Verificación de la Credencial

Esta sección describe los flujos técnicos y los mecanismos de comunicación para la verificación de la Licencia de Conducir Digital. El proceso está diseñado para ser seguro, rápido y funcional tanto en escenarios con conexión a internet como sin ella.

Protocolo de Comunicación Principal: OID4VP

Para una máxima interoperabilidad y seguridad, el protocolo recomendado es **OpenID for Verifiable Presentations (OID4VP)**. Este estándar define cómo una aplicación de verificador (Relying Party) solicita una Presentación Verificable (VP) de una wallet (Self-Issued OpenID Provider).

Caso de Uso 1: Verificación Estándar en Línea (Online)

Este es el escenario más común, donde el dispositivo del verificador tiene conexión a internet.

1. **Inicio del Flujo (Generación de la Solicitud)**
   - **Acción del Verificador**: El agente abre su aplicación de verificación y presiona "Escanear Licencia".
   - **Mecanismo Técnico**: La aplicación del verificador genera un código QR único para esta transacción específica. El QR contiene una URL que sigue el esquema OID4VP.
     Ejemplo de contenido del QR (URL):

     ```
     openid-vc://?request_uri=https://api.verificador.digesett.gob.do/vp-request/a7b3c9d1-e2f4-4a5b-8c6d-9e0f1a2b3c4d
     ```

2. **Solicitud de Presentación por parte de la Wallet**
   - **Acción del Portador**: El ciudadano utiliza su wallet `Soy Yo RD` para escanear el código QR.
   - **Llamada API (GET)**: La wallet extrae la `request_uri` y realiza una llamada **HTTP GET** a ese endpoint.
     - **Endpoint**: `https://api.verificador.digesett.gob.do/vp-request/a7b3c9d1-e2f4-4a5b-8c6d-9e0f1a2b3c4d`

- **Respuesta de la API (JSON)**: El servidor del verificador responde con una **Solicitud de Presentación (Presentation Request)** en formato JSON. Esta solicitud contiene una **Definición de Presentación (Presentation Definition)** que especifica los requisitos de la credencial.
  Ejemplo de la Respuesta (Presentation Definition):

```
{
 "response_type": "vp_token",
 "client_id": "digesett_verifier_app_01",
 "presentation_definition": {
  "id": "licencia_rd_pd_1",
  "input_descriptors": [{
   "id": "licencia_descriptor",
   "name": "Licencia de Conducir de la República Dominicana",
   "schema": [{ "uri": "https://w3id.org/vdl/v1" }],
   "constraints": {
    "fields": [
     {
      "path": ["$.type"],
      "filter": { "type": "string", "contains": "Iso18013DriversLicense" }
     },
     {
      "path": ["$.issuer"],
      "filter": { "type": "string", "pattern": "^did:web:intrant.gob.do$" }
     }
    ]
   }
  }]
 },
 "nonce": "n-0S6_WzA2Mj" // Nonce para prevenir ataques de repetición
}
```

3. **Construcción y Envío de la Presentación**
   - **Proceso Interno de la Wallet**:
     1. La wallet analiza la `presentation_definition` y busca en su almacenamiento una VC que cumpla con los `constraints`.
     2. Encuentra la licencia del INTRANT, la selecciona y pide autorización al ciudadano (PIN/biometría).
     3. Construye una **Presentación Verificable (VP)**, firmándola con la clave privada del portador e incluyendo el `nonce` de la solicitud para seguridad.
   - **Llamada API (POST)**: La wallet envía la VP al servidor del verificador. El endpoint para el envío suele ser parte de la configuración del protocolo OID4VP o se define en

la solicitud.
    - **Endpoint**: `https://api.verificador.digesett.gob.do/vp-response`
    - **Body (Form-urlencoded)**:

```
vp_token=<LA_VP_COMPLETA_EN_FORMATO_JWT_O_JSON-LD>
&presentation_submission=<DESCRIPTOR_MAP_JSON>
```

4. **Proceso de Verificación Criptográfica (Backend del Verificador)** Al recibir la VP, el sistema del verificador ejecuta una serie de comprobaciones automáticas en milisegundos:
    - **Paso 4.1: Verificar la Firma del Portador (Holder)**
        - **Objetivo**: Confirmar que la presentación fue autorizada por el legítimo propietario de la wallet.
        - **Mecanismo**: Extrae el DID del portador (`holder` en la VP), obtiene su clave pública (del campo `verificationMethod` si es `did:key`, o resolviéndolo) y verifica la firma externa de la VP. Se comprueba también que el `nonce` coincide con el enviado en la solicitud.
    - **Paso 4.2: Verificar la Firma del Emisor (Issuer)**
        - **Objetivo**: Confirmar que la licencia es auténtica, no ha sido alterada y fue emitida por el INTRANT.
        - **Mecanismo**: Extrae la VC de la VP. Resuelve el DID del emisor (`issuer`: `did:web:intrant.gob.do`) realizando una llamada **GET** a `https://intrant.gob.do/.well-known/did.json`. Usa la clave pública obtenida para validar la firma interna de la VC.
    - **Paso 4.3: Verificar el Estado de la Credencial (Revocación)**
        - **Objetivo**: Asegurarse de que la licencia no ha sido revocada por el INTRANT.
        - **Mecanismo**:
            1. Lee el bloque `credentialStatus` dentro de la VC.
            2. Realiza una llamada **HTTP GET** a la URL indicada (ej. `https://api.intrant.gob.do/status/1`).
            3. Recibe una lista de estado (ej. StatusList2021) que contiene un bitmap.
            4. Comprueba el bit en la posición correspondiente al `statusPurpose` y `statusListIndex` de la credencial. Si el bit es `1`, la credencial está revocada.
5. **Resultado Final**
    - La aplicación del verificador recibe una respuesta final de su backend y muestra un resultado claro al agente:
        - **VERDE (VÁLIDO)**: Muestra la foto, nombre y datos de la licencia.
        - **ROJO (INVÁLIDO)**: Muestra el motivo del fallo (ej. "Firma del emisor inválida", "Licencia revocada", "Licencia expirada").

Caso de Uso 2: Verificación sin Conexión (Offline)

Este escenario es crucial para operaciones en áreas rurales o con mala cobertura de red.

1. **Inicio y Solicitud**: El flujo es similar, pero el intercambio de datos debe ser directo entre dispositivos (Peer-to-Peer).
   - **Mecanismo de Intercambio**: En lugar de una `request_uri`, el QR puede contener la **solicitud de presentación completa** en formato JSON. La comunicación entre dispositivos se establece usando **Bluetooth Low Energy (BLE)**, **NFC** o **Wi-Fi Direct**. La especificación **OpenID Connect for Verifiable Presentations (SIOPv2)** detalla estos flujos P2P.
2. **Proceso de Verificación (en la App del Verificador)** La aplicación del verificador debe poder realizar las comprobaciones criptográficas sin conexión a internet.
   - **Paso 2.1: Verificar Firma del Portador (POSIBLE OFFLINE)**
     - Esta verificación es puramente matemática y no requiere red. Se puede completar siempre.
   - **Paso 2.2: Verificar Firma del Emisor (POSIBLE OFFLINE)**
     - Para que esto funcione offline, la aplicación del verificador debe tener **pre-cargada (en caché) la clave pública del INTRANT**. Se puede actualizar periódicamente cuando la app tiene conexión. Así, no necesita resolver el `did:web` en tiempo real.
   - **Paso 2.3: Verificar Estado de Revocación (NO ES POSIBLE OFFLINE)**
     - La comprobación del estado en tiempo real contra una lista en un servidor es imposible sin conexión.
3. **Resultado Final (Offline)** La aplicación del verificador debe presentar un resultado diferenciado:
   - **AMARILLO (PARCIALMENTE VERIFICADO)**: Muestra los datos de la licencia (foto, nombre) y un aviso claro: **"Autenticidad Verificada. Estado de Revocación no pudo ser comprobado. Última sincronización: [Fecha/Hora]"**.
   - **ROJO (INVÁLIDO)**: Si la firma del portador o del emisor falla, el resultado es inválido independientemente del estado de conexión.

Caso de Uso 3: Verificación con Divulgación Selectiva

Un verificador (ej. un bar) solo necesita saber si una persona es mayor de 18 años, no su dirección o nombre completo.

1. **Solicitud de Presentación Específica**: La `presentation_definition` generada por el verificador solicitará una "prueba de edad derivada" en lugar de la credencial completa.
   - **Mecanismo**: Utilizando técnicas como **Zero-Knowledge Proofs (ZKP)** o solicitando una credencial derivada firmada por el portador.
   - **Ejemplo de `presentation_definition` con predicados (si el estándar lo soporta):**

```
"constraints": {
  "fields": [{
```

```
    "path": ["$.credentialSubject.birthDate"],
    "purpose": "Necesitamos verificar que eres mayor de 18 años.",
    "predicate": { // Indica que se necesita una prueba, no el valor
      "type": "date",
      "operator": "<=",
      "value": "2007-06-17" // Fecha de hoy hace 18 años
    }
  }]
}
```

2. **Respuesta de la Wallet**: La wallet `Soy Yo RD` interpreta esta solicitud, realiza el cálculo localmente y genera una presentación que solo afirma: **"El campo `birthDate` en la credencial emitida por el INTRANT es anterior o igual a 2007-06-17: VERDADERO"**. Esta presentación está firmada por el portador y por el emisor (indirectamente a través de la VC original), sin revelar la fecha de nacimiento exacta.
3. **Resultado**: El verificador solo recibe un "Sí" o "No" a la pregunta que hizo, protegiendo al máximo la privacidad del ciudadano.

Revisión #3
Creado 17 junio 2025 18:54:44 por Tomás Familia
Actualizado 17 junio 2025 19:13:07 por Tomás Familia