¿Qué es una credencial verificable?

Una **credencial verificable** (en inglés *Verifiable Credential* o *VC*) es una representación digital de información que afirma algo sobre una persona, organización o cosa, y que puede ser **verificada criptográficamente** para confirmar su autenticidad y validez.

Las credenciales verificables están basadas en estándares definidos por el **W3C** y permiten emitir, presentar y validar datos de forma segura, privada y descentralizada.

¡Con gusto, Tomás! Aquí tienes una definición clara y técnica de una **credencial verificable**, ideal para incluir en tu documentación:

Ecosistema de una credencial verificable

Las credenciales verificables se basan en un ecosistema compuesto por entidades que desempeñan diferentes **"roles"**. Los roles principales son:

Emisor

Una entidad que crea una Credencial Verificable, compuesta por una serie de afirmaciones relacionadas con su sujeto. Un ejemplo es una universidad que emite credenciales de títulos universitarios o certificados para sus egresados.

Titular

Una entidad que posee una o más Credenciales, y que puede transmitir presentaciones de esas Credenciales Verificables a terceros. Un ejemplo puede ser la persona que "posee" sus propios títulos educativos. Otro ejemplo puede ser una billetera digital que contiene varias credenciales en nombre de alguien.

Verificador

Una entidad que realiza la verificación de una Credencial Verificable para comprobar su validez, consistencia, etc. Un ejemplo puede ser el sistema digital de un empleador que verifica la validez de un título universitario antes de decidir contratar a una persona.

Estructura

Las credenciales verificables siguen un formato estandarizado, normalmente en **JSON-LD**, que facilita la interoperabilidad entre sistemas.

JSON-LD (*JavaScript Object Notation for Linked Data*) es una extensión del formato JSON que permite representar datos con significado semántico, utilizando contextos (`@context`) que definen el significado de cada campo. Esto facilita la interoperabilidad entre sistemas al proporcionar una forma estándar de interpretar la información.

Componentes de una Credencial Verificable

A continuación, se detallan los elementos clave de una credencial verificable, con sus funciones y ejemplos:

`@context`

Este campo define el significado semántico de los términos utilizados en la credencial. Utiliza vocabularios establecidos y permite que los sistemas que reciben la credencial puedan interpretar correctamente su contenido, incluso si no conocen previamente su estructura exacta.

Ejemplo:

```
"@context": [
  "https://www.w3.org/2018/credentials/v1",
  "https://gob.do/contexts/driver-license"
]
```

`type`

El campo type define el tipo de credencial. Toda credencial verificable debe incluir `"VerifiableCredential"` como tipo base, y puede agregar tipos adicionales que indiquen su propósito específico, como `"DriverLicenseCredential"`, `"UniversityDegreeCredential"`, etc.

Ejemplo:

```
"type": ["VerifiableCredential", "DriverLicenseCredential"]
```

issuer

Es el identificador del emisor, es decir, la entidad que firma digitalmente la credencial. Suele representarse mediante un **DID (Decentralized Identifier)** o una URL que puede resolverse para obtener la clave pública del emisor. El verificador necesita este valor para buscar la clave pública del emisor y validar la firma de la credencial. También permite a los sistemas confiar en la fuente, ya que sabrán si la credencial proviene de una entidad oficial.

Ejemplo:

```
"issuer": "https://gob.do/dgeem"
```

issuanceDate

Es la fecha de emisión de la credencial. Debe expresarse en formato ISO 8601, y permite controlar la vigencia de la credencial en combinación con un campo opcional de expiración (`expirationDate` dentro del `credentialSubject`).

Ejemplo:

```
"issuanceDate": "2025-05-26T00:00:00Z"
```

credentialSubject

Contiene la información que se afirma sobre la entidad (persona, empresa, objeto, etc.). Es el contenido central de la credencial. Puede incluir campos como nombre, identificación, rol, atributos, permisos, etc. Siempre debe incluir un campo `id`, que representa al sujeto(normalmente con un DID). Dependiendo del caso de suo, puede contener información personal u organizacional.

Ejemplo:

```
"credentialSubject": {
  "id": "did:example:juanperez",
  "name": "Juan Pérez",
  "licenseNumber": "A1234567",
  "category": "Private vehicle",
  "expirationDate": "2030-05-26"
}
```

proof

El campo `proof` es lo que **convierte un simple documento JSON en una credencial verificable**, ya que contiene la **firma digital** que garantiza que el contenido no ha sido alterado y que fue emitido por una entidad confiable.

Ejemplo:

```
"proof": {
  "type": "Ed25519Signature2018",
  "created": "2025-05-26T00:00:00Z",
  "proofPurpose": "assertionMethod",
  "verificationMethod": "https://gob.do/keys/clave-publica.json",
  "jws": "eyJhbGciOiJFZERTQSJ9...firma..."
}
```

El `proof` se compone de tales componentes:

- `type`: especifica el **algoritmo de firma** o el método criptográfico que se usó para generar la firma de la credencial. Algoritmos que se usan mucho en estos son `"Ed25519Signature2018"`, `"RsaSignature2018"` o `"BbsBlsSignature2020"`.
- `created`: Es la **fecha y hora exacta** en la que se generó la firma digital. Debe estar en formato ISO 8601 (ej. `"2025-05-26T14:23:12Z"`).
- `proofPurpose`: Define **el propósito de la firma**, es decir, **qué intención tiene el emisor al firmar esta credencial**.Los valores típicos para este campo pueden ser:
  - `"assertionMethod"` – el emisor afirma que el contenido es verdadero (el más común en VC).
  - `"authentication"` – usado para firmar pruebas de que alguien es quien dice ser (ej. en autenticación con DIDs).
  - `"capabilityDelegation"` / `"capabilityInvocation"` – usados en sistemas de control de acceso.
- `verificationMethod`: Es un **identificador (generalmente una URL o DID)** que apunta a la **clave pública** que se debe usar para verificar la firma.
- `jws`: Contiene la **firma digital** como una cadena codificada en formato **JWS (JSON Web Signature)**, que es parte del estándar de JOSE (JSON Object Signing and Encryption). Se representa como una cadena Base64 que incluye: un encabezado (con algoritmo y tipo), el contenido firmado (payload), La firma propiamente dicha.

Revisión #5
Creado 27 mayo 2025 14:53:46 por Tomás Familia
Actualizado 12 junio 2025 19:49:34 por Tomás Familia