Sistema de Wallet Segura en NFC para Reciclaje

📝 Resumen

Sistema de almacenamiento seguro de wallets de criptomonedas usando NFC y base de datos para gestión de reciclaje. Combina índices de búsqueda rápida con encriptación AES-256 para proteger datos sensibles con contraseña del usuario.

🔑 Características Principales

- ✅ **Flexible**: Dos formas de autenticación (DNI o NFC)
- ✅ **Seguro**: Encriptación AES-256 con 100k iteraciones PBKDF2
- ✅ **Eficiente**: Búsquedas rápidas + datos protegidos por contraseña
- ✅ **Privado**: Clave privada solo se desencripta al firmar

🏗️ Arquitectura del Sistema

📊 Base de Datos

json

```
{
  // ÍNDICES DE BÚSQUEDA (sin encriptar)
  "email": "usuario@example.com",
  "dni": "12345678A",
  "publicAddress": "0x742d35Cc...",

  // DATOS PROTEGIDOS (encriptados con password del usuario)
  "encryptedDNI": "base64...",
  "encryptedPublicAddress": "base64...",
```

```
  "encryptedPrivateKey": "base64..."
}
```

¿Por qué este diseño?

- **Índices sin encriptar**: Permiten búsquedas rápidas por email, DNI o dirección
- **Datos encriptados**: Verifican que el password es correcto y protegen la clave privada
- **Doble capa**: Los índices permiten encontrar al usuario, los datos encriptados permiten autenticarlo

💾 Chip NFC

json

```
{
  "publicAddress": "0x742d35Cc...",
  "encryptedDNI": "base64...",
  "encryptedPrivateKey": "base64..."
}
```

Tamaño: ~250 bytes **Capacidad requerida**: Cualquier NFC estándar (512+ bytes)

🎯 Tres Formas de Autenticación

1. Login con DNI + Password

```
Usuario introduce: DNI + Password
↓
Sistema busca en BD: WHERE dni = '...'
↓
Sistema desencripta: encryptedDNI con password
↓
Sistema compara: DNI desencriptado == dni (índice)
```

↓

✅ Autenticado si coinciden

2. Login con NFC + Password

Usuario acerca NFC + introduce Password

↓

Sistema lee: publicAddress del NFC (sin encriptar)

↓

Sistema busca en BD: WHERE publicAddress = '...'

↓

Sistema desencripta: encryptedPublicAddress con password

↓

Sistema compara: Dirección desencriptada == publicAddress (índice)

↓

✅ Autenticado si coinciden

🔄 Flujo Completo de Uso

REGISTRO

1. Usuario introduce:

   - Email: usuario@example.com

   - DNI: 12345678A

   - Password: MiPassword123!

2. Sistema genera wallet:

   - Public Address: 0x742d35Cc...

- Private Key: 0x4c0883a...

3. Sistema encripta con password:

- encryptedDNI

- encryptedPublicAddress

- encryptedPrivateKey

4. Sistema guarda en BD:

✓ email, dni, publicAddress (índices)

✓ encryptedDNI, encryptedPublicAddress, encryptedPrivateKey

5. Sistema graba en NFC:

✓ publicAddress (sin encriptar)

✓ encryptedDNI

✓ encryptedPrivateKey

RECICLAJE (uso rápido)

1. Usuario acerca NFC al punto de reciclaje

2. Sistema lee publicAddress (sin encriptar)

3. Sistema registra: X kilos de plástico → publicAddress

4. Transacción guardada en blockchain

✅ Sin password necesario

LOGIN PRIVADO

Opción A: DNI + Password

Opción B: NFC + Password

↓

Sistema busca usuario en BD

Sistema desencripta dato correspondiente

Sistema verifica coincidencia

↓

✅ Acceso concedido al panel privado

📊 Ver historial de reciclaje

💰 Consultar saldo

FIRMAR TRANSACCIÓN

1. Usuario autenticado solicita enviar fondos

2. Usuario confirma con password

3. Sistema desencripta privateKey temporalmente

4. Sistema firma transacción

5. Sistema elimina privateKey de memoria

✅ Transacción enviada a blockchain

🛡️ Capas de Seguridad

Capa	Protección	Contra
Índices públicos	Búsqueda rápida	-
AES-256-CBC	Encriptación militar	Robo de BD

PBKDF2 100k	Derivación lenta	Fuerza bruta
Password único	Clave por usuario	Ataques masivos
Verificación cruzada	Índice vs encriptado	Passwords incorrectos

💡 Ventajas del Sistema

Característica	Beneficio
Doble autenticación	DNI o NFC según contexto
Búsquedas O(1)	Índices sin encriptar permiten queries rápidas
Password único	Usuario solo recuerda una contraseña
Reciclaje sin fricción	NFC lee dirección pública sin password
Privacidad garantizada	Clave privada siempre encriptada
Sin servidor central	Datos en BD local + NFC físico

📱 Casos de Uso

🔄 Reciclaje diario

Usuario → Acerca NFC

Sistema → Lee publicAddress (0x742d35Cc...)

Sistema → Registra: 2kg plástico + 1kg papel

Blockchain → Transacción guardada

✅ Sin login, sin password

📊 Consulta de datos

Usuario → Introduce DNI + Password

Sistema → Busca y autentica

Panel → Muestra: 150kg reciclados, 45 tokens ganados

✅ Datos privados accesibles

💸 Envío de fondos

Usuario → Login con Email + Password

Usuario → Solicita enviar 10 tokens

Sistema → Desencripta privateKey

Sistema → Firma transacción

Sistema → Elimina privateKey

✅ Fondos transferidos

🏪 Punto de reciclaje público

Máquina → Lee NFC automáticamente

Display → "Hola 0x742d35Cc... ¿Qué reciclas hoy?"

Usuario → Deposita materiales

Sistema → Registra en blockchain sin password

✅ Experiencia fluida

🔐 Matriz de Seguridad

Si roban la Base de Datos:

- ✅ Pueden ver: email, DNI, publicAddress (datos públicos)
- ❌ NO pueden: usar la wallet sin password
- ❌ NO pueden: desencriptar la clave privada
- ❌ NO pueden: firmar transacciones

Si roban el NFC:

- ✅ Pueden ver: publicAddress (dato público)
- ❌ NO pueden: ver el DNI (encriptado)
- ❌ NO pueden: acceder a la clave privada (encriptada)
- ⚠️ Pueden: reciclar a nombre del usuario (pero esto es beneficioso para él)

Si comprometen un admin de BD:

- ✅ Puede ver: datos de índice (públicos)
- ❌ NO puede: desencriptar sin passwords de usuarios
- ❌ NO puede: robar fondos

⚙️ Tecnología

- **Lenguaje**: TypeScript
- **Encriptación**: AES-256-CBC
- **Hash**: SHA-256
- **Derivación**: PBKDF2 (100,000 iteraciones)
- **Base de Datos**: Cualquiera (SQL, NoSQL)
- **NFC**: Chips estándar ISO 14443 (512+ bytes)

🚀 Implementación

Estructura de datos

typescript

```
interface UserDatabase {
```

```
  // Índices
  email: string;
  dni: string;
  publicAddress: string;

  // Encriptado
  encryptedDNI: string;
  encryptedPublicAddress: string;
  encryptedPrivateKey: string;

  // Metadata
  createdAt: Date;
  recyclingStats?: RecyclingStats;
}

interface NFCData {
  publicAddress: string;
  encryptedDNI: string;
  encryptedPrivateKey: string;
}
```

Funciones principales

typescript

```
// Registro
createUser(email, dni, password): UserDatabase & NFCData
```

```
    // Autenticación
    authenticateByEmail(email, password): boolean
    authenticateByDNI(dni, password): boolean
    authenticateByNFC(nfcData, password): boolean

    // Operaciones
    decryptPrivateKey(userId, password): string
recordRecycling(publicAddress, materials): Transaction
```

❓ FAQ

¿Por qué algunos datos están duplicados (encriptados y sin encriptar)? Los datos sin encriptar sirven como índices de búsqueda rápida. Los encriptados sirven para verificar el password y proteger información sensible.

¿Qué pasa si alguien ve mi DNI o email en la base de datos? Son datos de identificación, no permiten acceder a tu wallet ni firmar transacciones sin tu password.

¿Puedo reciclar sin llevar mi NFC? No, necesitas el NFC para que el sistema registre el reciclaje en tu dirección. Alternativamente podrías usar DNI + Password en un terminal.

¿Qué pasa si pierdo el NFC? Tu cuenta sigue existiendo en la base de datos. Puedes seguir accediendo con Email/DNI + Password. Se puede generar un nuevo NFC con los mismos datos.

¿Puedo cambiar la contraseña? Sí. El sistema desencripta todos tus datos con la contraseña antigua, los reencripta con la nueva, y actualiza la BD y el NFC.

¿Es compatible con cualquier NFC? Sí, chips NFC estándar NTAG215/216, MIFARE Classic, o cualquiera con 512+ bytes de memoria.

¿Necesito internet para reciclar? No inmediatamente. El sistema puede funcionar offline y sincronizar después. Para consultar saldo o enviar fondos sí necesitas internet.