean eficientes en la red.

Un ejemplo es:

- Una transacción en **JSON** ocupa **500 bytes**.
- La misma transacción en **binario** ocupa **250 bytes** (aproximadamente la mitad).

***| ¿Por qué es importante reducir el tamaño de una transacción en Bitcoin?***

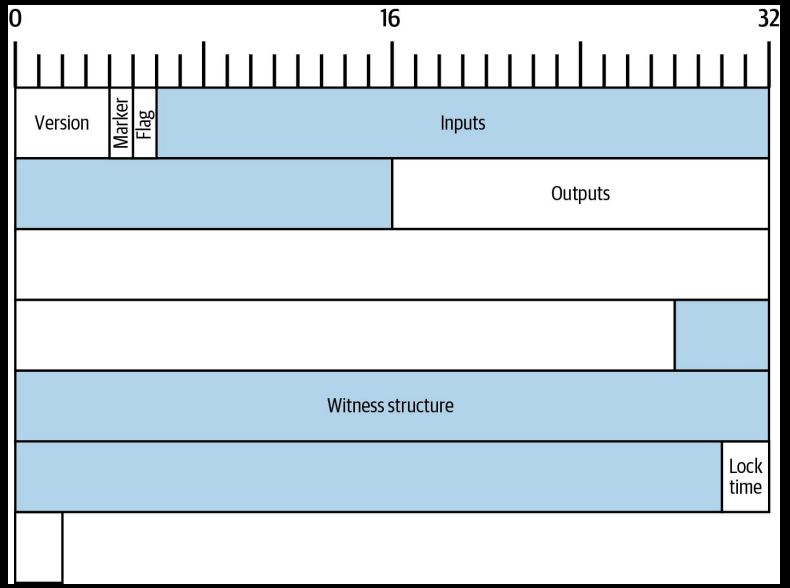
## | *¿Por qué es importante reducir el tamaño de una transacción en Bitcoin?*

- Menores comisiones
- Mayor capacidad en bloques
- Mayor velocidad de propagación

# Serialización


```
"hex":

"010000000000101913f1adc3278713e2dcf37ec68d28dc39b4128415cb1d4f101113ca55516b6800100000000fd
ffffff022347000000000000001600142bc513dd08e0cb0d14f3d9a2406afd5faa76eeb5e73900000000000016001
4358fccaa9abbeee4bbde4ecb0f29fbcc87c6d15a02473044022066af37296e2d6041922eed156f16fdc6c25aad
16c06f733e0f63991a7dbb59b402204fe0e4565e82c267f3ec2cb575361b8788bf0a3c8f10db81a8b2c19bd91f8
2b70121023e715bbdeae3b90a61f34dd08f5c21e8f8ae801890cedaf14dcbd2b241ec2ea300000000",
```

# Serialización

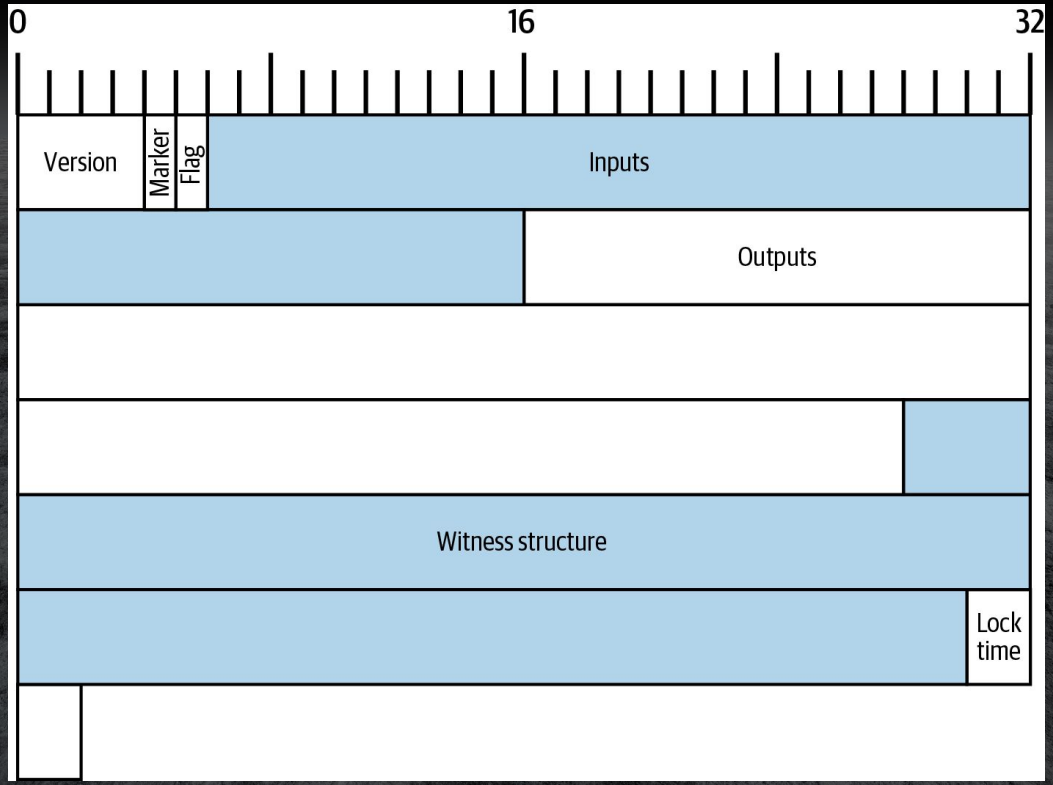

```
  "hex":

"01000000000101913f1adc3278713e2dcf37ec68d28dc39b4128415cb1d4f101113ca55516b6800100000000fd
ffffff022347000000000000001600142bc513dd08e0cb0d14f3d9a2406afd5faa76eeb5e73900000000000016001
4358fccaa9abbeee4bbde4ecb0f29fbcc87c6d15a02473044022066af37296e2d6041922eed156f16fdc6c25aad
16c06f733e0f63991a7dbb59b402204fe0e4565e82c267f3ec2cb575361b8788bf0a3c8f10db81a8b2c19bd91f8
2b70121023e715bbdeae3b90a61f34dd08f5c21e8f8ae801890cedaf14dcbd2b241ec2ea300000000",
```


[version][marker][flag][input_count][inputs][output_count][outputs][witness][locktime]

[version][input_count][inputs][output_count][outputs][locktime]

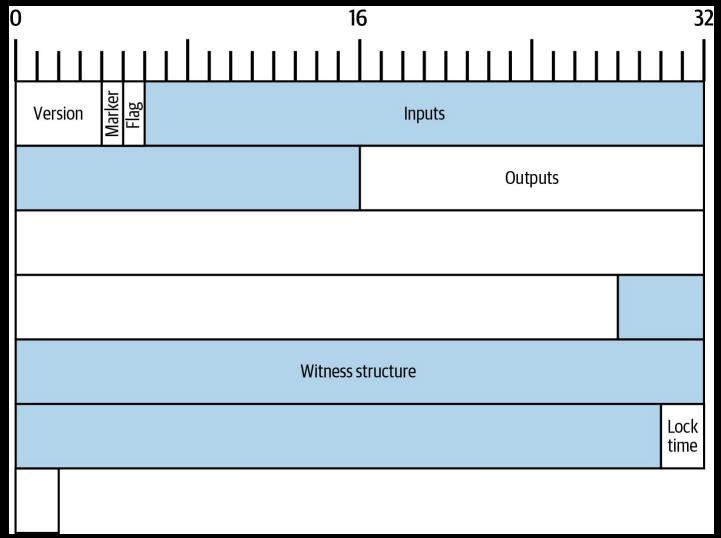
# Repaso de los campos



# Version (4 bytes), Marker y Flag (1 byte c/u)

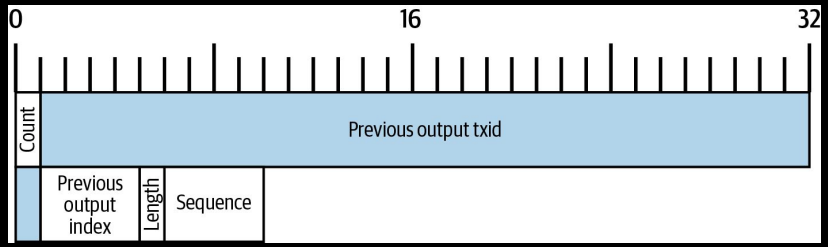
-Versión:Especifica reglas que se seguirán.  Hoy en día se usan 1 y 2 (campo little endian)

-Marker y Flag: Aquí está la "magia" de SegWit. Indica transacción SegWit para quien lo entienda, pero es compatible con los nodos que no conozcan SegWit



# Inputs:

Count: Compact Size integer
enumerando cantidad de entradas




# Cada input:

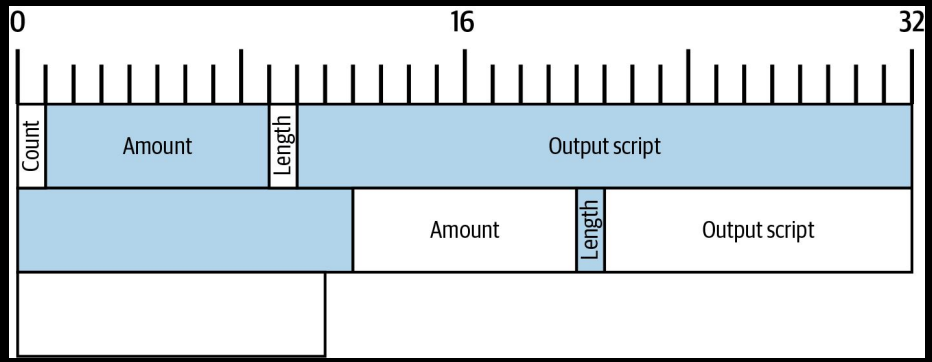
-TXID (32 bytes)
-VOut (4 bytes)
(Con esto: disponibilidad,monto, condiciones de gasto, timestamp)

-ScriptSig Length (Compact Size)
( 0 en SegWit, con campo ScriptSig Length indicando tamaño)
-ScriptSig

-Sequence (4 bytes)
Indica bloqueos de tiempo absolutos o relativos, RBF (viejo)

# Outputs:

Count: Compact Size integer
enumerando cantidad de salidas

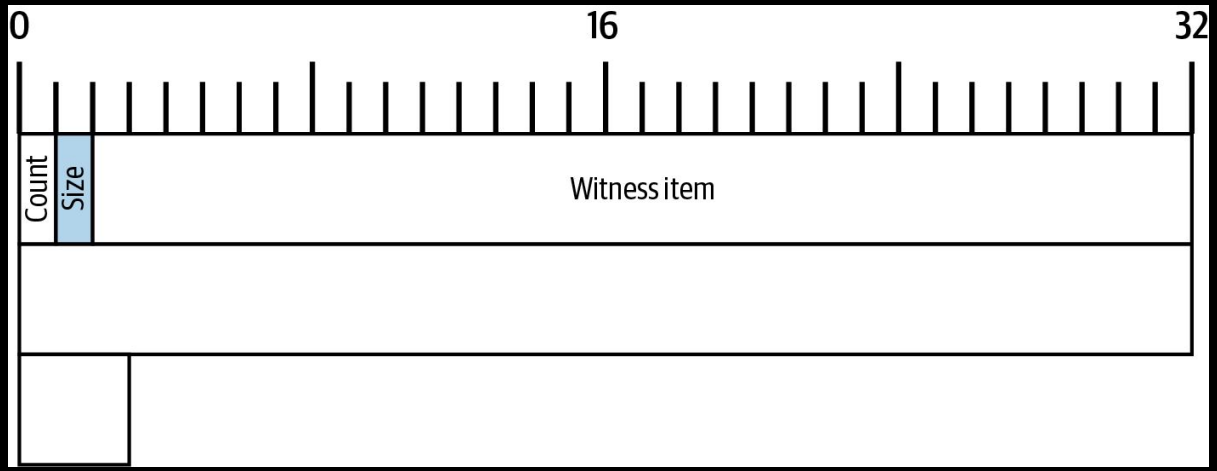



# Cada output:

-Amount (8 bytes, en Satoshis)
-Length (Compact Size)
-Output Script
(Describe el "candado" que bloqueará estos Satoshis)

# Witness Structure

-Count: Compact Size integer enumerando cantidad de witnesses
-Contenido del witness field




B4OS

# SegWit

Sin SegWit, los datos de firma pueden ocupar hasta el 65% de un bloque. (legacy)

Con SegWit, los datos de firma se retiran del input de las transacciones.

El tamaño real de los bloques sigue siendo 1MB, pero su límite efectivo de tamaño es de 4MB

Evitamos la "maleabilidad"

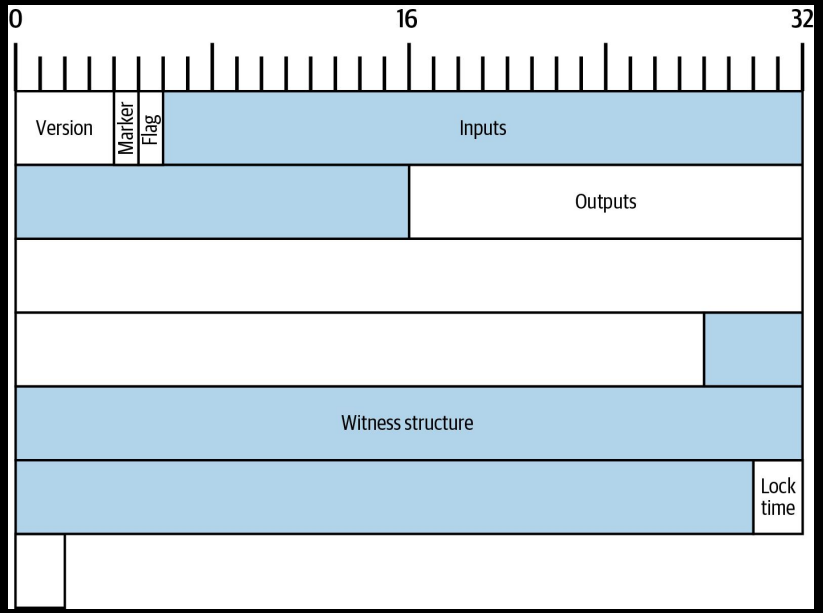
# SegWit

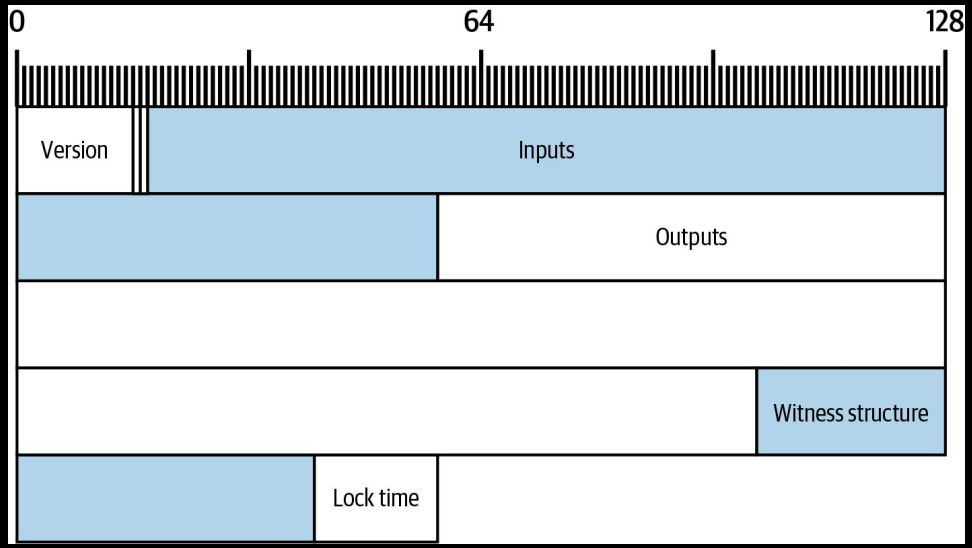
**SegWit** introdujo el concepto de **weight units (unidades de peso)**:

- Cada byte del bloque **base** vale 4 unidades de peso.
- Cada byte del **witness data** vale solo 1 unidad de peso.
- El límite total del peso de un bloque es de **4,000,000 unidades**.

Por eso decimos que **el "tamaño efectivo" de un bloque puede ser hasta 4 MB**, si usamos muchas transacciones SegWit (porque el witness pesa menos).

# SegWit


0
16
32
Version
Marker
Flag
Inputs
Outputs
Witness structure
Lock time


0
64
128
Version
Inputs
Outputs
Witness structure
Lock time

# nLockTime (4 Bytes)

## ¿Cómo funciona?

El campo nLockTime puede contener:

- Un número de bloque → si el valor es menor a 500 millones

**Si nLockTime < 500,000,000**
Se interpreta como un número de bloque.

Ej: **nLockTime** = 840000

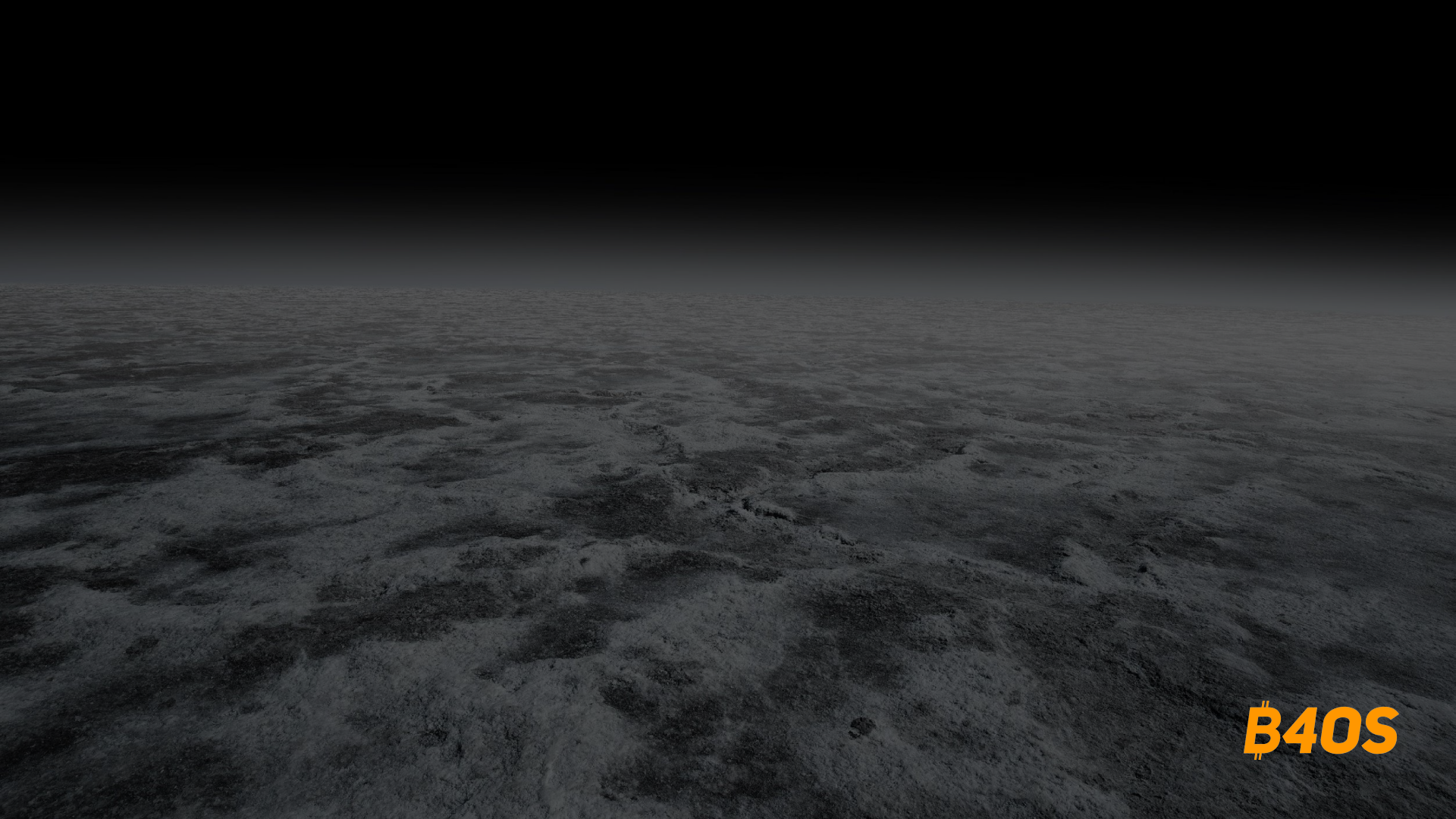
→ La transacción solo podrá ser incluida a partir del bloque #840000.

# Preguntas y Dudas

B4OS


libreriadesatoshi.com & b4os.dev

B4OS


B4OS