librería de Satoshi

₿4OS

</bitcoin for open source>

Clase 2: Arquitectura Técnica y Modelo de Seguridad

COURSE OVERVIEW

1. Flujos de protocolo
2. Estructuras de datos
3. Análisis de privacidad
4. Gestión de riesgos en Cashu

B4OS

¿Qué debes confiar y qué debe permanecer oculto cuando usas un protocolo de dinero digital?

B4OS

¿Qué debes confiar y qué debe permanecer oculto cuando usas un protocolo de dinero digital?

Debe estar OCULTO:

- Tu identidad
- Historial de pagos
- Saldos actuales
- Relaciones comerciales

Debes CONFIAR en:

- Validez criptográfica
- No double-spending
- Respaldo económico
- Disponibilidad del sistema

B4OS

Cashu

- **Sistema de eCash moderno sobre Bitcoin**
- **Creador: Calle**
- **Año: 2022**
- **Interoperable con Lightning Network**

B4OS

Denominación de los billetes

Denominación por potencias de dos:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024

Si queremos mintear 5 sats, el mint nos devuelve:

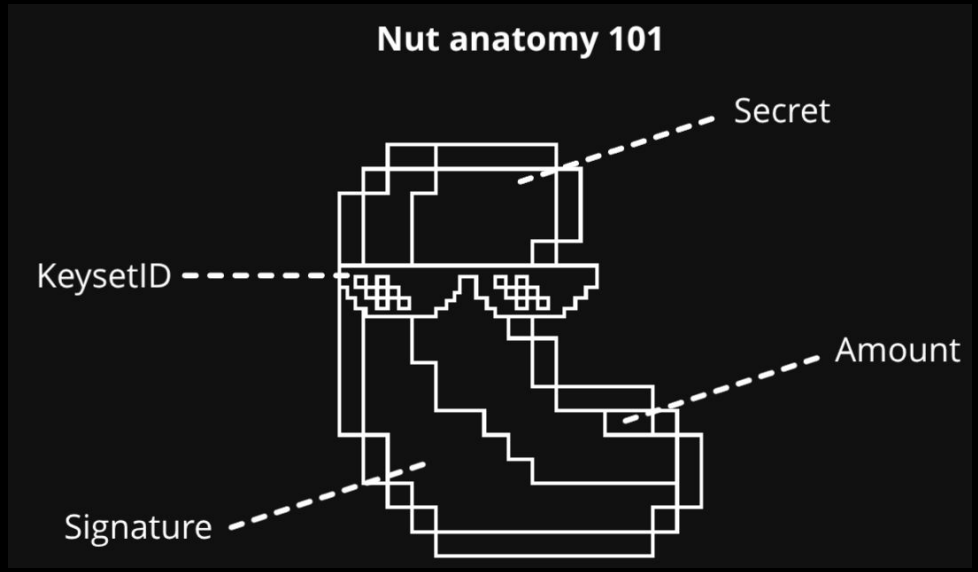
1 cashu de valor 4

1 cashu de valor 1

B4OS

Pregunta de Diseño

¿Por qué usar potencias de 2 en lugar de denominaciones arbitrarias como 1, 5, 10, 20?



Componentes Clave

- **amount: Denominación del token (potencias de 2)**
- **secret: Valor aleatorio único (previene double-spend)**
- **C: Punto de curva elíptica (signature blind)**
- **id: Identifica el keyset del mint usado**


```
{
"Secret":   "1ea6f7f0e907f5841285a3815a74c84e6d710b0b458b959d68c9d60aabbb40ac",
"C":        "02c6ee49eabc2bb027973ab5c79cdf379a74fff0731ccbae3b0d419b3d635075e6",
"Amount":   1,
 "Id":      "00b4cd27d8861a44"
}
```


https://snf2025.gandlaf.com/#/14

B4OS

Paso 1:
Envío de sats al Mint

Paso 2:
Emisión de cashu y entrega al usuario

Creación de Cashu por Alice

Paso 1:
Envío de sats al Mint

Paso 3:
Transferencia de Cashu a otra persona

Paso 2:
Emisión de cashu y entrega al usuario

Envío de Cashu a Bob


Paso 1:
Envío de Cashu al Mint original

Paso 3:
El nuevo mint crea nuevos Cashu y te los entrega

Paso 2:
El mint quema tus tokens y envía los sats vía Lightning Network al nuevo Mint

BDHKE: Combina la seguridad de Diffie-Hellman con blind signatures para crear tokens perfectamente privados pero verificables

Lo que ve el mint:

- ✅ Solo que emitió algo
- ✅ Valor por el que firmó
- ❌ Contenido del token
- ❌ A quién pertenece
- ❌ Historial de uso

Lo que tiene Alice

- ✅ Token con firma ciega válida
- ✅ Prueba criptográfica de autenticidad
- ✅ Privacidad total
- ✅ Transferible sin permiso

B4OS

Objetivos

1. Conocer especificaciones técnicas del protocolo Nuts
2. Describir flujos Mint - Swap - Melt en detalle
3. Conocer los riesgos al usar el protocolo y cómo mitigarlos

B40S

¿Qué son los NUTs?

Notation, Usage, Terminology

https://github.com/cashubtc/nuts

- Especificaciones técnicas estandarizadas
- Garantizan interoperabilidad entre implementaciones
- Definen formatos de datos y comportamientos

Importancia: Sin especificaciones claras, cada wallet/mint funcionaría diferente, rompiendo la interoperabilidad

B4OS

¿Qué son los NUTs?

Nuts esenciales para cualquier implementación de Cashu

Mandatory

NUT #	Description
00	Cryptography and Models
01	Mint public keys
02	Keysets and fees
03	Swapping tokens
04	Minting tokens
05	Melting tokens
06	Mint info

B4OS

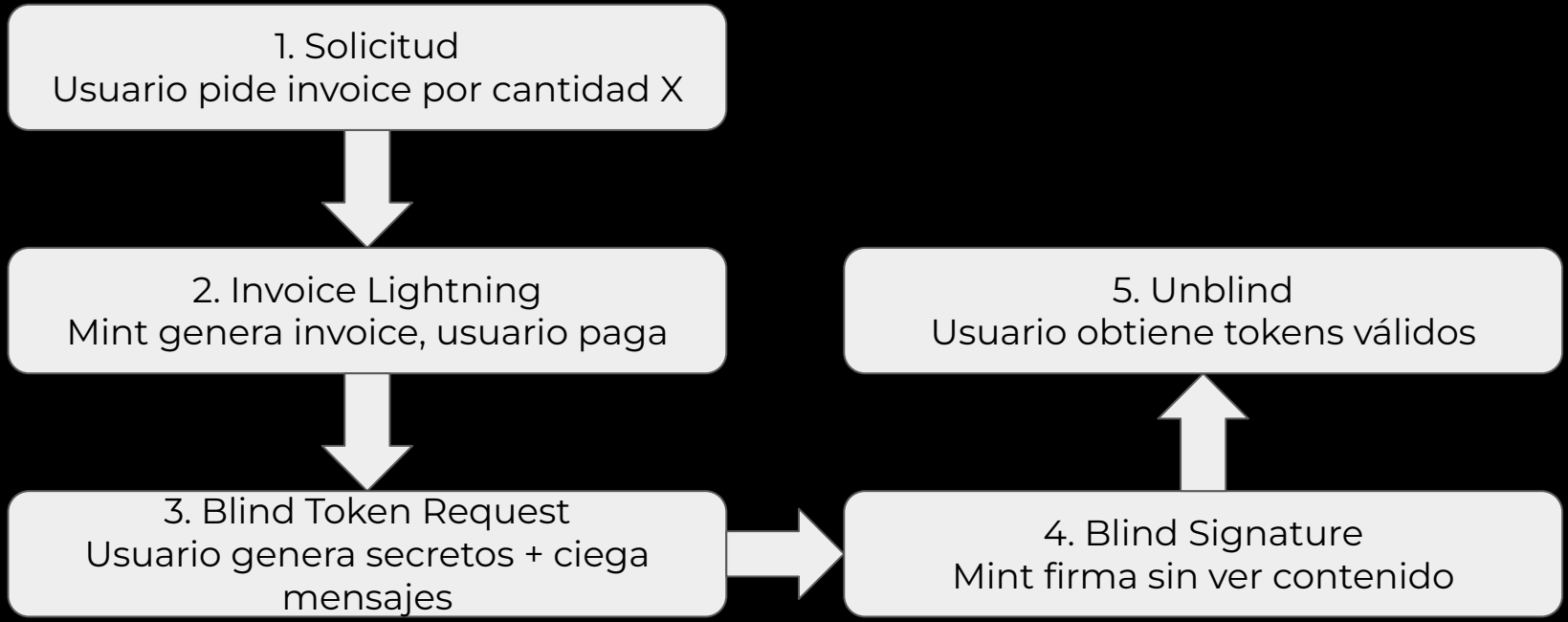
¿Qué son los NUTs?

#	Description	Wallets	Mints
07	Token state check	Nutshell, Nutstash, cashu-ts, cdk-cli, Minibits, macadamia	Nutshell, cdk-mintd, nutmix
08	Overpaid Lightning fees	Nutshell, Nutstash, cashu-ts, cdk-cli, Minibits, macadamia	Nutshell, cdk-mintd, nutmix
09	Signature restore	Nutshell, cdk-cli, Cashu.me, Minibits, macadamia	Nutshell, cdk-mintd
10	Spending conditions	Nutshell, cdk-cli, cashu-ts, Minibits	Nutshell, cdk-mintd, nutmix
11	Pay-To-Pubkey (P2PK)	Nutshell, cdk-cli, Cashu.me, Minibits	Nutshell, cdk-mintd, nutmix
12	DLEQ proofs	Nutshell, cdk-cli, cashu-ts	Nutshell, cdk-mintd, nutmix
13	Deterministic secrets	Nutshell, cashu-ts, cdk-cli, macadamia, Minibits	-

¿Qué son los NUTs?

14	Hashed Timelock Contracts (HTLCs)	Nutshell, cdk-cli	Nutshell, cdk-mintd, nutmix
15	Partial multi-path payments (MPP)	Nutshell, cdk-cli	Nutshell, cdk-mintd, nutmix
16	Animated QR codes	Cashu.me, macadamia, Minibits	-
17	WebSocket subscriptions	Nutshell, cdk-cli, Cashu.me, Minibits	Nutshell, cdk-mintd, nutmix
18	Payment requests	Cashu.me, Boardwalk, cdk-cli, Minibits	-
19	Cached Responses	-	Nutshell, cdk-mintd
20	Signature on Mint Quote	cdk-cli, Nutshell	cdk-mintd, Nutshell
21	Clear authentication	Nutshell, cdk-cli	Nutshell, cdk-mintd, nutmix
22	Blind authentication	Nutshell, cdk-cli	Nutshell, cdk-mintd, nutmix
23	Payment Method: BOLT11	Nutshell, cdk-cli	Nutshell, cdk-mintd, nutmix
24	HTTP 402 Payment Required	-	-

Proceso de mint de tokens Cashu



Proceso de mint de tokens Cashu

¿Qué información aprende el mint durante este proceso y qué permanece oculto?

B4OS

Proceso de swaps de tokens Cashu

¿Por qué SWAP?

Combinar tokens: 1 + 1 + 2 = 4

Dividir tokens: 8 = 4 + 2 + 2

Refrescar tokens: Nuevos secretos para privacidad

Análisis de privacidad:

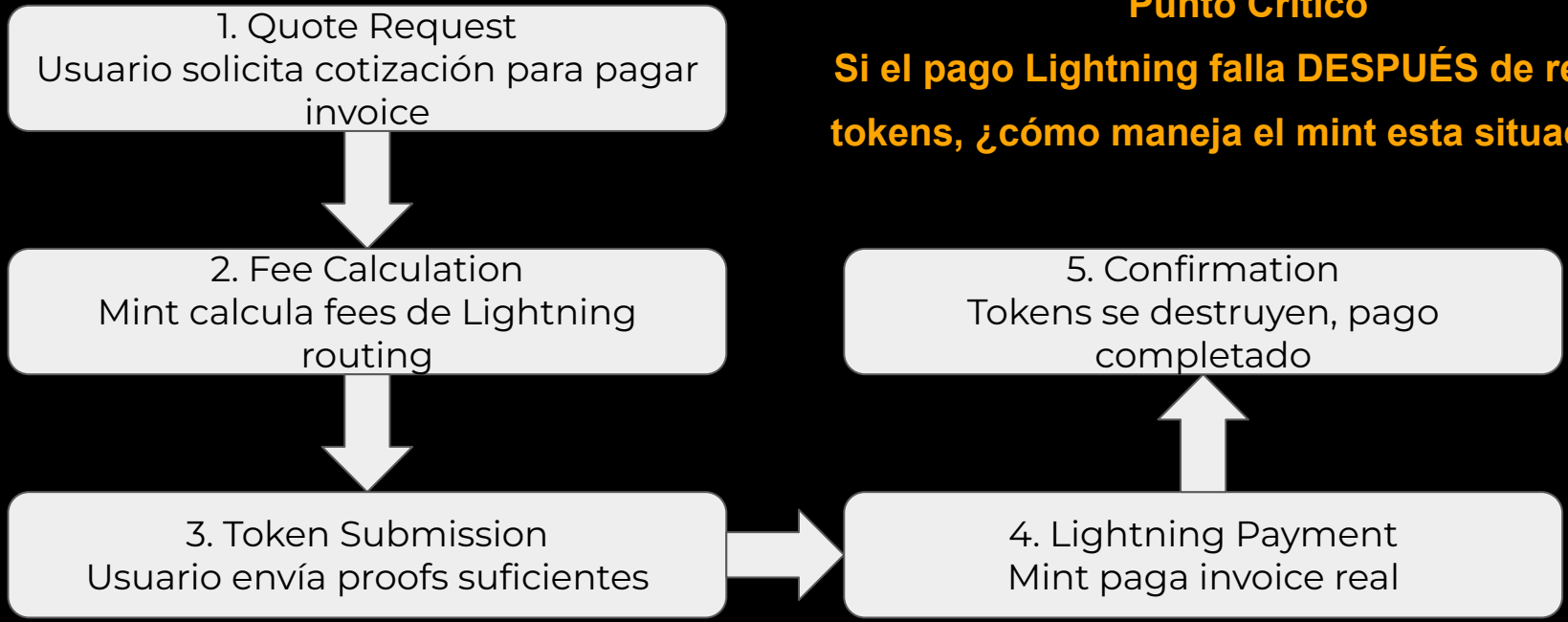
El mint ve: Cantidad total, timing de operación

El mint NO ve: Origen de tokens, destino final, identidad del usuario

B4OS

Proceso de swaps de tokens Cashu

Proceso de melt de tokens Cashu

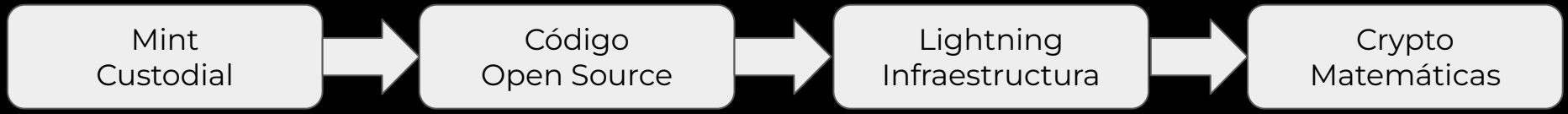


Punto Crítico

Si el pago Lightning falla DESPUÉS de recibir tokens, ¿cómo maneja el mint esta situación?

B4OS

Modelo de confianza



No Necesitas Confiar

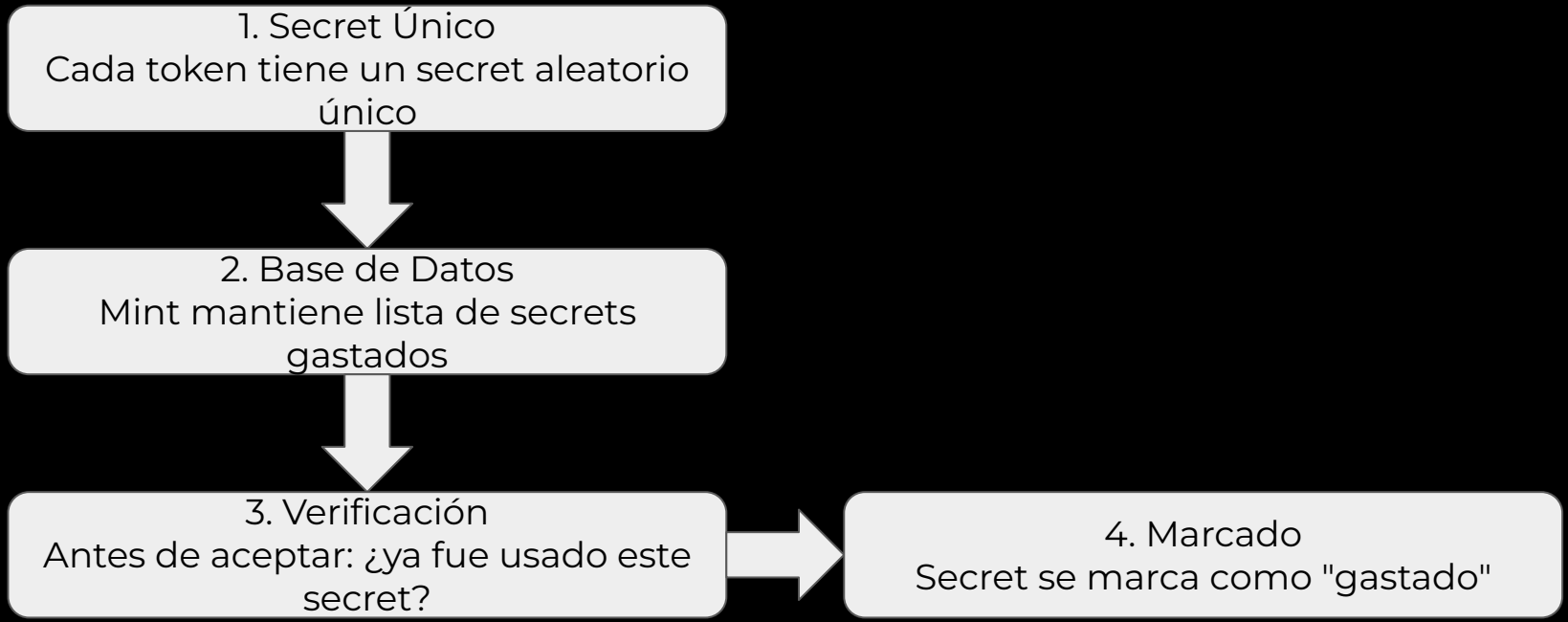
- **Privacidad criptográfica**
- **Validez de firmas**
- **Protocolo de blind signatures**
- **Matemáticas de curvas elípticas**

Debes Confiar En

- **Mint no desaparecerá**
- **Mint tiene respaldo en Lightning**
- **Implementación del código**
- **Infraestructura de red**

B4OS

Mecanismo de Control



B4OS

Mecanismo de Control

Pregunta Técnica

¿Cómo detectarías y defenderías contra un intento de double-spend si fueras el operador del mint?

B4OS

Riesgos de Seguridad

Mint Malicioso

Puede robar fondos o desaparecer

Pérdida de Seed

Sin respaldo = fondos perdidos

Análisis de Metadatos

Correlación de patrones

Fallas de Conectividad

Problemas de red/mint offline

Ataques Criptográficos

Matemáticas bien establecidas

B4OS

Mitigaciones de riesgo

Diversificación de Mints

- No pongas todos los tokens en un solo mint
- Distribuye riesgo entre operadores
- Monitorea salud de mints regularmente

B4OS

Mitigaciones de riesgo

Backups Seguros

- Respalda seed phrases regularmente
- Almacenamiento offline seguro
- Prueba procesos de recuperación

Ejercicio: Simula pérdida de wallet y restaura desde seed.

¿Qué información necesitas para recuperar completamente?

Ejercicios para la casa

1. **MINT Flow**

   Documentar cada paso del proceso mint → tokens

1. **SWAP Flow**

   Cambiar denominaciones y observar metadata

1. **MELT Flow**

   Redimir tokens de vuelta a Lightning

B4OS

Tiempo de caricaturas

Alice quiere enviar 1000 sats a Bob como regalo de cumpleaños.

Bob está en un vuelo de 8 horas sin internet.

Alice le envía un token por WhatsApp...

B4OS

Tiempo de caricaturas

Charlie (hacker) intercepta el mensaje de WhatsApp.

¿Puede Charlie gastar los tokens antes que Bob aterrice?

B4OS

Pay-to-Public-Key (NUT-11)

¿Qué es P2PK?

Mecanismo que permite bloquear tokens a una clave pública específica, requiriendo una firma digital válida para gastarlos.

Diferencia Clave

- **Tokens Normales: cualquiera puede gastarlos**
- **Tokens P2PK: Solo el dueño de la clave privada**

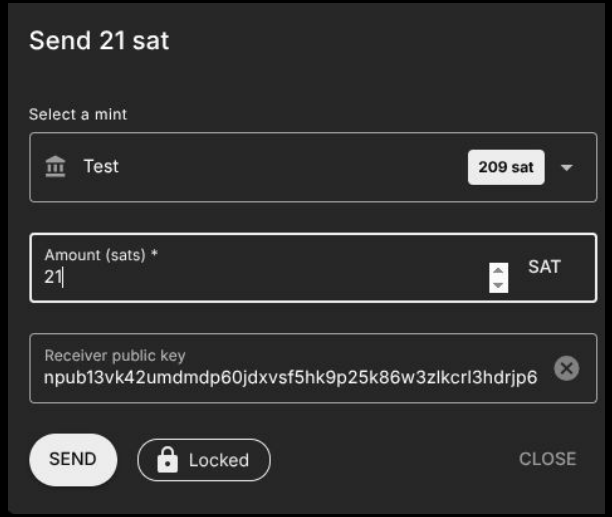
Casos de Uso

- Pagos Offline: Envío seguro sin conexión inmediata
- Gifts/Vouchers: Tokens dirigidos a personas específicas
- Comercio P2P: Garantías en intercambios
- Custody: Tokens que requieren autorización específica

B4OS

Funcionamiento

- Crear: Token se bloquea con pubkey del destinatario
- Transferir: Se envía el token (seguro en tránsito)
- Gastar: Destinatario firma con su privkey para desbloquearlo



Ventajas: Seguridad, control, pagos dirigidos

Desventajas: Menos privacidad, más complejidad

B4OS

Preguntas y Dudas

B40S

Clase 2: Arquitectura Técnica y Modelo de Seguridad

@Forte11

libreriadesatoshi.com & b4os.dev

B4OS