Gestión del Portador

Esta sección describe los procesos técnicos que ocurren dentro de la aplicación `Soy Yo RD` **(la wallet)**, controlada por el ciudadano (el Portador o *Holder*). Cubre desde la creación de la identidad digital hasta la presentación y gestión de la licencia de conducir verificable.

Caso de Uso 1: Onboarding y Creación de la Identidad Digital

Este es el flujo inicial cuando un ciudadano instala y configura la wallet `Soy Yo RD` por primera vez.

1. **Acción del Usuario**: El ciudadano instala y abre la app `Soy Yo RD`.
2. **Proceso Interno (Generación de Claves)**: La wallet invoca las APIs del sistema operativo para generar un par de claves criptográficas asimétricas directamente en el hardware de almacenamiento seguro.
   - **Mecanismo**: Se utiliza un algoritmo de firma eficiente para móviles, como **Ed25519** .
   - **Resultado**: Se obtiene una `clave privada` (que nunca abandona el almacenamiento seguro) y una `clave pública` correspondiente.
3. **Proceso Interno (Creación del DID)**: La wallet utiliza la clave pública recién generada para crear un Identificador Descentralizado (DID) utilizando un método soportado, como `did:key`.
   - **Formato del DID (`did:key`)**: `did:key:z6Mkt...H73tias` (donde la cadena después de `z6Mk` es una representación en Base58-BTC de la clave pública).
   - **Documento DID (Generado localmente)**: La wallet construye en memoria el Documento DID, que asocia el DID con su clave pública y define los métodos de verificación. Este documento no necesita ser publicado en una red para `did:key`.
4. **Proceso Interno (Creación del Respaldo de Seguridad)**: La wallet genera una frase mnemónica de 12 o 24 palabras **(BIP39)** a partir de la entropía utilizada para crear la clave privada maestra.
   - **Acción del Usuario**: Se le solicita al ciudadano que escriba y guarde esta frase en un lugar físico y seguro. Se le advierte que es la única forma de recuperar su identidad si pierde el dispositivo. La app no guarda esta frase.

Post-condición: El ciudadano tiene una identidad digital auto-soberana, representada por un DID y controlada por una clave privada segura en su dispositivo.

Caso de Uso 2: Almacenamiento y Visualización Segura de la Credencial

Una vez que la credencial ha sido emitida por el INTRANT (como se describió en la documentación anterior) y recibida por la wallet:

1. **Recepción y Verificación Inicial**:
   - **Mecanismo**: La wallet recibe la Credencial Verificable (VC) en formato **JSON-LD** a través de la respuesta de la API de emisión.
   - **Proceso Interno**: Antes de guardarla, la wallet realiza una verificación inmediata:
     1. Resuelve el DID del emisor (`issuer`: `did:web:intrant.gob.do`). Esto implica una llamada **API REST (GET)** a `https://intrant.gob.do/.well-known/did.json` para obtener el Documento DID del INTRANT y su clave pública.
     2. Usa la clave pública del INTRANT para verificar la firma (`proofValue`) de la VC.
     3. Si la firma es válida, procede a guardar. Si no, notifica al usuario de un error.
2. **Almacenamiento Seguro**:
   - **Mecanismo**: La wallet almacena la VC (el archivo JSON-LD) en una base de datos local cifrada (ej. SQLite con SQLCipher). La clave de cifrado de esta base de datos está protegida y gestionada por el Almacenamiento Seguro del dispositivo.
   - **Resultado**: Las credenciales no se pueden leer ni extraer si el dispositivo es comprometido o si la aplicación es accedida sin la autorización del usuario (biometría/PIN).
3. **Visualización**:
   - **Acción del Usuario**: El ciudadano navega a su sección de credenciales y toca la licencia.
   - **Proceso Interno**: La app lee el archivo JSON-LD de la base de datos cifrada, lo decodifica y renderiza los campos (`credentialSubject`) en una interfaz de usuario amigable y visualmente representativa de una licencia de conducir.

Caso de Uso 3: Creación y Presentación para Verificación

Este es el flujo técnico para presentar la licencia a un agente de la DIGESETT.

1. **Inicio del Intercambio (Solicitud de Presentación)**:
   - **Acción del Usuario**: El ciudadano presiona "Presentar Licencia" y la cámara de `Soy Yo RD` se activa para escanear un código QR mostrado por el agente verificador.
   - **Formato del QR**: El código QR no contiene datos personales, sino una solicitud de presentación (Presentation Request). Esta solicitud puede seguir el estándar **W3C Presentation Exchange** o estar encapsulada en un protocolo como **OpenID for**

Verifiable Presentations (OID4VP).
Ejemplo de Payload del QR (OID4VP - URL):

```
openid-vc://?request_uri=https://api.digesett.gob.do/request/12345
```

2. **Procesamiento de la Solicitud**:
   - **Mecanismo**: La wallet `Soy Yo RD` realiza una **API REST (GET)** a la `request_uri` obtenida del QR.
   - **Respuesta de la API (JSON)**: El servidor del verificador responde con un objeto JSON que define qué credenciales se solicitan.
     Ejemplo de Presentation Definition (JSON):

```
{
  "id": "definicion_licencia_1",
  "input_descriptors": [
    {
      "id": "licencia_rd_descriptor",
      "name": "Licencia de Conducir Dominicana",
      "schema": [{ "uri": "https://w3id.org/vdl/v1" }],
      "constraints": {
        "fields": [
          {
            "path": ["$.type"],
            "filter": { "type": "string", "const": "Iso18013DriversLicense" }
          },
          {
            "path": ["$.issuer"],
            "filter": { "type": "string", "pattern": "^did:web:intrant.gob.do$"
}
          }
        ]
      }
    }
  ]
}
```

     Este JSON le dice a la wallet: "Necesito una credencial que sea del tipo 'Iso18013DriversLicense' y que haya sido emitida por 'did:web:intrant.gob.do'".
3. **Construcción de la Presentación Verificable (VP)**:
   - **Acción del Usuario**: La wallet encuentra la credencial que coincide con la solicitud y le pide al usuario que autorice compartirla (usando biometría o PIN).
   - **Proceso Interno**: Tras la autorización, la wallet construye una **Presentación Verificable (VP)**.

Formato de la VP (JSON-LD):

```
{
  "@context": ["https://www.w3.org/2018/credentials/v1"],
  "type": ["VerifiablePresentation"],
  "verifiableCredential": [
    // Aquí se inserta la VC completa de la licencia de conducir
    { "@context": [...], "type": [...], "issuer": "did:web:intrant.gob.do", ...
}
  ],
  "holder": "did:key:z6Mkt...H73tias", // El DID del ciudadano
  "proof": {
    "type": "Ed25519Signature2020",
    "created": "2025-06-17T19:30:00Z",
    "verificationMethod": "did:key:z6Mkt...H73tias#z6Mkt...H73tias",
    "proofPurpose": "authentication",
    "challenge": "CHALLENGE_STRING_DE_LA_SOLICITUD", // Previene ataques de
repetición
    "proofValue": "z3aD8...k7gRt" // Firma del ciudadano sobre toda la VP
  }
}
```

- **Punto Clave**: La VP es firmada por el **ciudadano** (`holder`) usando su clave privada. Esto prueba dos cosas: que la credencial original no ha sido alterada (por la firma del INTRANT) y que el portador actual controla el DID del sujeto de la credencial (por la firma del ciudadano).

4. **Envío de la VP al Verificador**:
   - **Mecanismo**: La wallet envía la VP completa al servidor del verificador. Esto se puede hacer a través de una **API REST (POST)** al endpoint que se especificó en el protocolo de intercambio o directamente vía Bluetooth/NFC si la conexión es de proximidad.
   - **Endpoint de envío (ejemplo):** `https://api.digesett.gob.do/submit`
   - **Body (JSON)**: El objeto VP completo.

Caso de Uso 4: Gestión de la Wallet y Recuperación

1. **Respaldo (Backup)**:
   - **Mecanismo**: Como se describió en el onboarding, el único método de respaldo es la **frase mnemónica (BIP39)**. Es responsabilidad del ciudadano guardarla. No hay API

ni comunicación externa; es un proceso local.

2. **Recuperación (Restore)**:
   - **Acción del Usuario**: En un teléfono nuevo, el ciudadano instala `Soy Yo RD` y elige "Recuperar Wallet".
   - **Proceso Interno**:
     1. La aplicación le pide al usuario que ingrese su frase mnemónica de 12/24 palabras.
     2. Usando el algoritmo BIP39, la wallet re-genera la clave privada maestra original a partir de la frase.
     3. A partir de la clave maestra, deriva exactamente el mismo par de claves (privada/pública) y, por lo tanto, el mismo `did:key` que tenía en el dispositivo anterior.
   - **Recuperación de Credenciales**: La identidad está restaurada, pero las credenciales no. La wallet está vacía. El ciudadano debe solicitar la re-emisión de cada credencial (como la licencia de conducir) a sus respectivos emisores, repitiendo el flujo de emisión original. El emisor lo tratará como una nueva solicitud de un DID ya conocido.

Esta arquitectura centrada en el portador le da al ciudadano el control total de su identidad y sus datos, utilizando criptografía estándar y protocolos de comunicación abiertos para interactuar de forma segura con emisores y verificadores.

Revisión #4
Creado 2025-06-17 18:54:29 UTC por Tomás Familia
Actualizado 2025-06-17 19:04:08 UTC por Tomás Familia