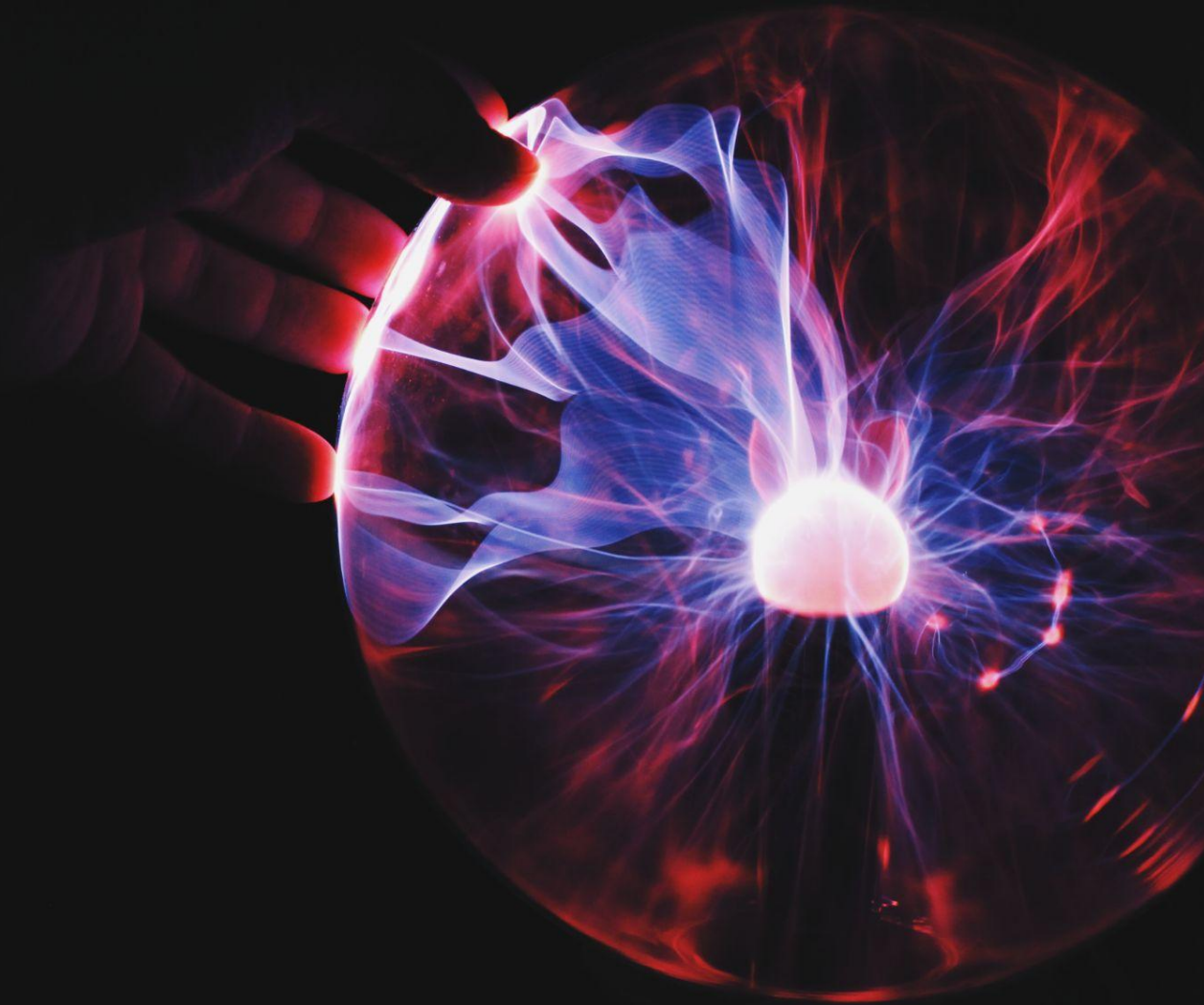


INPUT | OUTPUT

Introducción a Blockchain



+ CRIPTOGRAFÍA _

Introducción a conceptos claves de criptografía

2

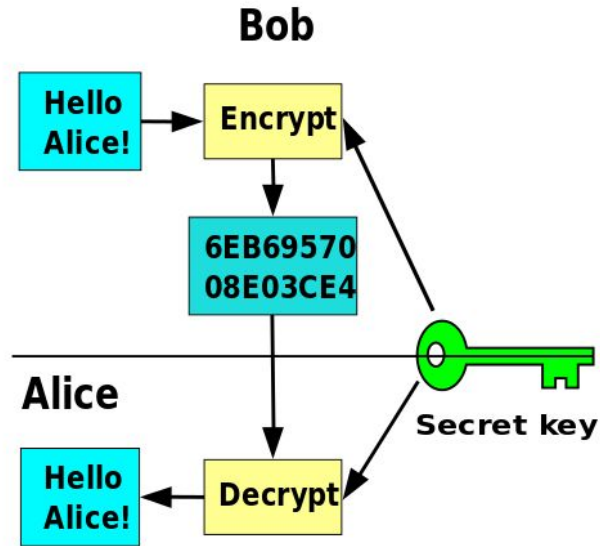
Criptografía - Definición

- Práctica y estudio de técnicas y protocolos para lograr la **comunicación segura en la presencia de comportamiento adversario**.
- La criptografía moderna se encuentra en la intersección entre:
 - Ciencias de la computación
 - Seguridad de la información
 - Matemática
 - Ingeniería electrónica
 - Procesamiento digital de señales
 - ...
- La seguridad de todo el internet (contraseñas, conexiones seguras con pag webs, transferencias de banco, etc.) depende de la criptografía moderna.

Criptografía - Objetivos/propiedades de sistemas criptográficos

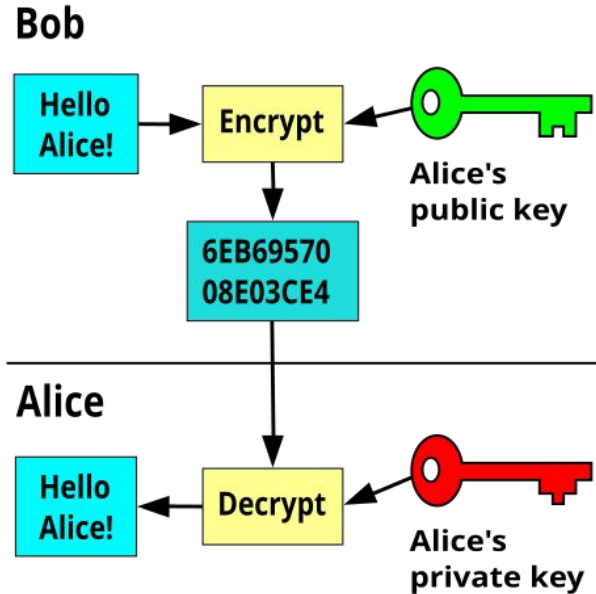
- **Confidencialidad:** Garantiza que la información sea accesible únicamente a personal autorizado.
- **Integridad de los datos:** Garantiza que la información es correcta y completa.
- **Vinculación:** Permite vincular un documento, mensaje, o transacción a una persona o identificador único (antiguamente llamado "No repudio").
- **Autenticación:** Permite verificar la identidad del comunicador.

Criptografía - Criptografía de clave simétrica



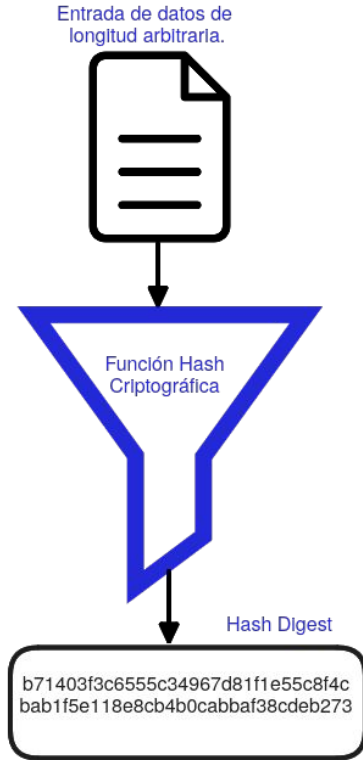
- $E = f(\text{msj}, \text{clave}) \rightarrow \text{cifrado}$
- $D = f(\text{cifrado}, \text{clave}) \rightarrow \text{mensaje}$
- Tanto el remitente como el receptor comparten la misma clave.
- La clave es secreta (solo el remitente y receptor la conocen)

Criptografía - Criptografía de clave asimétrica



- $E = f(\text{msj}, \text{clave}_{PK}) \rightarrow \text{cifrado}$
- $D = f(\text{cifrado}, \text{clave}_{SK}) \rightarrow \text{mensaje}$
- El remitente y el receptor tienen claves distintas.
- Cada participante tiene una clave secreta (conocida sólo por el participante) y otra pública (conocida por el resto de participantes).

Criptografía - Hashes - Definición



- $H = f(\text{datos}) \rightarrow \text{digest}$
- No hay clave 🚫🔑
- Función unidireccional
- Los datos (preimage) de entrada pueden tener tamaño arbitrario
- Los digests (hashes) tienen tamaño específico que depende del algoritmo

Criptografía - Hashes - Demo

Blockchain Demo

Hash

Block

Blockchain

Distributed

Tokens

Coinbase

SHA256 Hash

Data:

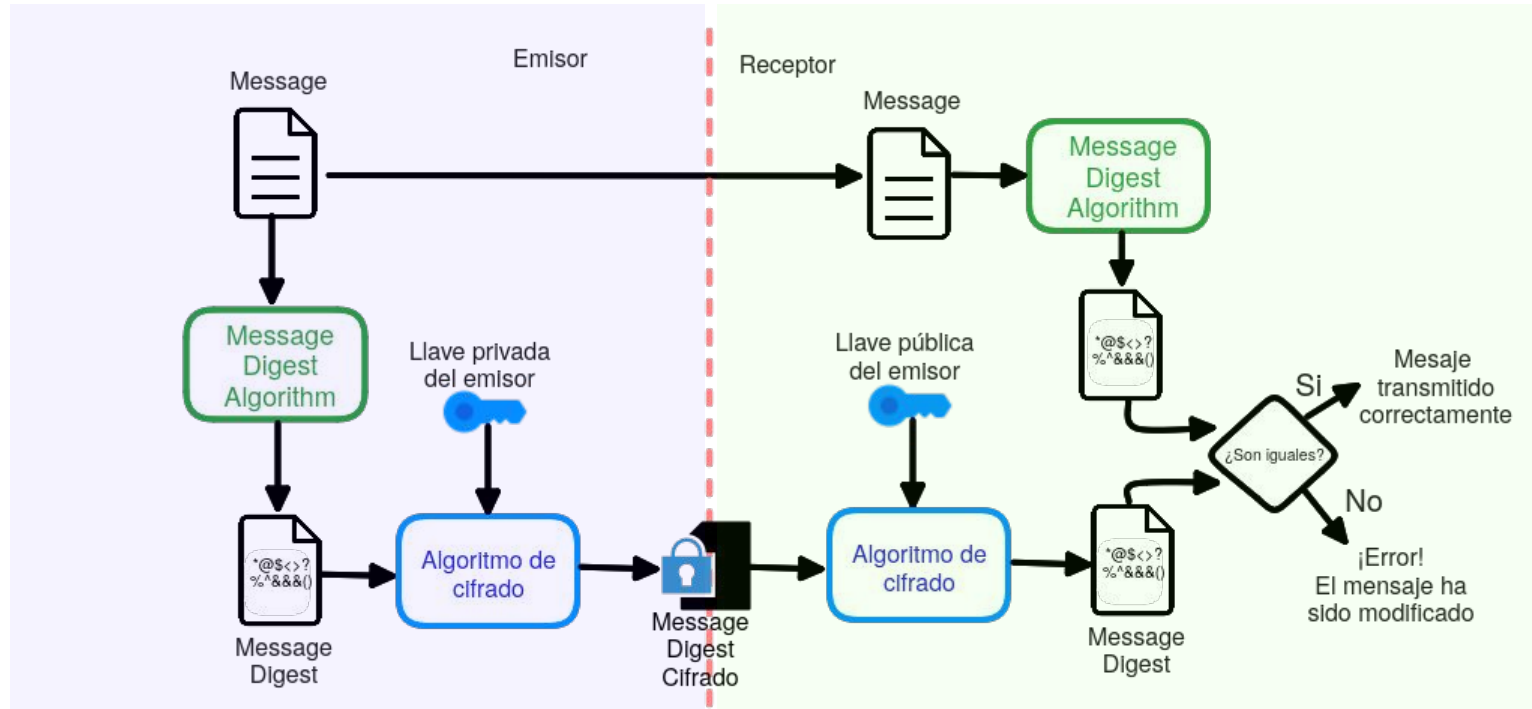
Hash:

<https://andersbrownworth.com/blockchain/hash>

Criptografía - Hashes - Propiedades

- **Efecto avalancha:** Un pequeño cambio en el valor de entrada x causa un gran cambio en el valor del hash $h = \text{hash}(x)$.
- **Resistencia a preimagen:** Dado un valor hash h , es computacionalmente inviable encontrar cualquier valor de entrada x tal que $h = \text{hash}(x)$.
- **Resistencia a segunda preimagen:** Dado un valor de entrada x , es computacionalmente inviable encontrar otro valor de entrada $x' \neq x$ tal que $\text{hash}(x) = \text{hash}(x')$. (También llamada “resistencia a colisión débil”).
- **Resistencia a colisión:** Es computacionalmente inviable encontrar dos valores de entrada x y x' distintos ($x' \neq x$) tal que $\text{hash}(x) = \text{hash}(x')$. (Resistencia a la colisión implica resistencia a segunda preimagen).
- **Efectivamente computable***

Criptografía - Firma digital - Definición



$$F = f(\text{msj}, \text{clave}_{SK}) \rightarrow (\text{msj}, \text{firma})$$

$$V = f(\text{msj}, \text{firma}, \text{clave}_{PK}) \rightarrow \text{bool}$$

Criptografía - Firma digital - Demo

Signatures

Sign

Verify

Message

Hi

Private Key

233636986358499457975939702495099620410278302332814248342583114120656902692

Sign

Message Signature

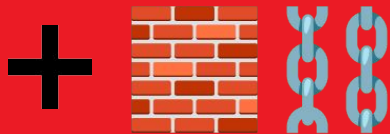
<https://andersbrownworth.com/blockchain/public-private-keys/keys>



<https://andersbrownworth.com/blockchain/public-private-keys/signatures>

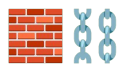
Criptografía - Firma digital - Propiedades

- **Integridad:** El receptor puede asegurarse de que el mensaje no fue alterado por un adversario.
- **Autenticación:** El remitente (e.g., Alice) puede probar que fue quien envió el mensaje al receptor (e.g, Bob).
- **Sin repudio de origen:** El receptor (e.g., Bob) puede probar que un remitente en específico (e.g., Alice) envió el mensaje. En otras palabras, el participante que firmó alguna información no puede luego negar que la firmó.

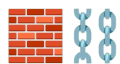


PRINCIPALES PRINCIPIOS

Previnendo vulnerabilidades hasta
llegar al concepto de Blockchain

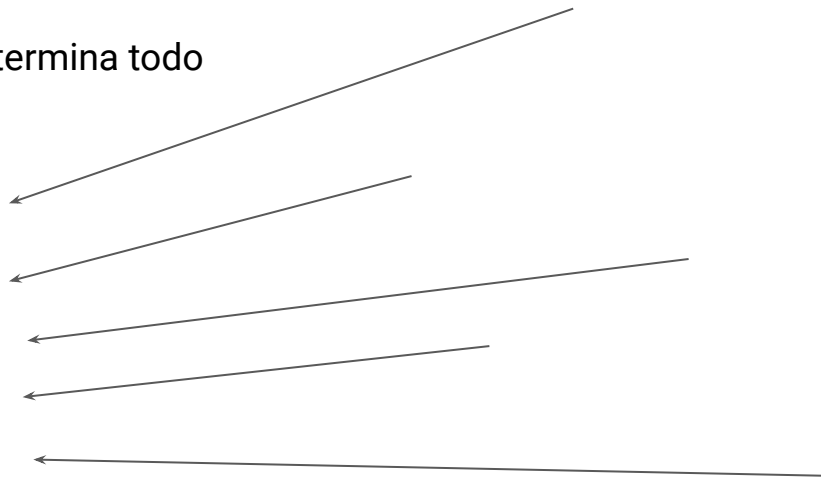
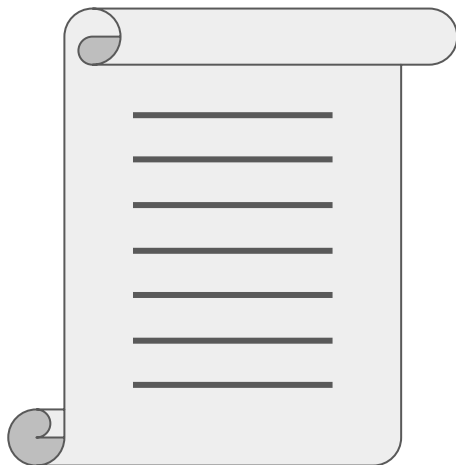


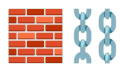
Intercambiar valor digital de forma segura entre partes que no confían los unos de los otros sin una autoridad centralizada



desde primeros principios - Fuente de verdad

- Necesitamos una fuente de verdad → Ledger → Lista de transacciones
- Todos los participantes pueden agregar transacciones
- En un mundo perfecto, ahí termina todo



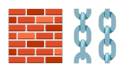


desde primeros principios - Prevenir abandono de deuda

Necesitamos que el ledger represente valor real (sino, nadie lo usa/respeta)



1. Para participar, ponés dinero físico en una caja fuerte a cambio de tener el mismo monto anotado en el ledger.
2. Cuando te querés retirar, se calcula lo que podés retirar basado en las transacciones.
3. Para prevenir sobregiro, no podés gastar más de lo que dice el ledger que está a tu nombre.



desde primeros principios - Prevención de receptor deshonesto

- Puede pasar que actor malicioso anote una transacción donde es el receptor sin permiso del remitente.
- Para prevenir esto, firmamos y verificamos las transacciones con firmas digitales

0 | \$100 | Alice → Bob | 0b1c...

1 | \$150 | Bob → Charlie | a74d...

...

$$F = f(\text{indice}, \text{plata}, \text{rem}_{PK}, \text{rec}_K, \text{clave}_{SKR}) \rightarrow \text{firma}_{REM}$$

$$V = f(\text{indice}, \text{plata}, \text{rem}_{PK}, \text{rec}_{PK}, \text{firma}_{REM}) \rightarrow \text{bool}$$



desde primeros principios - Firmar transacción - Demo

Transaction

SignVerify

Message

\$ 21.00

From: 0433e5491f714552a9b689c361e97 -> 048ff95e330e8dff4319559edf86

Signature

304502205ca277e74a9801015fd4a447251f00e0e9da15adc6b57316fea8507263c80f63022100c665d4de209d5e0e204efa4a250647

Verify

<https://andersbrownworth.com/blockchain/public-private-keys/transaction>



desde primeros principios - Prevención de remitente deshonesto

- Puede pasar que un actor malicioso anote una transacción (o varias) que envíen más valor del que está a su nombre.
- Para prevenir esto, el resto de participantes verifica que el remitente tenga suficiente valor a su nombre y firma la transacción.

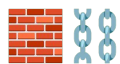
0 | \$100 | Alice → Bob | 0b1c... | 3fe...

1 | \$150 | Bob → Charlie | a74d... | 51a...

...

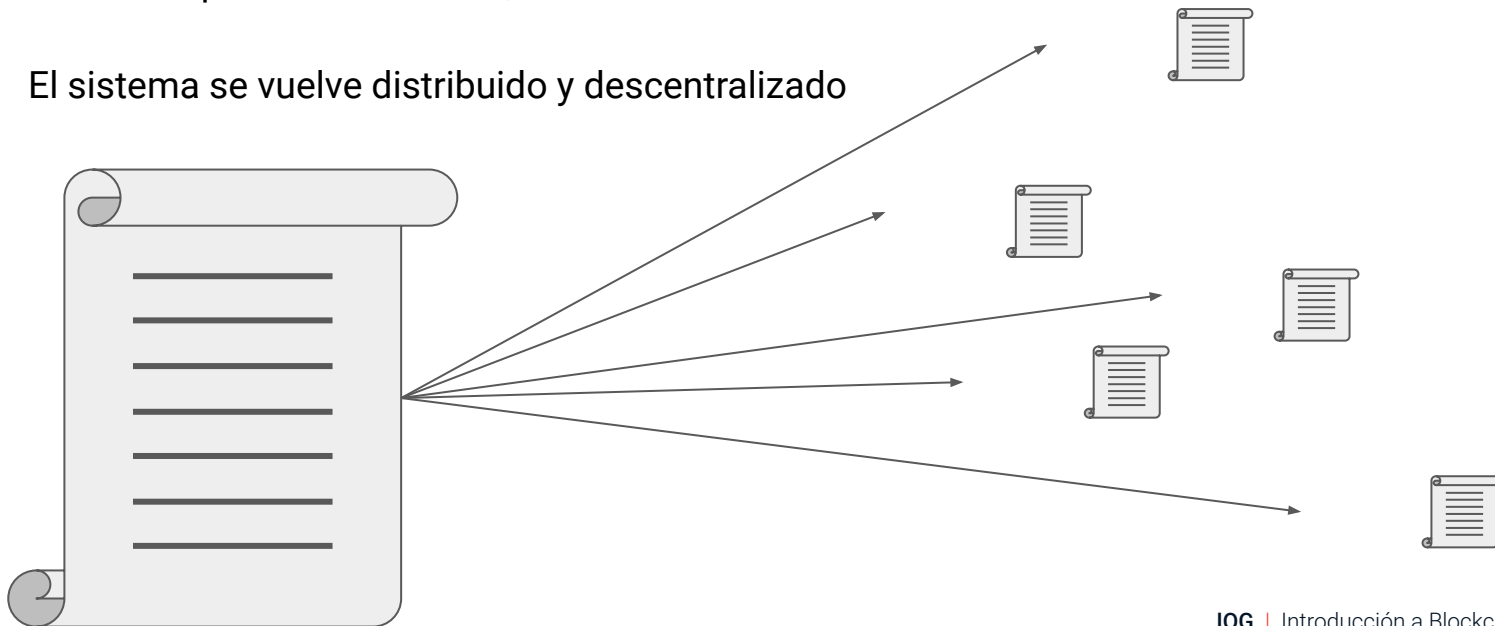
$$F = f(\text{índice}, \text{plata}, \text{rem}_{PK}, \text{rec}_{PK}, \text{firma}_{REM}, \text{clave}_{VI}) \rightarrow \text{firma}_{VI}$$

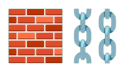
$$V = f(\text{índice}, \text{plata}, \text{rem}_{PK}, \text{rec}_{PK}, \text{firma}_{REM}, \text{firma}_{VI}) \rightarrow \text{bool}$$



desde primeros principios - Quién está a cargo del ledger?

- Cada participante tiene una copia del Ledger → Todos estan a cargo
- Cada vez que se hace una Tx, se avisa a todos
- El sistema se vuelve distribuido y descentralizado





desde primeros principios - Problemas de sistemas distribuidos

En nuestro sistema, hay todavía muchos problemas por resolver derivados de que los ledgers son un sistema distribuido:

- Networking
- Sincronicidad
- Tolerancia a fallos (fault tolerance)
- ...

Para los propósitos de entender cómo funciona una blockchain, vamos a centrarnos en la parte de teoría de juego. Osea:



QUÉ HACEMOS SI LOS LEDGERS SON DISTINTOS?



+ PROTOCOLO DE CONSENSO

Introducción a protocolos de consenso

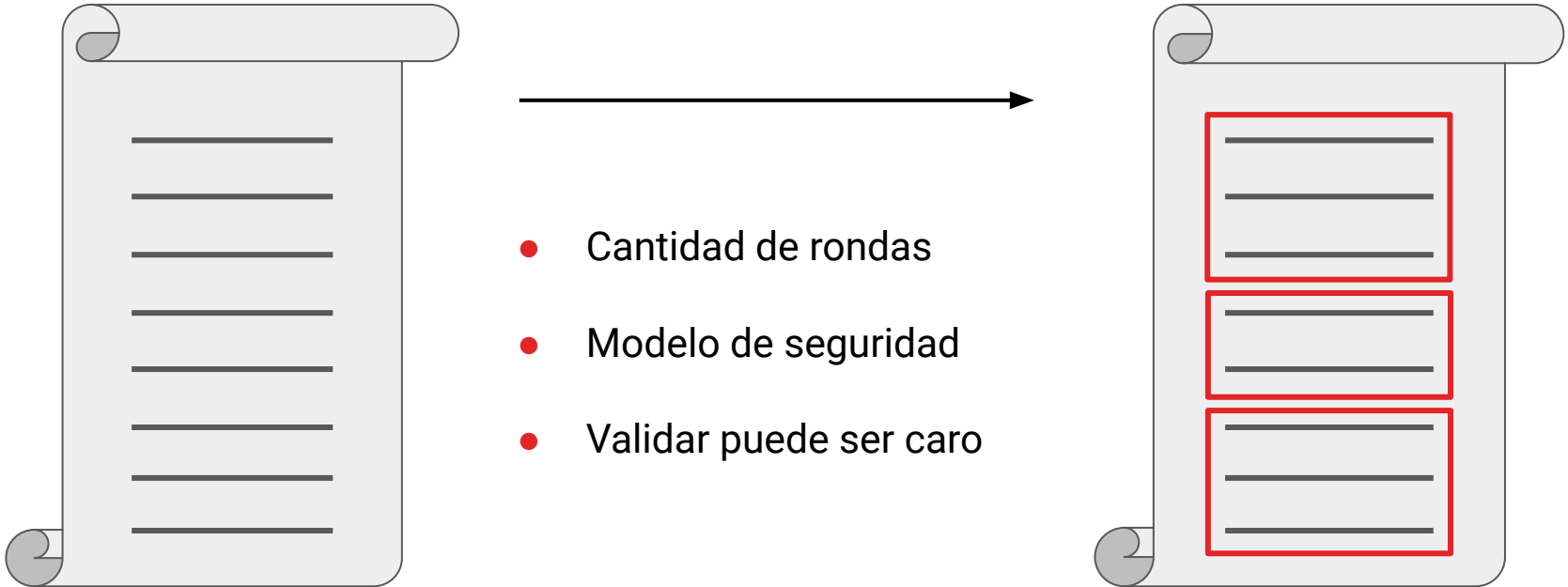
22

Serie de pasos/mecanismos realizados por los entes/nodos de un sistema descentralizado con el cual se llega a un acuerdo sobre un valor/información inclusive en presencia de cierta cantidad de nodos defectuosos o maliciosos

Vamos a hacer unos cambios en preparación al consenso

Protocolo de consenso - Bloques

Separamos las Tx en bloques y los validadores validan bloques enteros



Protocolo de consenso - Bloques - Demo

Block

Block: # 1

Nonce: 72608

Data:

Hash: 0000f727854b50bb95c054b39c1fe5c92e5ebcfa4bcb5dc279f56aa96a365e5a

Mine

<https://andersbrownworth.com/blockchain/block>

Protocolo de consenso - Orden - Problema

Necesitamos tener un orden criptográfico global de las Tx

Bloque				
David	Alice	100	3b..	✓
Alice	Bob	80	c1..	✓
Alice	Charlie	70	a7..	✗
4df31..				✓

?

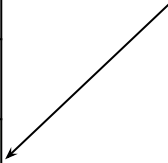
Bloque				
David	Alice	100	3b..	✓
Alice	Charlie	70	a7..	✓
Alice	Bob	80	c1..	✗
9f4ad..				✓

Bloque				
Alice	Eve	30	5b..	✗ / ✓
fe231..				✓

Protocolo de consenso - Orden - Solución

Orden criptográfico global → El hash del último bloque que vimos es parte del bloque actual

Bloque 1				
David	Alice	100	3b..	✓
Alice	Charlie	70	a7..	✓
9f4ad..				✓

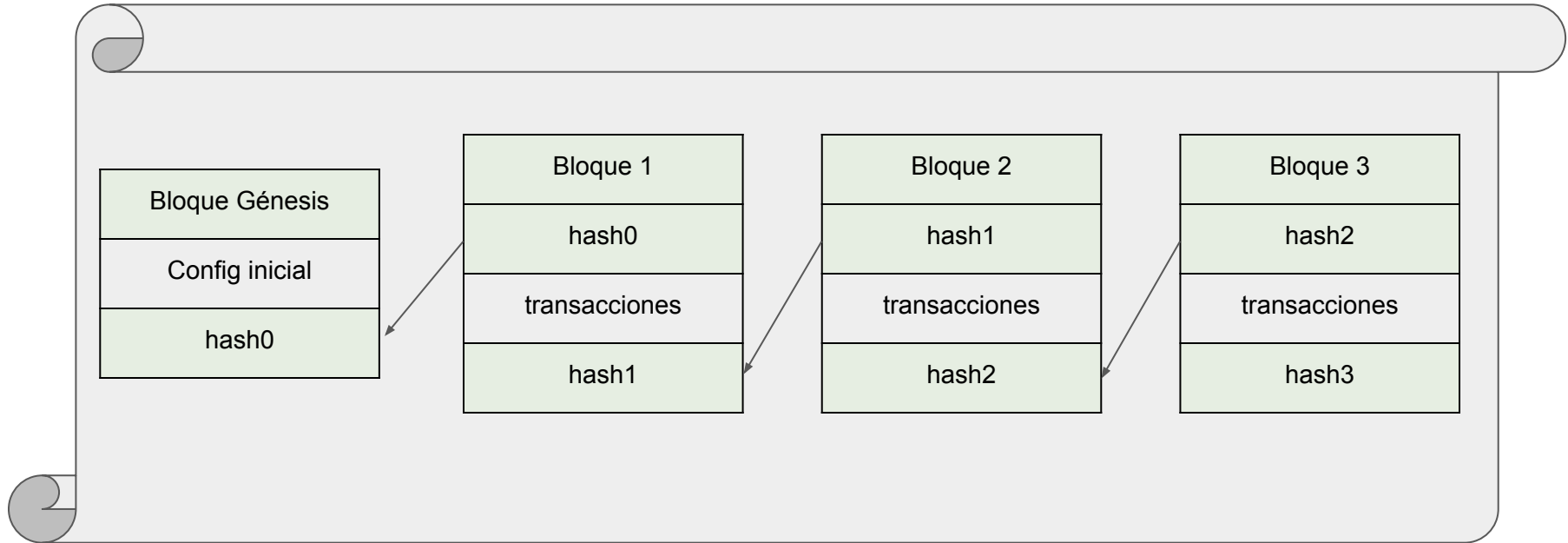


Bloque 2				
9f4ad..				
Alice	Eve	30	5b..	✓
a1c3e..				✓

- Bloque $N+1$ es imposible de obtener sin bloque N
- Bloque $N+1$ tiene que ser si o si el bloque después de N (orden inmutable)
- Es imposible insertar, eliminar, o modificar bloques (cadena inmutable)

Protocolo de consenso - Cadena de bloques

Ahora, el ledger es una cadena lineal de bloques. Cada nodo tiene su copia de la cadena.



Protocolo de consenso - Cadena de bloques - Demo

Blockchain

Block:

#

1

Nonce:

11316

Data:

Prev:

00

Hash:

000015783b764259d382017d91a36d206d0600e2cbb3567748f46e

Mine

Block:

#

2

Nonce:

35230

Data:

Prev:

000015783b764259d382017d91a36d206d0600e2cbb3567748f46e

Hash:

000012fa9b916eb9078f8d98a7864e697ae83ed54f5146bd84452c

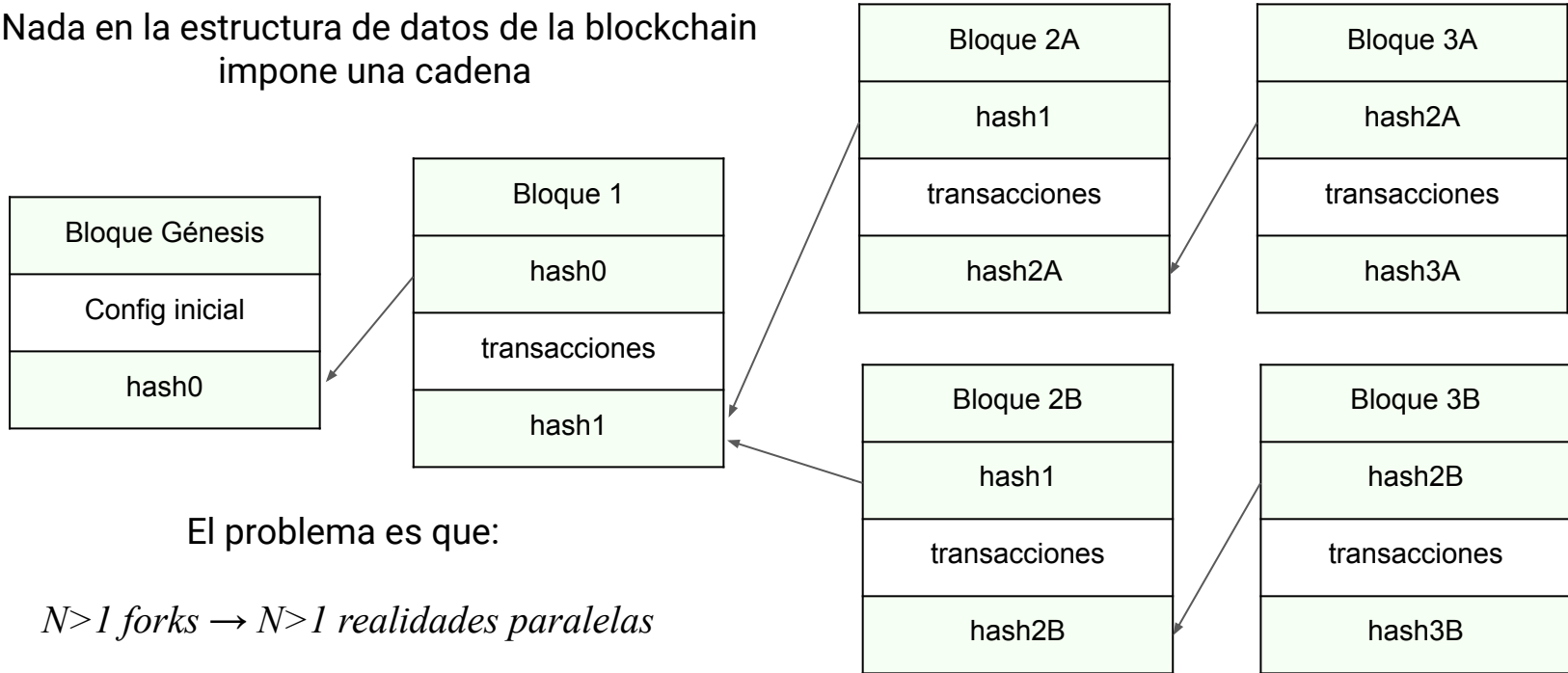
Mine

Block:	# 3
Nonce:	12937
Data:	
Prev:	000012fa9b916eb9078f8d98a7864e
Hash:	0000b9015ce2a08b61216ba5a07785
<div>Mine</div>	

<https://andersbrownworth.com/blockchain/blockchain>

Protocolo de consenso - Fork (bifurcación) - Problema

Nada en la estructura de datos de la blockchain impone una cadena



El problema es que:

$N > 1$ forks $\rightarrow N > 1$ realidades paralelas

Es posible gastar $N > 1$ veces la misma plata

Protocolo de consenso - Protocolos de Consenso

- Incluso en ausencia de jugadores maliciosos, no se pueden evitar completamente los forks (errores de red, etc.).
- Un Protocolo de Consenso (*Consensus Protocol*) asegura que los forks no se vuelvan demasiado profundos.
- El Prefijo Común (*Common Prefix*) debería mantenerse: Después de "descartar" los últimos k bloques, cada parte tiene la misma visión de la blockchain (el camino más largo en el árbol).
- El derecho a crear bloques está ligado a algún activo que "mayormente" pertenece a las partes honestas. Para Bitcoin: **Proof-of-Work** (poder de cómputo). Para Cardano: **Proof-of-Stake** (Ouroboros).

Protocolo de consenso - Consenso mediante Proof of Work - PoW

- Al validar el bloque, el validador debe modificar un parámetro (nonce) hasta que el hash comience con una cierta cantidad de bits cero.
- La cantidad de ceros elegida depende del **trabajo necesario para calcular un hash** con el hardware actual y la **probabilidad de que te toque un hash con esa cantidad de ceros**.
- El **trabajo promedio** requerido es **exponencial en la cantidad de bits cero requeridos**, pero se puede verificar ejecutando un solo hash.
- **El hash es la prueba** de que el nodo invirtió una cierta cantidad de poder computacional y, por ende, ganó el privilegio de **agregar ese bloque** a la blockchain.

<https://andersbrownworth.com/blockchain/block>

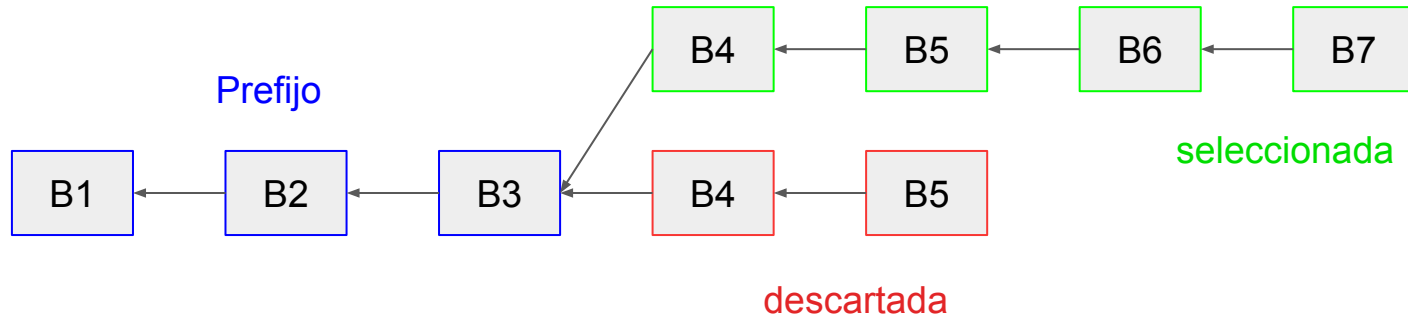
Protocolo de consenso - Consenso mediante Proof of Work - Protocolo Simplif.

1. Se envían nuevas Tx a todos los nodos.
2. Cada nodo junta las nuevas Tx en un bloque B_1 .
3. Cada nodo trabaja para obtener la prueba de trabajo (hash) para ese nodo.
4. Cuando un nodo encuentra el hash h_1 , difunde el nuevo bloque B_1 al resto de nodos
5. El resto de nodos sólo aceptan el bloque B_1 si las transacciones son correctas
6. El resto de nodos expresa su aceptación del bloque B_1 trabajando en la creación del siguiente bloque B_2 utilizando el hash h_1 como hash anterior.

<https://andersbrownworth.com/blockchain/blockchain>

Protocolo de consenso - Consenso mediante Proof of Work - Cadena más larga

- Los nodos siempre consideran que **la cadena más larga es la correcta** (cadena con mayor poder computacional invertido) y seguirán trabajando para extenderla.
- Si dos nodos transmiten **diferentes versiones del siguiente bloque** simultáneamente, cada nodo trabaja en la primer rama que recibió y guarda la otra en caso de que se alargue.
- **El empate se romperá** cuando se encuentre la **siguiente prueba de trabajo** y una rama se alargue; los nodos que estaban trabajando en la otra rama cambiarán a la más larga.



Protocolo de consenso - Consenso mediante Proof of Work - Incentivos

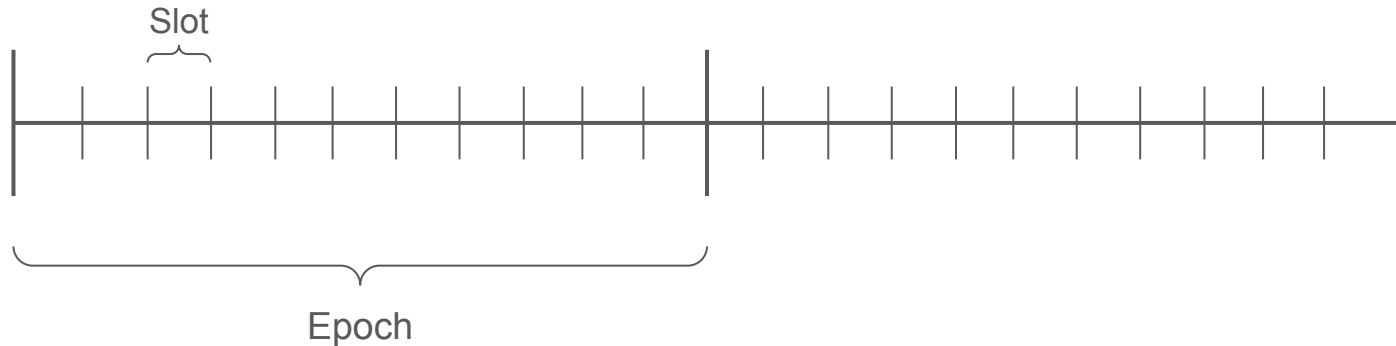
- El protocolo es seguro siempre y cuando la mayoría de poder computacional sea honesto.
- Para **incentivar** a los nodos, los nodos pueden agregar una transacción pagándose a sí mismos. De donde sale el dinero de este pago depende de la blockchain:
 - Fees (comisiones) cobradas a los que realizan transacciones son usualmente parte.
 - Crear más monedas.
 - Sacar dinero de un fondo común.
 - Combinación de anteriores y/u otras.
- Si un atacante codicioso es capaz de reunir más potencia de CPU que todos los nodos honestos, tendría que elegir entre usarla para defraudar a las personas robando sus pagos o usarla para ganar dinero validando transacciones.
- Debería encontrar más rentable seguir las reglas. Reglas que le favorecen con más dinero del sistema. Incentivando evitar socavar el sistema y la validez de su propia riqueza.

Protocolo de consenso - Consenso mediante Proof of Stake - Stake

- En los ámbitos que vimos, PoS (Proof of Stake) no es tan distinto a PoW. En ambos:
 - Se juntan las Tx en bloques que son validados por validadores
 - Importa la cadena más larga
 - Se incentivan a los nodos para que trabajen
- La diferencia clave, y a partir de la cual derivan otras diferencias, es en **cómo elegimos a los nodos líderes**.
- Le decimos “líderes” a los nodos encargados de **agregar bloques** a la blockchain. En PoW, la selección de líder es **proporcional al poder computacional** pero **inherentemente aleatoria**.
- En PoS seleccionamos los líderes basados en su “**stake/apuesta**” (cantidad de monedas que tiene el nodo). Esto es significativamente menos costoso, pero requiere **inyectar aleatoriedad**.

Protocolo de consenso - Consenso mediante Proof of Stake - Epoch y Slot

El tiempo está dividido en slots y una cierta cantidad de slots son una epoch



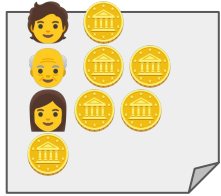
Protocolo de consenso - Consenso mediante Proof of Stake - Ronda slot

1. Determinar la cadena más larga:

Adoptar una **nueva cadena** si:

- a. ...es **más larga** pero no difiere más de k **bloques** de la local o...
- b. ...difiere **más de k bloques** pero tiene **mayor densidad** cerca del punto de bifurcación.

De lo contrario, seguir con local.

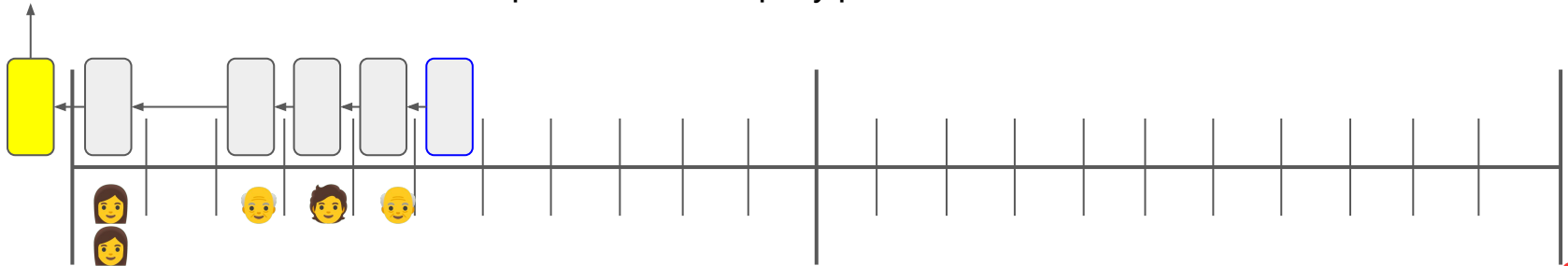


2. Determinar líder de slot con lotería privada usando VRF

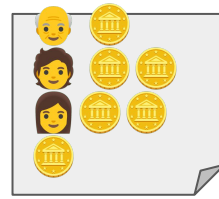
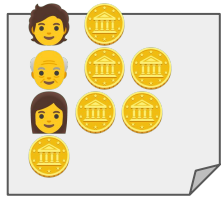
$$Lider = f(\text{👤}_S, \text{👤}_S, \text{👤}_S, \text{slot}, \text{seed}) \rightarrow \text{👤} / \text{👤} / \text{👤}$$

3. Líder de slot: Empacar Tx en bloque y publicarlo

+
Semilla
aleatoria



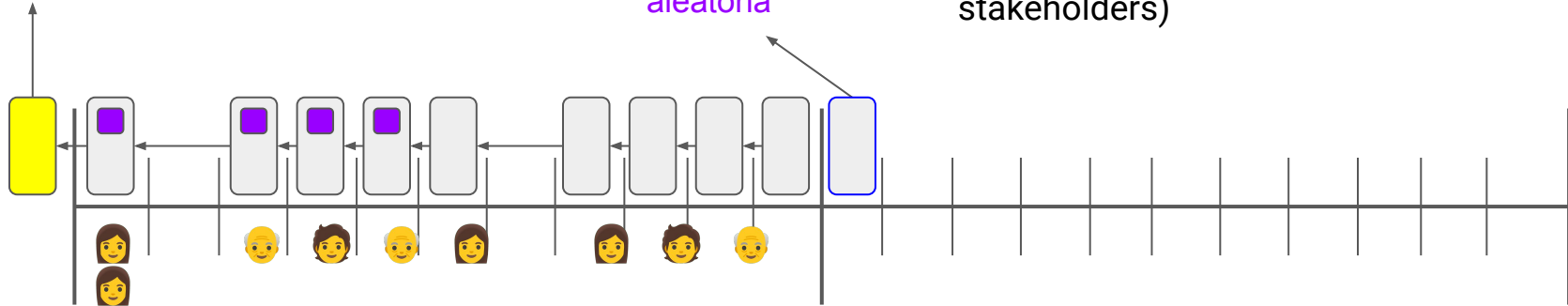
Protocolo de consenso - Consenso mediante Proof of Stake - Ronda Epoch



1. Todo lo que se hace para la ronda de slot
2. Recalcular distribuciones de stake
3. Refrescar semilla aleatoria para la lotería de la siguiente epoch.
4. Gestión de recompensas (e.g., pagar stakeholders)

+
Semilla
aleatoria

+
Nueva
semilla
aleatoria



¿Preguntas?



INPUT | OUTPUT