Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

MENCIÓN EN INGENIERÍA DE SOFTWARE

Modelo de Identidad Digital Autosoberana basado en Credenciales Verificables y *Blockchain*

Alumno:
Alejandro Nieto Gallego

Tutor:
Diego García Álvarez

Dedicado a todas las personas con las que me he cruzado hasta este momento en mi vida ya que todas y cada una de ellas han influido en mi evolución hasta llegar a este fin de una etapa que marca la entrega de este TFG. En especial a mi familia, tanto a los que siguen presentes como a los que ya no pueden ver el fruto de mi esfuerzo y de su apoyo incondicional.

Agradecimientos

Quiero agradecer a todas las personas que han estado a mi lado durante la elaboración de este Trabajo de Fin de Grado. En especial a mi compañero y responsable en la empresa en la que trabajo desde mis prácticas curriculares, Alejandro Alfonso Fernández, quien me dio la idea de enfocar mi trabajo relacionado con la tecnología Blockchain hacia el ámbito de la Identidad Digital Autosoberana y me ha proporcionado información valiosa sobre el estado del arte en este ámbito. También a mi tutor, Diego García Álvarez, que ha hecho un seguimiento constante de mi trabajo y me ha proporcionado *feedback* constante.

Resumen

En este trabajo se busca alcanzar una solución que mejore la privacidad del usuario modificando la forma de intercambiar sus datos personales con las entidades que los requieren para proveer de ciertos servicios a los usuarios.

El objetivo principal es resolver el problema actual de trazabilidad de la actividad de los usuarios, conocido como su *huella digital*, es decir, donde se registran, en que sitios web se han identificado, cuando etc por parte de terceros que no sean el proveedor de servicios al que el usuario debe consentir si o si sus datos para poder proveerle de servicios.

Este tipo de situaciones tienen lugar si se utilizan algunas de las soluciones *Single Sign On* de alguna de las grandes tecnológicas, como Facebook, Google o Twitter. El Single Sign On consiste en permitir registrarse e iniciar sesión en un servicio con la cuenta de otro, generalmente con la cuenta de una de las grandes tecnológicas citadas anteriormente. Esto hace que las grandes tecnológicas puedan saber qué otros servicios están utilizando sus usuarios. Además, debido a que los proveedores de servicios no siempre piden los datos que exclusivamente necesitan o no especifican de manera clara y precisa los datos a los que van a acceder hace que acrecente el problema, haciendo que el usuario decida aceptar sin saber realmente qué datos va a ceder y con qué propósitos.

Para solventar esta problemática se va a modelar una solución de Single Sign On con las mismas ventajas que las proporcionados por las grandes tecnológicas pero que evite la trazabilidad de la actividad de los usuarios por parte de estas compañías y que permitirá también conocer qué datos se están solicitando exactamente.

Para ello se presenta un modelo teórico y se incluirá el análisis y diseño de su aplicación en un caso concreto.

Una de las partes clave para conseguir este propósito es el uso de la tecnología *Blockchain* para establecer la confianza y vigencia de los datos intercambiados e incluso para almacenar las evidencias de revocación de la cesión de datos por parte de los propios usuarios.

Esta tecnología permite no tener que contactar con la plataforma dueña de los datos del usuario, que sería una de las grandes tecnológicas, para que esta dé fé de la veracidad de los datos, ya que mantiene la inalterabilidad del estado de validez de los mismos que esta deje almacenada en la *blockchain*.

Abstract

This work seeks to achieve a solution that improves user privacy by modifying the way users exchange their personal data with entities that require it to provide certain services.

The main objective is to address the current problem of traceability of user activity, known as their digital footprint, that is, where, when, and on which websites they have authenticated, as recorded by third parties other than the service provider to whom the user must necessarily consent to share their data in order to receive services.

Such situations occur when using some Single Sign On solutions offered by major tech companies like Facebook, Google, or Twitter. Single Sign On allows users to register and log in to a service using an account from another provider, typically of these major tech companies. This enables these companies to know which other services their users are accessing. Furthermore, since service providers do not always request only the data they strictly need or fail to clearly and precisely specify the data they will access, the problem is exacerbated, leading users to accept without truly knowing what data they are sharing or for what purposes.

To solve this issue, an alternative Single Sign On solution will be modeled, offering the same advantages as those provided by the major tech companies, but preventing these companies from tracing users' activities while also allowing users to know exactly what data is being requested.

To this end, a theoretical model is presented, including the analysis and design of its application in a specific case.

A key part of achieving this goal is the use of Blockchain technology to establish the trust and validity of the exchanged data, and even to store evidence of the revocation of data sharing by the users themselves.

This technology makes it unnecessary to contact the platform owning the user's data (which would typically be one

of the major tech companies) to attest to the authenticity of the data, as the blockchain ensures the immutability of the validity state of the data once stored on the blockchain by that same company.

Índice general

Índice de figuras

Índice de tablas

Glosario

A | C | D | E | F | G | H | I | J | L | O | P | Q | R | S | T | U | V | W

A

AdES

Advanced Electronic Signatures se refiere a un conjunto de formatos de firma electrónica que cumplen con los requisitos del reglamento *eIDAS*, proporcionando mayor validez legal y garantías técnicas Incluyen *XAdES*, *CAdES*, *PAdES*, asegurando integridad, autenticidad y verificabilidad a largo plazo de documentos firmados electrónicamente. XXX

ARF

Architecture and Reference Framework (Marco de Arquitectura y Referencia) Modelo estructurado que define los componentes y estándares en la arquitectura formulada para dar soporte a la normativa *eIDAS 2*. XXX

Authentication Request

Authentication Request (Petición de Autenticación) Petición *HTTP* en *OIDC* o *SIOPv2* para solicitar la autenticación del usuario mediante un tercero o *SIOP*. XXX

Authentication Response

Authentication Response (Respuesta de Autenticación) Petición *HTTP* en *OIDC* o *SIOPv2* que entrega un *ID Token* tras la autenticación, usado luego para acceder a recursos protegidos o solicitar más información, y opcionalmente incluir *Verifiable Presentations* vía *OIDC4VP*. XXX

C

Credential Offer

Credential Offer (Oferta de Credencial) Petición HTTP en *OIDC4VCI* que informa al usuario sobre las características de la credencial a emitir y provee datos para obtener metadatos del *Issuer*. XXX

Credential Request

Credential Request (Petición de Credencial) Petición HTTP en *OIDC4VCI* para solicitar la emisión de credenciales, proporcionando el *access_token* y una *proof of possession*. XXX

Credential Response

Credential Response (Respuesta de Credencial) Petición HTTP en *OIDC4VCI* que entrega las credenciales emitidas a la *Wallet*. XXX

D

DID

Decentralized Identifier (Identificador Descentralizado) Identificador único y distribuido cuya emisión, uso y revocación sigue una política de gobernanza participativa sin depender de una entidad central. XXX

DLT

Distributed Ledger Technology (Tecnología de Registro Distribuido) Tecnología donde los datos se replican, comparten y sincronizan entre nodos. XXX

E

End-User

End-User (Usuario Final) Usuario que interactúa con *Issuer* y *Verifier* y también usuario de una *Wallet*. XXX

EUDI Wallet

European Union Digital Identity Wallet (Cartera de Identidad Digital Europea) Cartera digital de la UE para almacenar y gestionar identidades y credenciales de forma segura. XXX

F

Fuente de verdad

La *Fuente de Verdad* de un dato es la entidad que lo genera y que puede afirmar y demostrar su veracidad si es necesario. XXX

G

GDPR

General Data Protection Regulation (Reglamento General de Protección de Datos) Reglamento de la UE sobre tratamiento y protección de datos personales. XXX

H

Holder

Holder (Poseedor) Entidad que presenta una *VP* con aserciones sobre uno o más *Subjects*. XXX

I

ID Token

ID Token Token definido en *OpenID Connect* y *SIOPv2* usado por el usuario para acceder a recursos gestionados por un *RP* tras autenticarse o autorizarse. XXX

IdP

Identity Provider (Proveedor de Identidad) Entidad que autentica a los usuarios y emite afirmaciones de identidad para otros servicios. XXX

Issuer

Issuer (Emisor) Entidad que emite aserciones sobre un *Subject* según *W3C Verifiable Credentials* Debe garantizar que dichas aserciones puedan verificarse y ser confiables por parte de un *Verifier*. XXX

J

JWS

JSON Web Signature Especificación que define cómo firmar digitalmente contenido *JSON*, generando *tokens* protegidos. XXX

JWT

JSON Web Token Formato compacto y seguro para transmitir afirmaciones entre partes, firmado digitalmente y opcionalmente cifrado. XXX

L

LoA

Level Of Assurance se refiere al nivel de confianza en la identidad de un *Subject* al realizar una firma electrónica Clasificado en bajo, medio, sustancial y alto. XXX

O

OAuth2

OAuth2 Protocolo de autorización para acceder a recursos protegidos mediante *tokens*. XXX

OIDC4VCI

OpenID Connect for Verifiable Credentials Issuance (*OpenID Connect* para Emisión de Credenciales Verificables) Protocolo para emitir credenciales verificables usando *OpenID Connect*. XXX

OIDC4VP

OpenID Connect for Verifiable Presentations (*OpenID Connect* para Presentaciones Verificables) Protocolo para presentar credenciales verificables usando *OpenID Connect*. XXX

OP

OpenID Provider (Proveedor *OpenID*) Sistema operado por un tercero que emite información sobre el usuario a un *RP* por solicitud del usuario para acceder a un servicio. XXX

P

PID

Personal Identification Data (Datos Personales de Identificación) Conjunto de datos personales que identifican de manera única a una persona. XXX

Presentation Definition

Presentation Definition (Definición de Presentación) Descripción de las credenciales requeridas mediante una presentación generada por el usuario. XXX

Proof of Possession

Proof of Possession (Prueba de Posesión) Demostración de que se posee determinada clave o material criptográfico asociado a una credencial. XXX

Protocolo de autenticación

Protocolo que define la verificación de un usuario para interactuar con un sistema determinado. XXX

Protocolo de autorización

Protocolo que define cómo determinar que un usuario está autorizado a acceder a un sistema determinado. XXX

Protocolo de consenso

Protocolo que establece la legitimidad en una red *Blockchain* ya sea para añadir un bloque u otras decisiones sobre su funcionamiento. XXX

Q

QEAA

Qualified Electronic Attestation of Attributes (Acreditación Electrónica de Atributos Cualificada) Acreditación emitida por un *QTSP* que certifica atributos con validez legal bajo la futura *eIDAS 2*. XXX

QES

Qualified Electronic Signature (Firma Electrónica Cualificada) Firma que cumple el nivel más alto de garantías legales bajo *eIDAS*, basada en un certificado cualificado. XXX

QTSP

Qualified Trust Service Provider (Proveedor de Servicios de Confianza Cualificado) Proveedor con la condición de cualificado según el reglamento *eIDAS*. XXX

R

Relying Party

Relying Party (Parte Confiante) Parte que debe confiar, previa verificación, en la información proporcionada por el usuario o un tercero, como un proveedor de servicios en *OAuth2*. XXX

S

Selective Disclosure

Selective Disclosure (Divulgación Selectiva) Técnica que permite extraer solo las aserciones necesarias sobre una credencial, manteniendo integridad y seguridad. XXX

Service Provider

Service Provider (Proveedor de Servicios) Entidad que interactúa con un usuario para proveer un servicio. XXX

SIOP

Self Issued OpenID Provider (Proveedor *OpenID* Autogestionado) Sistema operado por el usuario que emite aserciones verificables sobre sí mismo a un *RP*. XXX

SIOPv2

Self Issued OpenID Connect v2 Flujo de *OpenID Connect* donde el usuario actúa como su propio proveedor de identidad. XXX

SSI

Self Sovereign Identity (Identidad Autosoberana) Modelo de identidad descentralizada donde las personas controlan su identidad digital sin autoridades centrales que den fé acerca de su identificación. XXX

SSO

Single Sign On (Inicio de Sesión Único) Autenticación que permite acceder a varios servicios con una sola cuenta de usuario registrada. XXX

Subject

Subject (Sujeto) Entidad sobre la que se emiten aserciones No necesariamente es el *Holder* de las *VC* asociadas Puede ser una persona, cosa u otra entidad. XXX

Subject

Subject (Sujeto) Entidad (persona u organización) a la que se refieren las credenciales o atributos. XXX

T

TL

Trusted List (Lista de Confianza) Lista oficial de la UE que incluye información sobre proveedores de servicios de confianza supervisados y cualificados. XXX

Token Request

Token Request (Petición de Token) Petición HTTP en *OIDC4VCI* para solicitar un *token* que permita a la *Wallet* autenticarse ante el *Issuer* para la emisión de credenciales, usando *Pre-Authorized Code* y *PIN* si aplica. XXX

Token Response

Token Response (Respuesta de Token) Petición HTTP en *OIDC4VCI* que entrega a una *Wallet* un *token* para autenticarse ante el *Issuer* para la emisión de credenciales. XXX

Trust Framework

Trust Framework (Marco de Confianza) Modelo que establece relaciones de confianza entre *Issuer* y *Relying Party* para determinar cuándo una aserción es confiable. XXX

TSP

Trust Service Provider (Proveedor de Servicios de Confianza) Entidad que ofrece servicios como firmas electrónicas, sellos y sellos de tiempo conforme a la ley. XXX

U

URI

Universal Resource Identifier (Identificador Uniforme de Recursos) Secuencia de caracteres que identifica recursos lógicos o físicos en la web, como personas, lugares o conceptos. XXX

V

VC

Verifiable Credential (Credencial Verificable) Aserción sobre un *Subject* en un formato que permite verificar su integridad, autenticidad, identificar al *Issuer* e identificar al propio *Subject*. XXX

Verifiable Attestation

Verifiable Attestation (Testimonio Verificable) Modelo de datos definido por *EBSI* que establece una *Verifiable Credential* o testimonio de atributos. XXX

Verifier

Verifier (Verificador) Entidad que necesita confiar en la información proporcionada por el usuario para poder continuar su proceso, por ejemplo un proveedor de servicios en contexto *W3C Verifiable Credentials* que necesita recibir determinada información para decidir si deja acceder a su sistema. XXX

VP

Verifiable Presentation (Presentación Verificable) Contenedor de aserciones en forma de *Verifiable Credentials* o derivadas que mantienen integridad, autenticidad e identificación del *Issuer*, presentado por el *Holder*. XXX

VP Token

VP Token Token definido en *OIDC4VP* para devolver *Verifiable Presentations* como medio de intercambio seguro de su información. XXX

W

Wallet

Wallet (Cartera) Aplicación, generalmente móvil, que almacena *Verifiable Credentials* de un usuario u otras entidades, permitiendo presentarlas a terceros y gestionar los propios datos. XXX

Capítulo 1

Introducción

En este capítulo se establece el alcance de este Trabajo de Fin de Grado y la justificación de su necesidad para mejorar el modelo actual de identificación de usuarios en la Web. Además se detalla también la estructura del documento.

1.1. Contexto

En estos últimos años ha aumentado la preocupación por la pérdida de privacidad de los datos personales en internet. Esto ocasiona que se pueda perfilar y filtrar a los usuarios en base a gustos y creencias por parte de terceros a quienes se vende información, suplantaciones de identidad, chantajes etcétera.

También es un motivo de preocupación la creciente falta de libertad de expresión y la censura en medios digitales como portales de noticias, redes sociales, mensajería instantánea, entre otros. Estas deficiencias afectan a todos los ámbitos de la vida de los usuarios, como el ámbito económico-monetario, social, político e incluso la integridad física en caso de personas bajo amenaza debido a la creciente digitalización de la identidad de las personas.

Por otro lado es reseñable el caso de los países en vías de desarrollo. Estos carecen en su mayoría de un sistema bancario solvente para su población, con los problemas que conlleva de cara al comercio y, por tanto, al desarrollo económico. Además, según un artículo publicado por Reuters [2], los gobiernos muchos de estos países,

y de otros países desarrollados, son conocidos por producir grandes devaluaciones de sus respectivas monedas locales debido a la corrupción y pésimas políticas económicas que erosionan el poco dinero que tienen algunas familias algo más acomodadas que el grueso de su población, que soporta la pobreza extrema.

En último lugar es reseñable que, aún con la Web 2.0, existe una gran carga burocrática en muchos de los procesos administrativos que un ciudadano debe realizar en algún momento de su vida debido a que la mayoría de procesos digitales actuales no son capaces de llegar a un nivel de seguridad suficientemente elevado con una usabilidad aceptable. El DNIe es uno de los métodos que proporcionaría la seguridad más alta pero no es tan sencillo de utilizar como un usuario y password tradicional y, por tanto, no es tan habitual que sea utilizado.

Para solucionar algunos de estos problemas en este trabajo se plantea cambiar el actual modelo de Identidad Digital por un nuevo modelo de Identidad Digital Autosoberana. El actual modelo de Identidad Digital consiste en obtener información del usuario mediante un registro tradicional de los datos del usuario a través de un formulario en cada uno de los proveedores de servicios con los que interactúa. O bien mediante *Single Sign On* mediante una cuenta ya generada en algún proveedor de servicios, como Google, al que se puede considerar un tercero en el proceso, que ofrece este servicio a otros proveedores de servicios. Esto hace que el usuario deba acceder a través de una entidad centralizada, bien cada web en la que se registra para poder acceder a sus servicios, o bien a través de un tercero que proporciona los datos necesarios sobre su identidad a otro proveedor de servicios en el caso de utilizar Single Sign On. que es la que crea un conjunto de datos acerca del usuario que son lo que conforman la identidad ante la aplicación.

En estos modos de proceder del usuario ante la aplicación se va dejando trozos de su identidad, incluso repetidos, en los diferentes proveedores de servicios con los que interacciona pero sin poder utilizarlos a su conveniencia aún siendo de su propiedad. Además, a lo largo de la actividad del usuario en el sitio se generarán más datos que irán conformando la identidad digital del mismo, siendo siempre generados, controlados y dados de baja por la entidad en la que el usuario está realizando su actividad.

La Identidad Digital Autosoberana pretende que sea el usuario quien cree y maneje los datos referentes a su identidad digital para poder presentarlos o revocarlos cuando considere, manteniendo el ciclo de vida de esos datos en su control. Generalmente se asume que estos datos se encontrarán en una cartera digital o *wallet* que el usuario puede controlar de manera más o menos autónoma.

La Figura 1.1 muestra la evolución de los sistemas de identificación a lo largo del tiempo. En la Web 1.0 el usuario debía registrarse en cada sitio web que necesitara utilizar en contraposición a la Web 2.0 en la que se puede utilizar la cuenta de uno de estos sitios web para acceder a otros s. En la Web 3.0 se espera que el usuario pueda utilizar su identidad digital de manera autónoma para acceder a los servicios que necesite sin necesidad de ninguna entidad centralizada que gestione su identidad digital.

Figura 1.1: Del *login* y *password* al *Single Sign On* y de este a la identificación con *wallet*. Imagen tomada de [3].

1.2. El problema de la pérdida de control sobre los datos

Actualmente cuando los usuarios quieren acceder a los servicios proporcionados por una empresa, bien sea desde una aplicación móvil, aplicación web o aplicación de escritorio, deben proporcionar una serie de datos más o menos personal y sensible que se utilizará para proveerles del servicio y para obtener un rendimiento económico en muchos casos. El tratamiento al que se someten estos datos puede ser legal y conforme a lo descrito en los términos y condiciones pero no necesariamente ético. Además, se puede dar el caso en que se realice tratamiento ilegal de los datos de los usuarios, esto es, que las empresas realicen un uso no permitido por sus términos y condiciones publicados. Este tipo de tratamientos pueden tener consecuencias no deseables. A continuación se detallan dos ejemplos de estas consecuencias:

- **Interferencia en resultados electorales**. En 2018 tuvo lugar el *escándalo de Facebook-Cambridge Analytica* [4] en el que Facebook fue condenado a pagar una gran multa debido a la cesión de datos de sus usuarios para un uso no amparado por sus términos y condiciones a Cambridge Analytica, con pleno conocimiento de ello. Esta situación es aún más grave ya que Cambridge Analytica es una firma que se dedica al estudio del comportamiento del consumidor y estudio de tendencias y que asistió en las campañas electorales de diversos políticos estadounidenses. Este escándalo fue destapado durante la campaña electoral y la elección de Donald Trump como Presidente de los Estados Unidos de América, cuya campaña electoral también elaboró. En base a los datos de usuarios obtenidos de manera ilegítima crearon los anuncios para la campaña electoral de 2016 influyendo en el comportamiento de esas personas, mediante datos ilegales y sensibles, y, por tanto, en su elección ante las urnas, que dio la victoria a Donald Trump.

- **Publicidad basada en datos procesados sin consentimiento**. El 16 de julio de 2021 la Comisión Nacional de Protección de Datos de Luxemburgo impuso a Amazon la mayor multa hasta el momento por violar la Normativa General de Protección de Datos europea (GDPR). Se descubrió que Amazon procesaba los datos de sus clientes sin tener su consentimiento expreso para ello, para mostrar anuncios adaptados a cada cliente y, por tanto, generar un mayor beneficio económico [5].

1.3. La tecnología *Blockchain* como parte de la solución

1.3.1. Qué es una *blockchain*

Una red *blockchain* es una *base de datos distribuida* sobre un conjunto de dispositivos denominados *nodos*. Cada uno de estos nodos tiene una copia exacta de la base de datos y se mantienen sincronizadas entre sí. Estos nodos colaboran entre sí en base a un protocolo que determina la forma de gobierno de la red, esto es, cuándo se añade nueva información, quién puede añadirla, etc Una *blockchain* es un caso particular de Tecnología de Registro Distribuida (*DLT*) o *Distributed Ledger Technology*, que es un sistema digital para registrar transacciones de forma segura y descentralizada. En ella los datos se almacenan en una estructura de datos formada por *bloques*. Estos bloques contienen la información almacenada en forma de *transacciones* junto con metadatos relativos al bloque inmediatamente anterior, lo que permite verificar la validez de un nuevo bloque de datos que se quiera añadir a dicha

estructura a partir de los bloques anteriormente añadidos y validados. Que la adición de un nuevo bloque de datos a esta *cadena* de bloques sea válida o no dependerá de si se han seguido las reglas acordadas en el protocolo en que se basa la red. En la Figura 1.2 se puede ver una representación gráfica de esta explicación.

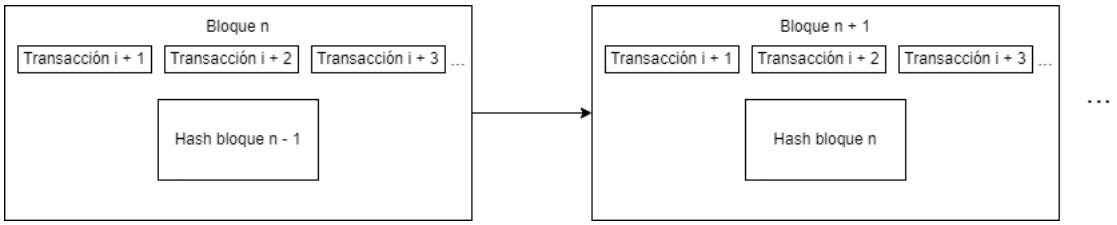


Figura 1.2: Estructura de datos de una blockchain. Imagen de autoría propia.

Los nodos participantes en la red lo hacen bajo las mismas condiciones, es decir, conforman conexiones *Peer To Peer* entre sí. Todos los nodos almacenan una copia de la cadena, proporcionando una alta disponibilidad y seguridad por diseño. Esta configuración proporciona gran seguridad debido a que en todos los bloques se almacena el *hash* del bloque anterior lo que, sumado a que cada nodo tiene una copia de la cadena de bloques, haría necesario que cada nodo aceptara recalcular el *hash* de todos los bloques de la cadena para que un bloque anterior pueda ser modificado. Esto hace que la cadena sea inmutable a no ser que se decida alterarla de forma consensuada, lo cual hace que sea vital que haya la mayor cantidad de nodos posibles e independientes entre sí para fomentar decisiones más democráticas. Es necesario apuntar que no sería necesario que todos los nodos acepten una modificación, solo deben hacerlo la mayoría de ellos aunque depende del *protocolo de consenso* correspondiente a la red concreta con la que se esté tratando.

Según como se regule la entrada de nuevos nodos a la *blockchain* y la visibilidad de la información almacenada en ella se puede hablar de diferentes tipos de blockchain:

- *Pública* si cualquier entidad puede decidir participar en cualquier momento con uno o más nodos y la información almacenada puede ser consultada por cualquiera.
- *Privada* si solo determinadas entidades previamente aprobadas pueden decidir participar con nuevos nodos y la información almacenada solo puede ser consultada por los participantes.
- *Pública-permisionada* si solo determinadas entidades previamente aprobadas pueden decidir participar con nuevos nodos.

1.3.1.1. Ethereum *blockchain* como caso particular de tecnología Blockchain pública

La *blockchain* de Ethereum cumple con todo lo descrito acerca de las *blockchains*, con la peculiaridad de que también permite desplegar código en su lenguaje, Solidity, mediante una transacción que queda grabado en un bloque concreto de los producidos dentro de la blockchain que podrá ser ejecutado por los nodos que soportan la red. A este código se le denomina *Smart Contract* o contrato inteligente. Dicho código es inmutable, esto es, no es posible modificarlo sin generar un nuevo despliegue, una vez desplegado, con lo que surjen problemas a la hora de tener que actualizar el mismo para solucionar errores o añadir funcionalidad. Que tenga lugar un redespliegue del código en la red implica que este nuevo código solo tiene acceso a nuevos datos que se almacenen, no a los que hayan sido almacenados por el código anterior. Para permitir utilizar el mismo almacén de datos dentro de la cadena en las siguientes evoluciones de un contrato inteligente existen patrones de diseño específico como el *Patrón Proxy* o el *Patrón Diamante*. El patrón Proxy se basa en tener un contrato inteligente que actúe como intermediario entre el contracto que almacena los datos con los que se trabaja, que es el propio contrato proxy, y el contrato que maneja la lógica de negocio. De esta forma, al cambiar el contrato que maneja la lógica de negocio se sigue utilizando el mismo espacio de almacenamiento mientras que se permite evolucionar la funcionalidad mediante el contrato de lógica. Por otro lado, el patrón Diamante persigue el mismo objetivo pero de manera ligeramente diferente. En este caso el contrato *Diamante*, equivalente al contrato proxy, es el que actúa como intermediario entre el contrato de almacenamiento y los diferentes contratos de lógica de negocio o *Facetas*. El patrón Diamante es más mantenible por ser más modular además de resolver algunas limitaciones sobre el tamaño del bytecode de los contratos a cambio de una mayor complejidad de implementación.

1.3.2. Ventajas de la tecnología *Blockchain*

Conocido el funcionamiento básico de una blockchain se puede concluir que la tecnología Blockchain presenta las siguientes ventajas:

- **Inmutabilidad**, impide que actores maliciosos manipulen de manera arbitraria la información almacenada en la blockchain, incluso si esta es pública. Se podría, por ejemplo, dejar pruebas en la blockchain de que un usuario ha decidido dejar de ceder ciertos datos a determinada entidad a la que se los proporcionó en primer

lugar y, basándose en la inmutabilidad de la cadena, podría tener validez legal.

- **Descentralización**, los nodos están distribuidos geográficamente y además la toma de decisiones viene regulada según el protocolo de consenso correspondiente. Puede no estar presente si no hay un número suficiente de nodos o si todos ellos pertenecen a una misma entidad, responden ante una misma entidad judicial, si no tienen una distribución geográfica suficientemente distribuida etc
- **Auditabilidad pública**, siempre que una blockchain sea pública será posible consultar cada dato almacenado en ella junto con los metadatos que se haya decidido almacenar.
- **Programabilidad**, gracias a los contratos inteligentes Ethereum permite automatizar lógica compleja y construir aplicaciones descentralizadas (*dApps*) sobre la propia red. Esto permite, por ejemplo, definir reglas automáticas para la gestión de datos de usuarios o establecer relaciones de confianza sin intervención humana.
- **Resistencia a la censura**, al no depender de un único servidor o autoridad central, resulta extremadamente difícil para gobiernos o entidades bloquear transacciones o impedir interacciones con contratos desplegados. Esto garantiza un alto nivel de autonomía y libertad operativa para los usuarios.

1.4. Objetivos

Los objetivos planteados en este Trabajo de Fin de Grado son los siguientes:

1. Justificar la necesidad de implantación de un modelo de Identidad Digital Autosoberana.
2. Justificar las ventajas de usar la tecnología Blockchain para implementar este nuevo modelo y utilizarla en el desarrollo.
3. Desarrollar un modelo teórico que permita la implementación de un modelo de Identidad Digital Autosoberana.
4. Realizar el análisis y diseño de un sistema de emisión y verificación de titulaciones digitales de la UVa aplicando el modelo de Identidad Digital Autosoberana.
5. Desarrollar una Prueba De Concepto que aplique el modelo de Identidad Digital Autosoberana a una propuesta ficticia de la emisión de titulaciones digitales de la Universidad de Valladolid y su presentación a terceros

que así lo requieran para autorizar el acceso de los usuarios a sus servicios.

1.5. Estructura del trabajo

En cuanto a la estructura del documento este se va a estructurar en los siguientes capítulos:

- Capítulo 1: Introducción. En este capítulo se trata el Estado del Arte, se justifica la motivación del objetivo perseguido y se concretan los objetivos y la estructura que sigue el trabajo. Para ello se explican los problemas relacionados con la privacidad de los datos de los usuarios que existen actualmente en la Web 2.0. También se explica por qué el hecho de que estos datos se vean comprometidos puede traer problemas a sus propietarios aunque parezcan inofensivos. Por último se explica muy simplificadamente el funcionamiento de la tecnología Blockchain para justificar que en el caso tratado en este trabajo puede ser muy útil y se justifica el cambio del modelo de Identidad Digital actual por uno nuevo que arregle sus grandes carencias.

- Capítulo 2: Planificación. En este capítulo se presenta la planificación del proyecto que ha llevado a este Trabajo de Fin de Grado. También se comentan las herramientas que utilizadas y el presupuesto necesario para ejecutar el proyecto.

- Capítulo 3: Modelo teórico para la Identidad Digital Autosoberana. En este capítulo se explican de manera teórica todos los nuevos conceptos, protocolos, formas de modelar la información, y, en definitiva, todo lo necesario para dar un soporte teórico al análisis y diseño posterior de un caso concreto.

- Capítulo 4: Requisitos. En este capítulo se detallan los requisitos funcionales y no funcionales que debe cumplir el caso concreto a desarrollar.

- Capítulo 5: Análisis. En este capítulo se detalla el diagrama de clases en análisis y los diagramas de secuencia correspondientes a los casos de uso que se han definido en el capítulo anterior.

- Capítulo 6: Diseño. En este capítulo se detalla la arquitectura del sistema y se presentan los diagramas de clases en diseño, además del diagrama de componentes y la arquitectura física del sistema.

- Capítulo 7: Implementación. En este capítulo se detalla la implementación de la Prueba De Concepto. Por un

lado se detallan las librerías y frameworks utilizados y por otro lado se presenta una demostración completa de la Prueba De Concepto, incluyendo logs que desmuestran su funcionamiento interno.

- Capítulo 8: Conclusiones. En este capítulo se plantean posibles líneas de trabajo futuras.

Capítulo 2

Planificación

En este capítulo se detalla la planificación temporal del proyecto para poder finalizarlo con la calidad necesaria en el marco temporal disponible. También se explica la metodología utilizada para la planificación y se detalla el presupuesto junto con un análisis de riesgos del proyecto.

2.1. Metodología utilizada

La planificación del Trabajo de Fin de Grado se ha basado principalmente en un enfoque iterativo e incremental, adoptando prácticas de la metodología ágil SCRUM, como el trabajo por *sprints*, las reuniones de seguimiento y la entrega continua de incrementos funcionales con cada *sprint*. Esta metodología se basa en trabajar en iteraciones cortas y regulares, a lo que se le da el nombre de *sprints* con el objetivo de conseguir una entrega continua de valor, en forma de mejoras a la Memoria o funcionalidad de la Prueba de Concepto. Esta forma de trabajo combinada con reuniones frecuentes para validar los avances hacen que en cuanto se detecta una desviación en el desarrollo se pueda corregir rápidamente, lo que es especialmente útil en un Trabajo de Fin de Grado (TFG) donde el tiempo es limitado y la calidad del contenido es fundamental.

No obstante, la organización del trabajo ha seguido una estructura que se alinea con el Proceso Unificado, abordando de forma explícita y secuencial las fases de análisis de requisitos, diseño, implementación y validación.

Esta combinación ha permitido aprovechar la flexibilidad de Scrum para adaptarse a cambios durante el desarrollo, algo vital en un TFG basado en tecnología tan experimental como este, sin renunciar al enfoque sistemático y controlado del Proceso Unificado.

En consecuencia, puede decirse que el proyecto ha seguido una metodología híbrida, en la que se integran prácticas ágiles para manejar el ciclo de vida del proyecto dentro de un marco de proceso más estructurado, lo que ha resultado especialmente útil en el contexto académico y en el desarrollo individual del proyecto.

Aunque el marco Scrum propone el uso de historias de usuario como técnica habitual para reflejar requisitos, en este proyecto se ha optado por una planificación basada en tareas y subtareas. Esta decisión responde a la naturaleza del TFG, donde no existe un interlocutor externo que ejerza de *Product Owner* ni un usuario final que valide directamente los entregables. El enfoque basado en tareas permite una mayor granularidad, trazabilidad y control del progreso, adaptándose mejor al contexto académico y unipersonal del proyecto. Esta adaptación respeta los principios fundamentales de Scrum, como la iteración, la inspección y la entrega incremental, sin introducir rigidez innecesaria. Los sprints definidos han sido de 2 semanas de duración. Se ha utilizado la herramienta SCRUM *Taiga* para manejar cada sprint. Se ha ido definiendo cada uno de ellos componiéndolos mediante tareas que se han ido creando en el *Product Backlog*, que es la lista de tareas global.

En la Figura 2.1 se pueden ver los principales elementos de la metodología SCRUM.

2.2. Planificación

Los *sprint* se han ido planteando sin contenido fijo para cada uno debido a la imposibilidad de estimar el tiempo disponible para cada uno de los mismos por la situación laboral del estudiante. Por ello se elaboró un plan de trabajo con fechas y horas estimadas al que tratar de ceñirse *sprint* a *sprint*. Este se puede observar en la Figura 2.2.

2.3. Seguimiento de los *sprints*

2.3.1 *Sprint* 1 (22/03/22 - 04/04/22)

En este *sprint* se dió inicio al proyecto, con varias reuniones con el tutor para establecer los apartados fundamentales de la memoria y la forma de trabajo. También se realizó la investigación inicial sobre la tecnología Blockchain de cara a elaborar el capítulo de introducción.

Tareas	**Subtareas**	**Tiempo empleado**	**Estado**
Definición estructura memoria	■ Reuniones ■ Construcción del esqueleto en LaTex	6 horas	Finalizada
Estudio teórico blockchain	■ Revisión del funcionamiento de la tecnología Blockchain	4 horas	Finalizada

Tabla 2.1: Desglose de tareas del *sprint* 1.

2.3.2 *Sprint* 2 (05/04/22 - 18/04/22)

En este *sprint* se redactó la introducción en base a la información acerca de la blockchain. También se realizó el análisis de riesgos del proyecto junto con la planificación del mismo.

Tareas	**Subtareas**	**Tiempo empleado**	**Estado**
Capítulo de introducción	■ Redacción del capítulo de introducción.	6 horas	En progreso
Análisis de riesgos	■ Realización del análisis de riesgos.	2 horas	Finalizada
Planificación	■ Realización de la sección de planificación del proyecto.	2 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.2: Desglose de tareas del *sprint* 2.

2.3.3 *Sprint* 3 (19/04/22 - 02/05/22)

En este *sprint* se inició la investigación acerca de la Identidad Digital Autosoberana.

Tareas	Subtareas	Tiempo empleado	Estado
Investigación de la Identidad Digital Autosoberana	▪ Comprensión del modelo de Identidad Digital Autosoberana del W3C	12 horas	En progreso
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.3: Desglose de tareas del *sprint* 3.

2.3.4 *Sprint* 4 (03/05/22 - 16/05/22)

En este *sprint* se continuó la investigación acerca de la Identidad Digital Autosoberana.

Tareas	Subtareas	Tiempo empleado	Estado
Investigación de la Identidad Digital Autosoberana	▪ Comprensión del modelo de Identidad Digital Autosoberana del W3C	14 horas	En progreso
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.4: Desglose de tareas del *sprint* 4.

2.3.5 *Sprint* 5 (17/05/22 - 30/05/22)

En este *sprint* se continuó la investigación acerca de la Identidad Digital Autosoberana a la vez que se fue incluyendo en la memoria.

Tareas	Subtareas	Tiempo empleado	Estado
Investigación de la Identidad Digital Autosoberana	■ Comprensión del modelo de Identidad Digital Autosoberana del W3C	12 horas	Finalizada
Modelo teórico para la Identidad Digital Autosoberana	■ Elaboración de las secciones relacionadas con el modelo de Identidad Digital Autosoberana según el estándar del W3C.	8 horas	En progreso
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.5: Desglose de tareas del *sprint* 5.

2.3.6 *Sprint* 6 (31/05/22 - 13/06/22)

En este *sprint* se terminó la sección acerca de la Identidad Digital Autosoberana, con más horas empleadas de las inicialmente previstas.

Tareas	Subtareas	Tiempo empleado	Estado
Modelo teórico para la Identidad Digital Autosoberana	■ Elaboración de las secciones relacionadas con el modelo de Identidad Digital Autosoberana según el estándar del W3C.	8 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.6: Desglose de tareas del *sprint* 6.

2.3.7 *Sprint* 7 (20/09/22 - 03/10/22)

Tras una temporada de no poder trabajar en el proyecto debido al trabajo del estudiante, se retoma el trabajo en el proyecto. En este *sprint* se estudió el protocolo *Self Issued OpenID Connect v2* (SIOPv2) para proponer aplicarla junto con el modelo de Identidad Digital Autosoberana del W3C como propuesta de un nuevo modelo de identidad completo.

Tareas	Subtareas	Tiempo empleado	Estado
Estudio de SIOPv2	▪ Estudio del protocolo SIOPv2 para su aplicación en el modelo de Identidad Digital Autosoberana.	20 horas	En progreso
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.7: Desglose de tareas del *sprint* 7.

2.3.8 *Sprint* 8 (04/10/22 - 17/10/22)

En este *sprint* se continuí el estudio del protocolo SIOPv2.

Tareas	Subtareas	Tiempo empleado	Estado
Estudio de SIOPv2	▪ Estudio del protocolo SIOPv2 para su aplicación en el modelo de Identidad Digital Autosoberana.	11 horas	En progreso
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.8: Desglose de tareas del *sprint* 8.

2.3.9 *Sprint* 9 (18/10/22 - 31/10/22)

En este *sprint* se terminó el estudio del protocolo SIOPv2.

Tareas	Subtareas	Tiempo empleado	Estado
Estudio de SIOPv2	▪ Estudio del protocolo SIOPv2 para su aplicación en el modelo de Identidad Digital Autosoberana.	8 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.9: Desglose de tareas del *sprint* 9.

2.3.10 *Sprint* 10 (01/11/22 - 14/11/22)

En este *sprint* se comenzó a trabajar en la sección acerca del protocolo SIOPv2.

Tareas	Subtareas	Tiempo empleado	Estado
Redacción de la sección acerca de SIOPv2	▪ Redacción de la sección pertinente.	8 horas	En progreso
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.10: Desglose de tareas del *sprint* 10.

2.3.11 *Sprint* 11 (15/11/22 - 28/11/22)

En este *sprint* se terminó la sección acerca del protocolo SIOPv2.

Tareas	Subtareas	Tiempo empleado	Estado
Redacción de la sección acerca de SIOPv2	▪ Redacción de la sección pertinente.	10 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.11: Desglose de tareas del *sprint* 11.

2.3.12 *Sprint* 12 (29/11/22 - 12/12/22)

En este *sprint* se realizaron correcciones propuestas por el tutor para la memoria.

Tareas	Subtareas	Tiempo empleado	Estado
Correciones en la memoria	▪ Corregida el capítulo de introducción y el del modelo de Identidad Digital Autosoberana W3C.	8 horas	En progreso
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.12: Desglose de tareas del *sprint* 12.

2.3.13 *Sprint* 13 (13/12/22 - 26/12/22)

En este *sprint* se terminaron las correcciones propuestas por el tutor para la memoria.

Tareas	Subtareas	Tiempo empleado	Estado
Correciones en la memoria	▪ Corregido el capítulo acerca de SIOPv2.	8 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.13: Desglose de tareas del *sprint* 13.

2.3.14 *Sprint* 14 (02/05/23 - 15/05/23)

Nuevamente tuvo lugar una temporada de no poder trabajar en el proyecto debido a la carga de trabajo del estudiante. En este *sprint* se estudió el protocolo *OpenID Connect For Verifiable Presentations* (OIDC4VP) para proponer aplicarlo junto con SIOPv2 y el modelo de Identidad Digital Autosoberana del W3C como nueva propuesta de modelo de identidad completo.

Tareas	Subtareas	Tiempo empleado	Estado
Estudio de OIDC4VP	▪ Estudio del protocolo.	16 horas	En progreso
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.14: Desglose de tareas del *sprint* 14.

2.3.15 *Sprint* 15 (16/05/23 - 29/05/23)

En este *sprint* se continuó el estudio del protocolo OIDC4VP.

Tareas	Subtareas	Tiempo empleado	Estado
Estudio de OIDC4VP	■ Estudio del protocolo.	13 horas	En progreso
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.15: Desglose de tareas del *sprint* 15.

2.3.16 *Sprint* 16 (30/05/23 - 12/06/23)

En este *sprint* se finalizó el estudio del protocolo OIDC4VP. También se estudió como aplicarlo en conjunción con SIOPv2.

Tareas	Subtareas	Tiempo empleado	Estado
Estudio de OIDC4VP	■ Estudio del protocolo. ■ Estudio de su combinación con SIOPv2.	15 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.16: Desglose de tareas del *sprint* 16.

2.3.17 *Sprint* 17 (13/06/23 - 26/06/23)

En este *sprint* se redactó la sección acerca del protocolo *OpenID For Verifiable Presentations* (OIDC4VP) y su aplicación conjunta con SIOPv2.

Tareas	Subtareas	Tiempo empleado	Estado
Redacción de la sección de OIDC4VP	■ Explicación del protocolo OIDC4VP. ■ Explicación de la combinación del protocolo con SIOPv2.	13 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.17: Desglose de tareas del *sprint* 17.

2.3.18 *Sprint* 18 (27/06/23 - 10/07/23)

En este *sprint* se estudió el protocolo *OpenID Connect For Verifiable Credentials Issuance* (OIDC4VCI) para proponer aplicarlo junto con SIOPv2, OIDC4VP y el modelo de Identidad Digital Autosoberana del W3C como propuesta final de modelo de identidad completo.

Tareas	**Subtareas**	**Tiempo empleado**	**Estado**
Estudio de OIDC4VCI	■ Estudio del protocolo. ■ Estudio de su combinación con SIOPv2.	15 horas	En progreso
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.18: Desglose de tareas del *sprint* 18.

2.3.19 *Sprint* 19 (11/07/23 - 24/07/23)

En este *sprint* se terminó el estudio del protocolo OIDC4VCI y su encaje con SIOPv2 y OIDC4VP.

Tareas	**Subtareas**	**Tiempo empleado**	**Estado**
Estudio de OIDC4VCI	■ Estudio del protocolo. ■ Estudio de su combinación con SIOPv2 y OID4VP.	12 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.19: Desglose de tareas del *sprint* 19.

2.3.20 *Sprint* 20 (25/07/23 - 07/08/23)

En este *sprint* se redactó la sección acerca del protocolo OIDC4VCI y su aplicación conjunta con SIOPv2 y OID4VP.

Tareas	Subtareas	Tiempo empleado	Estado
Redacción de la sección de OIDC4VCI	■ Explicación del protocolo. ■ Explicación del encaje con SIOPv2 y OID4VP.	14 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.20: Desglose de tareas del *sprint* 20.

2.3.21 *Sprint* 21 (08/08/23 - 21/08/23)

En este *sprint* se realizaron correcciones indicadas por el tutor sobre las secciones de OIDC4VP y OIDC4VCI.

Tareas	Subtareas	Tiempo empleado	Estado
Correcciones	■ Corrección de la sección acerca de OIDC4VP. ■ Corrección de la sección acerca de OIDC4VCI.	12 horas	En progreso
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.21: Desglose de tareas del *sprint* 21.

2.3.22 *Sprint* 22 (22/08/23 - 04/09/23)

En este *sprint* se terminaron las correcciones indicadas por el tutor sobre las secciones de OIDC4VP y OIDC4VCI.

Tareas	Subtareas	Tiempo empleado	Estado
Correcciones	■ Corrección de la sección acerca de OIDC4VP. ■ Corrección de la sección acerca de OIDC4VCI.	8 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.22: Desglose de tareas del *sprint* 22.

2.3.23 *Sprint* 23 (05/09/23 - 18/09/23)

En este *sprint* se estudió la normativa *electronic IDentification, Authentication and trust Services* (eIDAS) y la futura eIDAS 2.

Tareas	**Subtareas**	**Tiempo empleado**	**Estado**
Estudio normativa eIDAS y eIDAS2	-	6 horas	En progreso
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.23: Desglose de tareas del *sprint* 23.

2.3.24 *Sprint* 24 (19/09/23 - 02/10/23)

En este *sprint* se terminó el estudio de la normativa eIDAS y eIDAS 2.

Tareas	**Subtareas**	**Tiempo empleado**	**Estado**
Estudio normativa eIDAS y eIDAS2	-	7 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.24: Desglose de tareas del *sprint* 24.

2.3.25 *Sprint* 25 (03/10/23 - 16/10/23)

En este *sprint* se redactó la sección en la que se detalla la normativa eIDAS y eIDAS2.

Tareas	**Subtareas**	**Tiempo empleado**	**Estado**
Redacción secciones normativa eIDAS y eIDAS2	▪ Explicación normativa eIDAS. ▪ Explicación normativa eIDAS2.	10 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.25: Desglose de tareas del *sprint* 25.

2.3.26 *Sprint* 26 (17/10/23 - 30/10/23)

Este *sprint* se enfocó en realizar una descripción detallada del sistema del caso concreto escogido para aplicar el modelo de Identidad Digital Autosoberana planteado. También se realizó la elicitación de requisitos del mismo.

Tareas	**Subtareas**	**Tiempo empleado**	**Estado**
Descripción detallada del sistema	-	3 horas	Finalizada
Elicitación de requisitos	▪ Requisitos funcionales. ▪ Requisitos no funcionales. ▪ Requisitos información.	5 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.26: Desglose de tareas del *sprint* 26.

2.3.27 *Sprint* 27 (31/10/23 - 13/11/23)

Este *sprint* se enfocó en realizar el diagrama de casos de uso (CUs) y el modelo de dominio del caso concreto escogido.

Tareas	**Subtareas**	**Tiempo empleado**	**Estado**
Diagrama de Casos de Uso	-	4 horas	Finalizada
Modelo de Dominio	-	4 horas	En progreso
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.27: Desglose de tareas del *sprint* 27.

2.3.28 *Sprint* 28 (14/11/23 - 27/11/23)

Este *sprint* se enfocó en finalizar el modelo de dominio del caso concreto escogido.

Tareas	Subtareas	Tiempo empleado	Estado
Modelo de Dominio	-	8 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.28: Desglose de tareas del *sprint* 28.

2.3.29 *Sprint* 29 (28/11/23 - 11/12/23)

Este *sprint* se enfocó en realizar correcciones indicadas por el tutor a la sección correspondiente a la normativa eIDAS y eIDAS 2, la descripción detallada del sistema y los requisitos del sistema.

Tareas	Subtareas	Tiempo empleado	Estado
Correcciones de normativa, descripción detallada del sistema y requisitos	-	8 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.29: Desglose de tareas del *sprint* 29.

2.3.30 *Sprint* 30 (12/12/23 - 25/12/23)

Este *sprint* se enfocó en realizar correcciones indicadas por el tutor al modelo de dominio.

Tareas	Subtareas	Tiempo empleado	Estado
Correcciones del modelo de dominio	-	6 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.30: Desglose de tareas del *sprint* 30.

2.3.31 *Sprint* 31 (26/12/23 - 08/01/24)

Este *sprint* se enfocó en trabajar en la descripción de los casos de uso: ”Identificarse ante el sistema de emisión de titulaciones digitales” y ”Solicitar emisión de titulación digital”.

Tareas	**Subtareas**	**Tiempo empleado**	**Estado**
Descripción de los CU identificarse antes el sistema de emisión y solicitar emisión	▪ CU Identificarse ▪ CU Solicitar emisión	7 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.31: Desglose de tareas del *sprint* 31.

2.3.32 *Sprint* 32 (09/01/24 - 22/01/24)

Este *sprint* se enfocó en trabajar en la descripción de los casos de uso solicitar revocación de titulación digital y verificar posesión de titulación digital.

Tareas	**Subtareas**	**Tiempo empleado**	**Estado**
Descripción de los CU identificarse antes el sistema de emisión y solicitar emisión	▪ CU Solicitar revocación ▪ CU Verificar posesión	8 horas	En progreso
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.32: Desglose de tareas del *sprint* 32.

2.3.33 *Sprint* 33 (23/01/24 - 05/02/24)

Este *sprint* se enfocó en terminar la descripción de los casos de uso solicitar revocación de titulación digital y verificar posesión de titulación digital.

Tareas	Subtareas	Tiempo empleado	Estado
Descripción de los CU identificarse antes el sistema de emisión y solicitar emisión	■ CU Solicitar revocación ■ CU Verificar posesión	2 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.33: Desglose de tareas del *sprint* 33.

2.3.34 *Sprint* 34 (06/02/24 - 19/02/24)

Este *sprint* se enfocó en trabajar en la realización en análisis del CU Identificarse ante el sistema de emisión de titulaciones digitales.

Tareas	Subtareas	Tiempo empleado	Estado
Realización en análisis del CU Identificarse ante el sistema de emisión de titulaciones digitales	-	8 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.34: Desglose de tareas del *sprint* 34.

2.3.35 *Sprint* 35 (20/02/24 - 04/03/24)

Este *sprint* se enfocó en trabajar en la realización en análisis del CU Solicitar emisión titulación digital.

Tareas	Subtareas	Tiempo empleado	Estado
Realización en análisis del CU Solicitar emisión titulación digital	-	10 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.35: Desglose de tareas del *sprint* 35.

2.3.36 *Sprint* 36 (05/03/24 - 18/03/24)

Este *sprint* se enfocó en trabajar en la realización en análisis del CU Solicitar emisión titulación digital.

Tareas	Subtareas	Tiempo empleado	Estado
Realización en análisis del CU Solicitar emisión titulación digital	-	10 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.36: Desglose de tareas del *sprint* 36.

2.3.37 *Sprint* 37 (19/03/24 - 01/04/24)

Este *sprint* se enfocó en trabajar en la realización en análisis del CU Solicitar revocación titulación digital.

Tareas	Subtareas	Tiempo empleado	Estado
Realización en análisis del CU Solicitar revocación titulación digital	-	9 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.37: Desglose de tareas del *sprint* 37.

2.3.38 *Sprint* 38 (02/04/24 - 15/04/24)

Este *sprint* se enfocó en trabajar en la realización en análisis del CU Solicitar revocación titulación digital.

Tareas	Subtareas	Tiempo empleado	Estado
Realización en análisis del CU Verificar posesión titulación digital	-	11 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.38: Desglose de tareas del *sprint* 38.

2.3.39 *Sprint* 39 (16/04/24 - 29/04/24)

Este *sprint* se enfocó en realizar correcciones en la realización en análisis del CU Identificarse ante el sistema de emisión de titulaciones digitales, así como en la realización en análisis del CU Solicitar emisión titulación digital. También se realizó el diagrama de clases del análisis.

Tareas	Subtareas	Tiempo empleado	Estado
Correcciones CU Identificarse ante el sistema de emisión de titulaciones digitales y Solicitar emisión	-	8 horas	Finalizada
Diagrama de clases del análisis	-	6 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.39: Desglose de tareas del *sprint* 39.

2.3.40 *Sprint* 40 (30/04/24 - 13/05/24)

Este *sprint* se enfocó en realizar correcciones en la realización en análisis del CU Solicitar revocación titulación digital, así como en la realización en análisis del CU Verificar posesión titulación digital.

Tareas	Subtareas	Tiempo empleado	Estado
Correcciones CU Solicitar revocación titulación digital y Verificar posesión titulación digital	-	8 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.40: Desglose de tareas del *sprint* 40.

2.3.41 *Sprint* 41 (14/05/24 - 27/05/24)

Este *sprint* se enfocó en realizar el desarrollo correspondiente a la interfaz de usuario del emisor de titulaciones digitales.

Tareas	Subtareas	Tiempo empleado	Estado
Desarrollo interfaz de usuario del emisor de titulaciones	■ Creación de repositorios y configuración de herramientas de desarrollo. ■ Página de inicio	8 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.41: Desglose de tareas del *sprint* 41.

2.3.42 *Sprint* 42 (28/05/24 - 10/06/24)

Este *sprint* se enfocó en finalizar el desarrollo correspondiente a la interfaz de usuario del emisor de titulaciones digitales. Faltaba conectarlo con el lado del servidor, que se abordó en siguientes *sprints*.

Tareas	Subtareas	Tiempo empleado	Estado
Desarrollo interfaz de usuario del emisor de titulaciones	■ Página de selección de titulación a emitir ■ Página de emisión de titulación	13 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.42: Desglose de tareas del *sprint* 42.

2.3.43 *Sprint* 43 (11/06/24 - 24/06/24)

Este *sprint* se enfocó en el desarrollo correspondiente a la parte del servidor, o *backend*, del emisor de titulaciones digitales.

Tareas	Subtareas	Tiempo empleado	Estado
Desarrollo backend del emisor de titulaciones	-	14 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.43: Desglose de tareas del *sprint* 43.

2.3.44 *Sprint* 44 (25/06/24 - 08/07/24)

Este *sprint* se enfocó en conectar la interfaz de usuario del emisor con el backend correspondiente al mismo. También se realizaron pruebas de la interfaz de usuario y del backend, solucionando los errores encontrados.

Tareas	**Subtareas**	**Tiempo empleado**	**Estado**
Conexión del frontal del emisor de titulaciones con el backend	-	2 horas	Finalizada
Testeo del emisor de titulaciones	-	15 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.44: Desglose de tareas del *sprint* 44.

2.3.45 *Sprint* 45 (09/07/24 - 22/07/24)

Este *sprint* se enfocó en desarrollar una vista adicional en el emisor de credenciales que permita ver las titulaciones ya emitidas, con la opción de revocar las que se considere oportuno.

Tareas	**Subtareas**	**Tiempo empleado**	**Estado**
Vista revocación de titulaciones	-	7 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.45: Desglose de tareas del *sprint* 45.

2.3.46 *Sprint* 46 (23/07/24 - 05/08/24)

Este *sprint* se enfocó en desarrollar una vista adicional en el emisor de credenciales que permita ver las titulaciones ya emitidas, con la opción de revocar las que se considere oportuno.

Tareas	Subtareas	Tiempo empleado	Estado
Desarrollo backend revocación de titulaciones	■ Creación de Base de Datos para almacenar las titulaciones emitidas.	17 horas	En progreso
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.46: Desglose de tareas del *sprint* 46.

2.3.47 *Sprint* 47 (06/08/24 - 19/08/24)

Este *sprint* se enfocó en continuar el desarrollo del backend del emisor relacionado con la revocación de titulaciones.

Tareas	Subtareas	Tiempo empleado	Estado
Desarrollo backend revocación de titulaciones	■ Desarrollo de una lista de revocaciones sobre la red blockchain de Ethereum ■ Conexión de la interfaz de usuario con el backend	19 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.47: Desglose de tareas del *sprint* 47.

2.3.48 *Sprint* 48 (20/08/24 - 02/09/24)

Este *sprint* se enfocó en el desarrollo de la interfaz de usuario del verificador de credenciales.

Tareas	Subtareas	Tiempo empleado	Estado
Desarrollo interfaz del verificador de titulaciones	■ Desarrollo de la página de inicio del verificador ■ Desarrollo de la página de presentación de titulacione ■ Desarrollo de la página de titulación verificada correctamente	16 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.48: Desglose de tareas del *sprint* 48.

2.3.49 *Sprint* 49 (03/09/24 - 16/09/24)

Este *sprint* se enfocó en el desarrollo del backend del verificador de credenciales.

Tareas	Subtareas	Tiempo empleado	Estado
Desarrollo backend verificador de titulaciones	-	20 horas	En progreso
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.49: Desglose de tareas del *sprint* 49.

2.3.50 *Sprint* 50 (17/09/24 - 30/09/24)

Este *sprint* se enfocó en terminar el desarrollo del backend del verificador de credenciales.

Tareas	Subtareas	Tiempo empleado	Estado
Desarrollo backend verificador de titulaciones	-	25 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.50: Desglose de tareas del *sprint* 50.

2.3.51 *Sprint* 51 (01/10/24 - 14/10/24)

Este *sprint* se enfocó en elaborar el diagrama de arquitectura del sistema, tanto desde el análisis ya realizado como desde la prueba de concepto desarrollada. También se realizó el diagrama de clases del diseño.

Tareas	**Subtareas**	**Tiempo empleado**	**Estado**
Realización diagrama de arquitectura del sistema	-	8 horas	Finalizada
Diagrama de clases del diseño	-	4 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.51: Desglose de tareas del *sprint* 51.

2.3.52 *Sprint* 52 (15/10/24 - 28/10/24)

Este *sprint* se enfocó en la realización en diseño del CU Identificarse ante el sistema de emisión de titulaciones digitales.

Tareas	**Subtareas**	**Tiempo empleado**	**Estado**
Realización en diseño del CU Identificarse ante el sistema de emisión	-	13 horas	En progreso
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.52: Desglose de tareas del *sprint* 52.

2.3.53 *Sprint* 53 (29/10/24 - 11/11/24)

Este *sprint* se enfocó en continuar la realización en diseño del CU Identificarse ante el sistema de emisión de titulaciones digitales.

Tareas	Subtareas	Tiempo empleado	Estado
Realización en diseño del CU Identificarse ante el sistema de emisión	-	10 horas	En progreso
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.53: Desglose de tareas del *sprint* 53.

2.3.54 *Sprint* 54 (12/11/24 - 25/11/24)

Este *sprint* se enfocó en terminar la realización en diseño del CU Identificarse ante el sistema de emisión de titulaciones digitales.

Tareas	Subtareas	Tiempo empleado	Estado
Realización en diseño del CU Identificarse ante el sistema de emisión	-	3 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.54: Desglose de tareas del *sprint* 54.

2.3.55 *Sprint* 55 (26/11/24 - 09/12/24)

Este *sprint* se enfocó en realizar el diagrama de componentes del sistema. También se trabajó en la realización en diseño del CU Revocar titulación digital.

Tareas	Subtareas	Tiempo empleado	Estado
Diagrama de componentes	-	2 horas	Finalizada
Realización en diseño del CU Revocar titulación digital	-	18 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.55: Desglose de tareas del *sprint* 55.

2.3.56 *Sprint* 56 (10/12/24 - 23/12/24)

Este *sprint* se enfocó trabajar en la realización en diseño del CU Verificar posesión titulación digital.

Tareas	Subtareas	Tiempo empleado	Estado
Realización en diseño del CU Verificar posesión titulación digital	-	9 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.56: Desglose de tareas del *sprint* 56.

2.3.57 *Sprint* 57 (24/12/24 - 06/01/25)

Este *sprint* se enfocó en realizar correcciones en varios diagramas derivados del progreso en el diseño del sistema.

Tareas	Subtareas	Tiempo empleado	Estado
Correcciones	■ Correcciones en el diagrama de clases del diseño ■ Correcciones en el diagrama de componentes	13 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.57: Desglose de tareas del *sprint* 57.

2.3.58 *Sprint* 58 (07/01/25 - 20/01/25)

Este *sprint* se enfocó en realizar correcciones en varios diagramas derivados del progreso en el diseño del sistema.

Tareas	Subtareas	Tiempo empleado	Estado
Correcciones	▪ Correcciones en el diagrama de clases del diseño	8 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.58: Desglose de tareas del *sprint* 58.

2.3.59 *Sprint* 59 (21/01/25 - 03/02/25)

Este *sprint* se enfocó en terminar con las correcciones pendientes, en este caso en el diagrama de componentes. También se hizo una revisión del flujo completo del emisor y verificador de titulaciones con el tutor, para asegurarse de que fuese suficientemente claro y entendible. Se detectó la posibilidad de utilizar archivos de extensión ".pkpass"para hacer que la demostración del emisor fuera más visual, permitiendo importarla en una aplicación tipo cartera digital en Android.

Tareas	Subtareas	Tiempo empleado	Estado
Correcciones	▪ Correcciones en el diagrama de componentes	4 horas	Finalizada
Revisión flujo completo emisor y verificador con el tutor	-	4 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.59: Desglose de tareas del *sprint* 59.

2.3.60 *Sprint* 60 (04/02/25 - 17/02/25)

Este *sprint* se enfocó en investigar el uso de archivos de extensión ".pkpass".

Tareas	Subtareas	Tiempo empleado	Estado
Investigación sobre archivos .pkpass	-	2 horas	En progreso
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.60: Desglose de tareas del *sprint* 60.

2.3.61 *Sprint* 61 (18/02/25 - 03/03/25)

Este *sprint* se enfocó en terminar la investigación acerca del uso de archivos de extensión ".pkpass".

Tareas	Subtareas	Tiempo empleado	Estado
Investigación sobre archivos .pkpass	-	2 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.61: Desglose de tareas del *sprint* 61.

2.3.62 *Sprint* 62 (04/03/25 - 17/03/25)

Este *sprint* se enfocó en trabajar en los presupuestos del proyecto.

Tareas	Subtareas	Tiempo empleado	Estado
Elaboración presupuestos del proyecto	-	4 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.62: Desglose de tareas del *sprint* 62.

2.3.63 *Sprint* 63 (18/03/25 - 31/03/25)

Este *sprint* se enfocó en trabajar en los presupuestos del proyecto.

Tareas	Subtareas	Tiempo empleado	Estado
Elaboración presupuestos del proyecto	-	4 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.63: Desglose de tareas del *sprint* 63.

2.3.64 *Sprint* 64 (01/04/25 - 14/04/25)

Este *sprint* se enfocó en trabajar en la realización en diseño del CU Solicitar emisión de titulaciones digitales.

Tareas	Subtareas	Tiempo empleado	Estado
Realización en diseño CU Solicitar emisión de titulaciones digitales	-	5 horas	En progreso
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.64: Desglose de tareas del *sprint* 64.

2.3.65 *Sprint* 65 (15/04/25 - 28/04/25)

Este *sprint* se enfocó en terminar la realización en diseño del CU Solicitar emisión de titulaciones digitales.

Tareas	Subtareas	Tiempo empleado	Estado
Realización en diseño CU Solicitar emisión de titulaciones digitales	-	4 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.65: Desglose de tareas del *sprint* 65.

2.3.66 *Sprint* 66 (29/04/25 - 12/05/25)

Este *sprint* se enfocó en terminar la realización en diseño del CU Solicitar emisión de titulaciones digitales. También se trabajó en conseguir desplegar los componentes del emisor de manera aislada y programática mediante contenedores y en el capítulo de implementación.

Tareas	**Subtareas**	**Tiempo empleado**	**Estado**
Realización en diseño CU Solicitar emisión de titulaciones digitales	-	4 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada
Redacción del capítulo de implementación	-	6 horas	Finalizada

Tabla 2.66: Desglose de tareas del *sprint* 66.

2.3.67 *Sprint* 67 (13/05/25 - 26/05/25)

Este *sprint* se enfocó en completar el capítulo de conclusiones y en realizar correcciones indicadas por el tutor.

Tareas	**Subtareas**	**Tiempo empleado**	**Estado**
Realización en diseño CU Solicitar emisión de titulaciones digitales	-	4 horas	Finalizada
Correcciones a la memoria	-	28 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.67: Desglose de tareas del *sprint* 67.

2.3.68 *Sprint* 68 (27/05/25 - 09/06/25)

Este *sprint* se enfocó en en realizar correcciones indicadas por el tutor.

Tareas	Subtareas	Tiempo empleado	Estado
Correcciones a la memoria	-	8 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.68: Desglose de tareas del *sprint* 68.

2.3.69 *Sprint* 69 (10/06/25 - 23/06/25)

Este *sprint* se enfocó en realizar la presentación PowerPoint del TFG.

Tareas	Subtareas	Tiempo empleado	Estado
Presentacion Powerpoint	-	8 horas	Finalizada
Reunión de seguimiento	-	1 hora	Finalizada

Tabla 2.69: Desglose de tareas del *sprint* 69.

2.4. Presupuesto

En la Figura 2.3 se visualiza el presupuesto final del proyecto. Como se puede observar las horas realizadas finalmente, 777 horas, exceden por bastante las 300 horas requeridas para la realización de este TFG. Esto ha sido debido a las varias y frecuentes reconexiones necesarias tras no poder trabajar en este TFG de manera continua por motivos laborales. Por otro lado 416 horas se han empleado en la investigación, redacción y correcciones, lo que es un 50 % de las horas empleadas en el proyecto aproximadamente. Esto es debido a que el TFG se ha centrado principalmente en la investigación de la Identidad Digital Autosoberana y su aplicación a un caso real mediante una prueba de concepto, y no tanto en la implementación de una solución completa y finalista.

El total del presupuesto es la suma de los recursos, más el sueldo por las horas del trabajo realizado por el estudiante. Se ha considerado que el salario que vendría cobrando una persona para la realización de este proyecto es de 13,45 € / hora, que es el salario medio de un Ingeniero Informático como Programador Junior en España según *Jooble.org*[6].

2.5. Riesgos y oportunidades

Como en todo proyecto hay una serie de riesgos y oportunidades, con una probabilidad e impacto determinadas. Los riesgos y oportunidades considerados para este proyecto pueden consultarse en la Figura 2.4 y 2.5. Para valorar qué hacer con cada riesgo en cada momento se ha utilizado la matriz impacto-probabilidad de la Figura 2.6

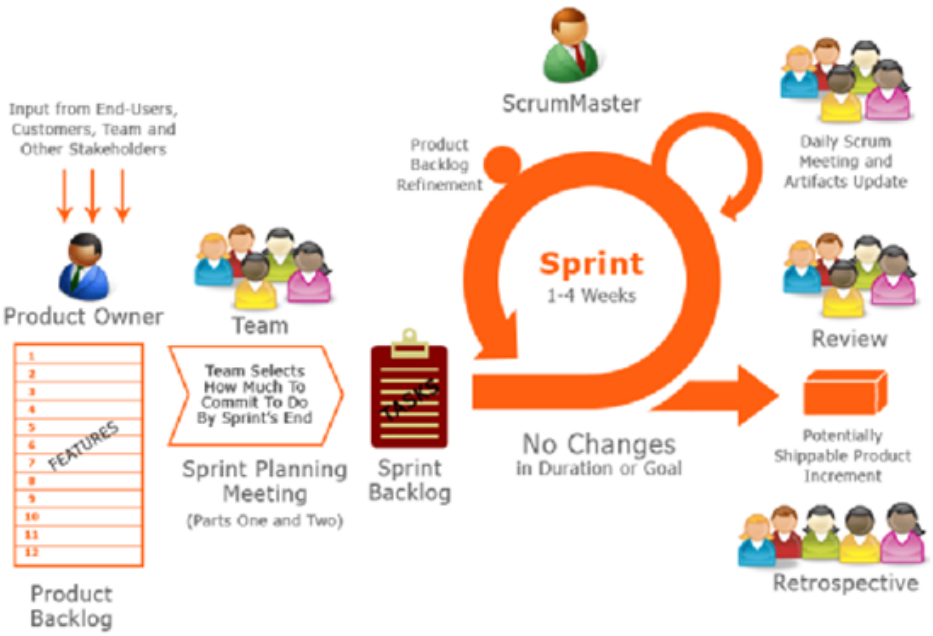


Figura 2.1: Metodología SCRUM [1]

PLAN DE ACTIVIDADES Sólo las del Alumno

	Fecha Inicio	Fecha Final	Participantes	Horas Estimadas
Reunión de arranque del TFG	mar-22	mar-22	Alejandro Nieto	2
Reuniones de seguimiento de TFG	mar-22	jun-25	Alejandro Nieto, Diego García	20
Trabajo inicial de estudio: Revisión de artículos técnicos	mar-22	ago-22	Alejandro Nieto	100
Trabajo inicial de estudio: Soluciones en el mercado similares, estudios teóricos	abr-22	abr-22	Alejandro Nieto	8
Introducción, resúmen en memoria y estado del arte	abr-22	abr-22	Alejandro Nieto	16
Definición y gestión de los requisitos del caso práctico Uva	ene-23	feb-23	Alejandro Nieto	25
Análisis funcional	feb-23	jul-23	Alejandro Nieto	30
Diseño funcional	jul-23	oct-23	Alejandro Nieto	40
Prueba de Concepto	oct-23	mar-24	Alejandro Nieto	30
Elaboración de la memoria	mar-24	abr-24	Alejandro Nieto	25
Presentación del TFG	abr-24	may-24	Alejandro Nieto	4
Correcciones	mar-22	jun-24	Alejandro Nieto	14
				300

Figura 2.2: Plan de trabajo del proyecto.

PLAN DE ACTIVIDADES - RECURSOS HUMANOS

	Horas Reales
Reunión de arranque del TFG	2
Reuniones de seguimiento de TFG	32
Trabajo inicial de estudio: Revisión de artículos técnicos	192
Trabajo inicial de estudio: Soluciones en el mercado similares, estudios teóricos	10
Introducción, resúmen en memoria y estado del arte	14
Definición y gestión de los requisitos del caso práctico Uva	8
Análisis funcional	83
Diseño funcional	80
Prueba de Concepto	156
Elaboración de la memoria	74
Presentación del TFG	15
Correcciones de la memoria	111
	777

RECURSOS MATERIALES

	Fecha Inicio U	Fecha Final U	Fabricante	Coste Personal	
Portátil	feb-22	jun-25	MacBook Pro 2021	2,000.00 €	
(licencia gratís)	mar-22	jun-25	Hammersley & John Lees-	0.00 €	
AstahUML PRO (licencia académica)	feb-24	jun-25	ChangeVision	72.00 €	36€ por año desde feb 2024
PlantUML	mar-22	jun-25	Arnaud Roques	0.00 €	
Microsoft Office estudiantes	mar-22	jun-25	Microsoft	149.00 €	
Nodo Infura (plan gratuito)	jun-22	jun-25	ConSensys	0.00 €	
Software Visual Studio CODE	mar-22	jun-25	Microsoft	0.00 €	
Taiga Project Management	feb-24	jun-25	Taiga Agile	0.00 €	
GitHub	feb-22	jun-25	Wanstrath	0.00 €	
				2,221.00 €	

PRESUPUESTO del PROYECTO

	Coste Total
Personal	10,450.65 €
Recursos materiales	2,221.00 €
TOTAL	12,671.65 €

Figura 2.3: Presupuesto.

Id de Riesgo	Categoria del riesgo	Descripción	Probabilidad	Impacto	Acciones de Mitigación	Responsable/s del Riesgo	Acciones de Contingenci	Estado	Proximidad	Alcance
R001	Funcionalidad	El proyectante no sea capaz de cerrar el alcance del trabajo al ser un trabajo teórico que busca una solución innovadora que debe culminar en un caso práctico	alta	alto	1. Limitar en requisitos 2. Limitar en objetivos (focalizar en qué se pretende hacer) 3.- Consensuar con los tutores del TFG	Alejandro Nieto	1. Modificar la fecha de entrega y retrasarla 2. Reducir la funcionalidad total a entregar	cerrado	En cada sprint	Entrega TFG
R002	Calendario	Debido a otras responsabilidades del proyectante o del tutor, que no de tiempo a realizar las tareas a realizar y se tenga que posponer la entrega del Trabajo más allá de Julio	baja	alto	1. Usar metodología ágil para tener un sistema de entregables (incrementos) y control y mejor adaptación al cambio 2.- Tener reuniones semanales y diarias (si hiciera falta) para poder tener un control cercano 3.- Empezar a trabajar con tiempo (6 meses)	Alejandro Nieto	1. Modificar la fecha de entrega y retrasarla 2. Reducir la funcionalidad total a entregar	mitigandose	En cada sprint	Entrega TFG
R003	Funcionalidad	El proyecto entre en un punto muerto, al ser un estudio puede que nos encontremos ante la posibilidad de que estemos en un punto muerto sin poder concluir el estudio	baja	alto	1. Estructurar el proyecto como todo método científico con una investigación previa y/o la elaboración de unas hipotesis. 2. Desarrollar un DAFO para estar seguros de la viabilidad del proyecto 3. Realizar una planificación exhaustiva	Alejandro Nieto	1. Cambiar el objeto y objetivos del estudio a realizar.	mitigandose	A principio, despues de planificación del TFG y antes de su ejecucción.	Fases iniciales del TFG

Figura 2.4: Riesgos y oportunidades.

Id de Riesgo	Categoria del riesgo	Descripción	Probabilidad	Impacto	Acciones de Mitigación	Responsable/s del Riesgo	Acciones de Contingenci	Estado	Proximidad	Alcance
R004	Empresa	El proyectante está realizando prácticas con tecnologías similares a las que se van a aplicar en el TFG, lo cual puede ayudarle a realizar el proyecto actual	media	bajo	1. Investigar posibles sinergias de lo que realiza en la empresa que pueda ayudarle a establecer nuevos conceptos o desarrollar los que ha planteado en el proyecto 2. Mencionar y aplicar conocimientos (considerando las restricciones de confidencialidad)	Alejandro Nieto	1. Buscar otras sinergias o aprovechamientos que ayuden al alumno en su cometido	aprovechando	Al principio, despues de planificación del TFG y antes de su ejecucción.	Hasta la entrega del TFG
R005	Empresa	El proyectante está trabajando con tecnologías similares a las que se van a aplicar en el TFG, lo cual puede ayudarle a realizar el proyecto actual	media	bajo	1. Investigar posibles sinergias de lo que realiza en la empresa que pueda ayudarle a establecer nuevos conceptos o desarrollar los que ha planteado en el proyecto 2. Mencionar y aplicar conocimientos (considerando las restricciones de confidencialidad)	Alejandro Nieto	1. Buscar otras sinergias o aprovechamientos que ayuden al alumno en su cometido	aprovechando	Durante la ejecución del TFG.	Hasta la entrega del TFG
R006	Calendario	Es que debido al trabajo no se pueda dedicar el proyectante a realizar la memoria del trabajo	baja	alto	1. Gestionar por metodología ágil la memoria 2. Dedicar en los fines de semana tiempo a la realización	Alejandro Nieto	1. Posponer la fecha de entrega del TFG	mitigandose	En Julio	Fases finales del TFG

Figura 2.5: Riesgos y oportunidades.

NIVEL DE RIESGO		IMPACTO		
		baja	medio	alta
PROBABILIDAD	bajo	bajo	bajo	medio
	medio	bajo	medio	medio
	alto	medio	medio	alta

bajo	No se mitiga, no hace falta hacer nada
medio	Si estamos en nivel de riesgo medio, tenemos que aplicar la mitigación
alta	Si estamos nivel de riesgo alto: mitigar lo que se pueda y preparar la contingencia

Figura 2.6: Matriz impacto-probabilidad.

Capítulo 3

Modelo teórico para la Identidad Digital Autosoberana

En este capítulo se presenta el marco teórico de la Identidad Digital Autosoberana y los estándares estudiados en este Trabajo de Fin de Grado para conseguir aplicar la Identidad Digital Autosoberana a un caso real.

3.1. ¿Qué es la Identidad Digital Autosoberana?

La Identidad Digital de un usuario está compuesta por todas y cada una de las acciones que realiza en la web que dan imagen de un individuo de cara al resto de usuarios, empresas y gobiernos [7]. Estos datos pueden ser, en algunos casos, datos sensibles que, en caso de ser compartidos con alguna entidad, es decir, un proveedor de servicios para el usuario, es necesario controlar su distribución y uso para poder evitar que se usen de manera inadecuada bajo el pretexto de ser usados para proveer de un servicio. En la Sección 1.2 se presentan algunos ejemplos de malos usos de los datos personales de los usuarios. Para solventar estos problemas de la actual Identidad Digital surge un nuevo concepto de Identidad Digital, denominado Identidad Digital Autosoberana, o *Self Sovereign Identity (SSI)*, cuyo principal objetivo es hacer que los usuarios que comparten sus datos con ciertas entidades puedan conocer claramente qué datos van a compartir con ellas y con qué finalidad puede ser utilizado

cada uno de ellos. Aunque esto debería asegurarse mediante los términos y condiciones de uso de cada proveedor de servicios digitales es necesario hacerlo de forma más útil y explícita al usuario. Este nuevo modelo permite detallar específicamente para qué acción se va a utilizar cada dato, en lugar de dar una descripción más general, y cada dato cedido se detalla concretamente. Cada dato cedido conforma una unidad diferente al resto de estos, a la que denominamos *credencial*. Además en la versión más estricta de SSI el usuario puede prohibir el acceso a esas credenciales presentadas en cualquier momento. Este tipo de credenciales se denominan *Credenciales Verificables* o *Verifiable Credentials* (*VC*), ya que es el modelo que se va a utilizar para ellas.

Es evidente que este nuevo concepto de identidad digitalse llevase a la práctica, supondría una mejora sustancial en cuanto a la privacidad de los usuarios en el mundo digital. Además, permite a los usuarios no tener que crear una cuenta para cada servicio consumido. Esto supondría un acceso homogéneo a los servicios que provee cada una de las entidades existentes en la actualidad, tanto públicas como privadas. Esto eliminaría la necesidad de tener una cuenta por cada entidad o tener que usar una cuenta de alguno de los grandes proveedores de servicios informáticos como Facebook, Amazon o Google, que serían los encargados de facilitar sus datos a un tercero que se le indique, con los riesgos que ello conlleva. Este último mecanismo de identificarse en un sistema con la cuenta de otro proveedor es lo que se denomina *Single Sign On* o *SSO* y puede ocasionar que sus acciones sean trazadas por el proveedor de datos acerca de su identidad, esto es, Amazon, Facebook, ya que el tercero ante el que quiere identificarse el usuario se conectará con estos cada vez que lo hagamos y se podrán correlacionar las identificaciones que se realicen en los diferentes sitios utilizados.

3.2. El estándar para Credenciales Verificables del W3C

El 3 de marzo de 2022 el World Wide Consortium (W3C) elaboró un estándar que permite modelar credenciales de uso cotidiano en formato digital [8], de tal forma que se pueda comprobar que estas son auténticas y saber quién las ha emitido para decidir si se confía en esa entidad o no. Cuando es posible saber si estas credenciales emitidas han sido manipuladas y se puede comprobar criptográficamente la autoría de las mismas entonces se denominan *Credenciales Verificables* o *Verifiable Credentials (VCs)*. A partir de las Credenciales Verificables se pueden generar *Presentaciones Verificables* o *Verifiable Presentations (VPs)*, que simplementen son listas de Credenciales Verificables. Estas también pueden ser verificadas para detectar una manipulación de las mismas y su autoría puede ser

comprobada por medios criptográficos. Estas contienen una o más Credenciales Verificables o datos derivados a partir de estas.

En la Figura 3.1 se muestran los roles principales y los flujos de información definidos por el estándar. Los roles definidos son:

- *Issuer* o *Emisor* que es sería desempeñado por una entidad, como puede ser la Policía Nacional en España, que genera Credenciales Verificables a partir de información de un Sujeto tras verificar su validez. Cabe destacar que esta emisión no tendría por qué realizarla una entidad confiable, como es el caso del ejemplo, por lo que será necesario acompañar este estándar de un *Marco de Confianza* o *Trust Framework*, que es un modelo en el que se define cuándo se puede confiar en una Credencial Verificable.
- *Holder* o *Titular*, que es aquella entidad que posee una o más VCs para poder presentárselas a un tercero que las requiera para proporcionarle una serie de servicios. En general suele ser el usuario o sujeto. Este puede utilizarlas para generar Presentaciones Verificables para un Verifier que se encargue de proporcionarle esos servicios.
- *Verifier* o *Verificador*, que es el tercero que provee de determinados servicios a Holders tras recibir determinadas aserciones mediante VPs. Un proveedor de servicios que permite iniciar sesión mediante cuentas de Google sería un ejemplo típico de Verificador.
- *Verifiable Data Registry* o *Registro de Datos Verificable* que es un sistema en el que se registran los identificadores de los distintos participantes del modelo, material criptográfico para comprobar las firmas digitales de las Credenciales y/o Presentaciones y otros datos relevantes como almacenar qué credenciales han sido revocadas y por tanto no tienen validez. En este trabajo será la propia *blockchain* pero no necesariamente tendría que ser así, podría ser una base de datos tradicional confiable, una base de datos descentralizada u otro tipo de registro que pueda ser considerado como confiable por los roles que participan en el modelo.

El *Issuer* genera VCs que entrega al Holder. El holder las presenta mediante VPs para conseguir los servicios que ofrece un Verifier. Por ejemplo, la Policía Nacional podría emitir una VC a un usuario que ha solicitado un certificado de antecedentes penales. Este lo presentaría a un banco que le ofrece una cuenta bancaria y que le pide que demuestre que no tiene antecedentes penales. El banco sería el Verifier y el usuario el Holder.

Cada participante que ejerce uno de los roles del modelo tiene asociado un *Decentralized Identifier*, o *DID*. Esto no es más que un identificador global y único para cada participante, que viene dado en forma de *URI* para que no se tenga que volver a emitir una credencial si esta se cambia de lugar de almacenamiento. Estos almacenes de Credenciales, que son utilizados por los *Holders*, pueden ser *software* o *hardware* y se les denomina *Wallets*.

Por último, es necesario destacar que en el estándar se advierte de que esta especificación no está diseñada para ocuparse de administrar la autorización de usuarios a los recursos correspondientes y que, por tanto, se requiere utilizar por encima de esta especificación un *protocolo de autorización*, como puede ser OpenID Connect, que se detalla más adelante. Además advierte que tampoco se fija un mecanismo de intercambio de estas credenciales mediante Presentaciones Verificables ni tampoco un mecanismo de emisión para las Credenciales, por lo que en siguientes secciones se detallarán los estándares adicionales que se han considerado para extender esta especificación y posibilitar la construcción de un sistema de Identidad Digital Autosoberana basado en estándares reconocidos. Esto significa que lo único que se define en el modelo de Credenciales Verificables del W3C es el modelo de datos por el cual se definen diferentes credenciales que sean necesarias dependiendo del caso de uso concreto y sus presentaciones. Además en este trabajo se ha considerado un formato de representación concreto para las credenciales, *JSON* combinado con *JSON-LD*, pero no tendría por qué utilizarse estos formatos, simplemente es la representación por defecto del modelo. JSON-LD es un método de codificación de datos en formato JSON enlazados entre si. Mapea las diferentes propiedades de los objectos JSON enlazados a *RDF*, que es un lenguaje que permite expresar las relaciones entre grafos, como es el caso de datos enlazados. Esto permite otorgar semántica a propiedades de los objetos JSON que de otra manera no tendrían.

En el Anexo A se presenta un ejemplo de Credencial Verificable en formato JSON-LD que modela la pertenencia de un sujeto a una universidad con el rol de alumno de esta, firmada para proteger su integridad y autenticidad. La Presentación Verificable que se correspondería a esta credencial para demostrar que se es alumno de la universidad indicada se encuentra en ese mismo anexo.

3.3. *Decentralized Identifiers*

Un Identificador Descentralizado, o *Decentralized Identifier (DID)*[9], es un identificador único que sirve sirven para identificar a las entidades que intervienen en el flujo del modelo de Credenciales Verificables. A diferencia de los identificadores tradicionales, que son considerados *federados* por ser generados y gestionados en todo momento por cada empresa que se haya encargado de su emisión, estos no son almacenados en *registros centralizados*, tales como pueden ser las bases de datos propietarias de cada empresa en que se almacenan los identificadores tradicionales. Esto hace que sea posible identificar a cada entidad con un identificador único global. La entidad propietaria de ese DID, que sería aquella a la que identifica, es capaz de probar que posee ese identificador ya que este está ligado a un material criptográfico que posee y, por tanto, se puede demostrar que es esa entidad a la que se emitió el DID en un primer lugar, consiguiendo así autoidentificarse sin intervención de ningún tercero.

Los DIDs tienen forma de Identificador Unico de Recursos, (URI) y cada uno se encuentra asociado a su *DID Document* correspondiente, que son un conjunto de datos que representan el material criptográfico público asociado al DID para permitir que se pueda comprobar que quien está usando un DID posee la clave privada asociada a este y es por tanto su dueño legítimo. En un DID se pueden distinguir las partes indicadas en la Figura [**?**]

Cada DID concreto y su respectivo DID Document son almacenados en el Verifiable Data Registry, como por ejemplo en una blockchain. Como una red blockchain es pública no se pueden almacenar los DID correspondientes a personas físicas porque son consideradas datos de carácter personal ante la normativa GDPR europea ya que permiten identificar de manera unívoca a la persona y por tanto comprometen su privacidad al ser combinados con otros datos relacionados con la persona. Esto ocasionaría que se pudiera seguir la actividad de una persona en la red violando su privacidad.

3.4. Protocolo *OpenID Connect*

OpenID Connect [10] es un protocolo de autenticación de usuarios ante un sistema estandarizado por la OpenID Foundation. En esta autenticación del *End-User* o usuario final la realiza un *Servidor de Autorización (Authorization Server)*, es decir, un tercero, que también se encarga de la transmisión de información del usuario para que otra

entidad pueda darle un servicio que solicite. Esto hace que la autenticación y recuperación de información básica del usuario esté delegada en ese tercero. Para ello se basa en el protocolo de autorización *OAuth2.0* [11]. Este protocolo es usado por Facebook o Google para ofrecer el *Single Sign On* presentado en la Sección 3.1.

3.4.1. Flujo básico del protocolo

Para entender un flujo básico de este protocolo deben tenerse claros los términos *Relying Party (RP)* o parte confiable, que es la entidad que solicita la autenticación del usuario y la delega en un tercero, al que denominaremos *OpenID Provider (OP)* o Proveedor OpenID. El OP es también el Authorization Server. Otra forma más genérica de denominar al OP en el contexto de la Identidad Digital es con el término *Identity Provider* o proveedor de identidad. El RP es la plataforma web o servicio que permite identificarse a un usuario a través de una cuenta de Google o Facebook, que serían quienes actuarían de OP. El flujo del protocolo, que se resume en la Figura 3.3, es el siguiente:

1. El RP envía una petición denominada Solicitud de Autenticación *(Authentication Request)* al OP para solicitar que autentique al usuario.

2. El OP autentica al usuario, por ejemplo, a través de una pantalla de *login* y *password* tradicional.

3. El OP envía una petición al RP denominada Respuesta de Autenticación *(Authentication Response)* en la que le envía un *ID Token* y, generalmente, también un *Access Token*. El ID Token es un *JSON Web Token (JWT)*, que es un objeto JSON que contiene aserciones relacionadas con el proceso de autenticación realizado por el OP. Con este *token* el usuario puede atacar al API subyacente del RP asegurando la autenticación y autorización del usuario correspondiente, todo ello basándose en un tercero como es el OP. El Access Token es una simple cadena de texto que actúa como autorización a la hora de intentar acceder a información del OP acerca del usuario.

4. Si la autorización es afirmativa el usuario puede acceder a recursos del RP utilizando el ID Token desde su agente de usuario *(user-agent)*, que puede ser el navegador web con el que accede a la web del RP. El RP pone a disposición del usuario el ID Token a través de su web.

5. Si el RP requiere información adicional acerca del usuario puede consultar al OP haciendo uso del *Access Token* mediante una Petición de Información del Usuario *(User Info Request)*.

6. El RP recibe la información solicitada mediante una Respuesta de Información del Usuario *(User Info Response)*.

Como ya se ha comentado, al utilizar esta solución el Service Provider o RP invoca en repetidas ocasiones durante el proceso de identificación del usuario e incluso posteriormente, al OP y este, por tanto, puede saber cada uno de los datos que está haciendo falta acerca de cada usuario. Con esta información puede intuir qué está realizando el usuario y explotar esos datos económicamente o de alguna otra forma.

3.5. Protocolo *Self Issued OpenID Connect v2*

En secciones anteriores se ha presentado el flujo característico del protocolo OpenID Connect en el que se confía en una entidad centralizada como es el Proveedor de Identidad (OP), que es quien provee de datos al proveedor de servicios que así se lo solicita en nombre del usuario, que puede trazar la actividad del usuario si así lo desea. Existe una evolución de este protocolo llamada *Self Issued OpenID Connect v2* o *SIOPv2* [12], que aún está en borrador y, por tanto, no es estándar todavía. Este protocolo propone que sea el propio usuario final el que actúe de Proveedor de Identidad auto-emitiendo aserciones u afirmaciones sobre sí mismo y autenticándose ante la Parte Confiable (RP) que así lo requiera. Esto haría que el intercambio de información de un RP con un tercero que autoriza al usuario, el OP, desaparezca, siendo sustituido por el usuario y colocando al usuario como intermediario tanto a la hora de la autenticación ante el OP, que ahora es él mismo, como a la hora de solicitar los servicios al RP.

Esta extensión del concepto de OP de OpenID Connect a un Proveedor OpenID Auto-emitido (*SIOP*) plantea la dificultad de por qué debería un RP confiar en información que un usuario totalmente desconocido para él le está entregando. Para establecer una relación de confianza entre el usuario y el RP es necesario interponer nuevamente a una tercera parte en la que el RP confíe. En este caso, se tratará de limitar sus acciones a las mínimas imprescindibles para asegurar esa confianza. Este tercero se encargará únicamente de dar fé de que cierta información acerca del usuario, en la forma de Credenciales Verificables, es fidedigna y auténtica. Ni se encargará de emitir *tokens* para que el RP autentique y autorice al usuario en sus APIs ni hará de intermediario para acceder a la información acerca del usuario cuando el RP lo necesite, a diferencia del OP en OpenID Connect. Esta cuestión no la resuelve este estándar, se requerirá establecer un Marco de Confianza (*Trust Framework*) adicional. Como

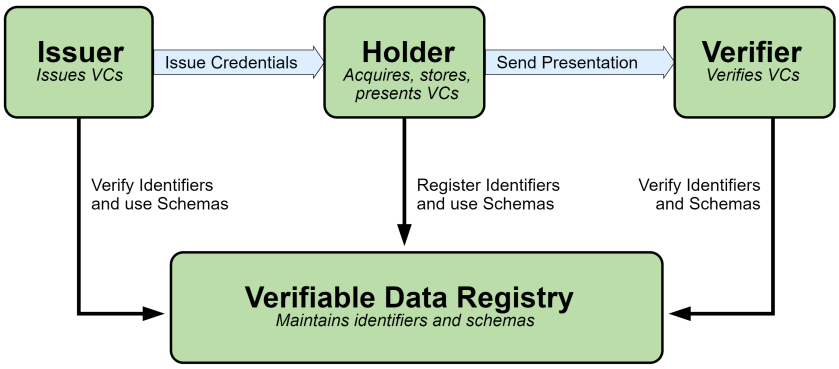


Figura 3.1: Roles y flujos de información del estándar W3C para Credenciales Verificables. Imagen tomada de [8].

Figura 3.2: Partes de un DID. Imagen tomada de [9].

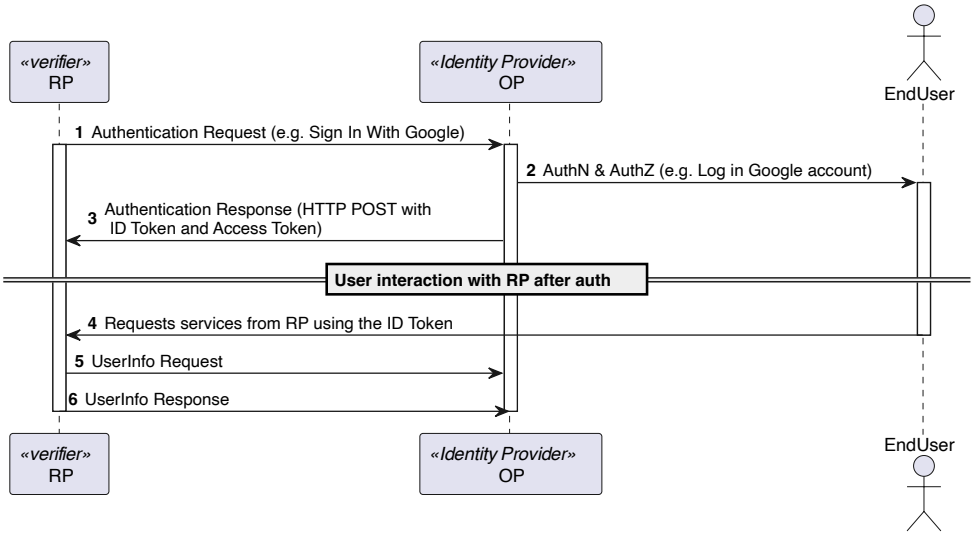


Figura 3.3: Flujo del protocolo OpenID Connect [10]. Imagen de autoría propia.

consecuencia de esa necesidad de un tercero aparece la figura del emisor de información confiable acerca del usuario, que no necesariamente tendría que emitirla en forma de Credenciales Verificables porque este estándar es agnóstico respecto al modelo de datos utilizado para las aserciones o afirmaciones acerca del usuario, que en nuestro modelo serán las credenciales verificables.

3.5.1. Flujo básico del protocolo

Una vez vistas las principales diferencias con OpenID Connect se puede explicar cómo quedaría el flujo de esta nueva versión del protocolo. Hay dos flujos posibles, dependiendo de si la interacción del usuario con el RP y con el SIOP ocurren en el mismo dispositivo o entre dispositivos diferentes. El flujo entre dispositivos diferentes tiene la particularidad de que no se puede realizar ningún tipo de redirección de uno al otro, ya que el SIOP no necesariamente dispone de endpoints. Las interacciones que ocurren en este flujo, resumidas en la Figura 3.4, son las siguientes:

1. El RP prepara una Authentication Request, que en este caso denominamos *SIOP Request* para solicitarle al SIOP del End-User que autentique y autorice al mismo. Representa esta petición como un código QR para poder hacerla llegar al dispositivo en el que se encuentra el SIOP desde aquel en que se está realizando la interacción entre End-User y RP, como podría ser un ordenador con la página web del proveedor de servicios que pretende utilizar el End-User. El usuario escanea el QR con el dispositivo en que tiene el SIOP.

   Si el SIOP se encuentra en un *smartphone*, que es lo más común, la Authentication Request se puede hacer llegar a este a través de un *deeplink* o *enlace universal*, que no es más que una *URI* o identificador único de un recurso, en lugar de mediante un QR. Este es una cadena que identifica de manera única a la aplicación SIOP, que se configura para esperar una llamada a través de esa URI personalizada, como podría ser, por ejemplo: `wallet://...`. Cuando la interacción transcurre en el mismo dispositivo el flujo cambia la representación en QR de la Petición de Autenticación (Authentication Request) por una redirección hacia la misma URI que se representa en el QR. Es importante destacar que podrían utilizarse otros métodos para hacer llegar la Authentication Request en lugar de usar un QR, pero este mecanismo es el más común.

2. La lectura del QR invoca a la autenticación del usuario del SIOP, como le solicita el RP. El protocolo no define

una forma de autenticación concreta.

3. Como respuesta a la Petición de Autenticación (Authentication Request) recibida el SIOP envía mediante una petición HTTP POST una Respuesta de Autenticación (Authentication Response), que contiene el *ID Token* y/o información acerca del usuario, dependiendo de lo que haya solicitado el RP en la Authentication Request.

4. El usuario accede a recursos protegidos del RP haciendo uso del *ID Token* que ha quedado en el *User-agent* o navegador web que ha utilizado el usuario para conectar al sitio del Service Provider.

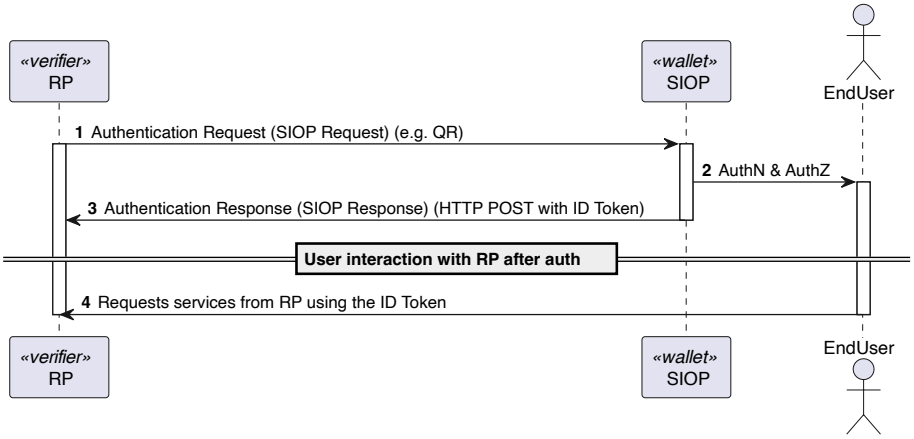


Figura 3.4: Flujo del protocolo SIOPv2 [12]. Imagen de autoría propia.

3.6. Protocolo *OpenID Connect for Verifiable Credentials Issuance*

OpenID Connect for Verifiable Credentials Issuance u *OIDC4VCI* [13] es el protocolo aún en borrador pero candidato a estándar de la OpenID Foundation que estandariza la emisión de Credenciales Verificables. A la hora de solicitar la emisión de credenciales a un Emisor (Issuer) determinado, en primer lugar, el usuario deberá autenticarse/autorizarse contra este para asegurar que está habilitado a solicitar la emisión de ciertas credenciales con la información necesaria que se le requiera para cada una de ellas. Tras este proceso se podrá emitir la Credencial correspondiente como tal. El flujo básico de este protocolo se muestra en la Figura 3.5. Los pasos que se siguen son los siguientes:

1. Se produce la autenticación del usuario ante el Emisor, bien mediante OAuth2, OpenID Connect u otro tipo de mecanismo. Puede ser un *login/password* tradicional.

2. El usuario proporciona la información que corresponda para poder generar una Credencial Verificable a partir de la misma.

3. El Issuer realiza una Oferta de Credenciales ()*Credential Offer*) hacia la wallet del usuario en la que utiliza un Código de Pre-Autorización ()*Pre-Authorized Code*), que es un código generado por el Emisor (Issuer) para dar acceso a la emisión de una determinada credencial solamente con presentarlo. Por esto mismo no constituye un método totalmente seguro.

4. La Oferta de Credenciales (Credential Offer) proporciona determinada información que debe ser utilizada para solicitar los metadatos necesarios para continuar el proceso al Emisor (Issuer) correspondiente.

5. Se solicita la introducción de un PIN, si así se requería en la Credential Offer, para añadir seguridad extra al uso del *Pre-Authorized Code*. Debe ser establecido y solicitado en un flujo diferente al de emisión y, por tanto, no queda establecido en este protocolo.

6. Se solicita al Emisor (Issuer) un *token* para poder solicitar la credencial ya emitida como tal haciendo uso del *Pre-Authorized Code* y el *PIN*. Esto se realiza mediante una *Token Request*.

7. El issuer envía un *token* denominado *access_token* que permitirá solicitar las credenciales emitidas para almacenarlas finalmente en la *wallet* del usuario. Esto se realiza mediante una *Token Response*.

8. Haciendo uso del *access_token* y proporcionando una Prueba de Posesión (*Proof of Posession*) del material criptográfico asociado a la clave pública de los usuarios a los que se quiere ligar la credencial que se está emitiendo. Se solicita la emisión de las credencial correspondiente mediante una Solicitud de Credencial (*Credential Request*). Cabe destacar que se habla de clave pública para no establecer una forma concreta de representar ese material criptográfico, como podría ser un DID cuyo DID Document haga referencia a esa clave pública y solo por el uso de ese DID para firmar un desafío (*challenge*), que no es más que un dato cualquiera, se pueda verificar al resolverlo que clave pública asociada coincide con la utilizada para firmar ese desafío. Con este desafío tendría esa Proof of Possession, que básicamente consiste en demostrar que el usuario es poseedor del material criptográfico al que se va a ligar la credencial.

9. Por último se recibe la credencial en la *wallet* mediante una Respuesta de Credencial (*Credential Response*).

3.7. Protocolo *OpenID Connect for Verifiable Presentations*

OpenID Connect for Verifiable Presentations o *OIDC4VP* [14] es el protocolo aún en borrador pero candidato a estándar de la OpenID Foundation que posibilita que los RP sean capaces de admitir la presentación de aserciones u afirmaciones en forma de Credenciales Verificables mediante el uso de Presentaciones Verificables y estandariza el procedimiento a realizar. Por sí mismo ya permite intercambiar aserciones en formar de Credenciales acerca del End-User de manera que se puede verificar su autenticidad, pero puede combinarse con el uso de SIOPv2 para permitir también autenticación y autorización en base a Credenciales, que es lo que pretendemos en este nuevo modelo de Identidad Digital. Es importante destacar que este candidato a estándar, a parte de ser compatible con el modelo de Credenciales del W3C también lo es con el modelo *ISO mDL* también conocido como *ISO/IEC 18013-5:2021* [15], entre otros. Este estándar ISO es simplemente otra modelo de datos diferente al del W3C para las Credenciales Verificables.

Para conseguir estos objetivos se añade un nuevo parámetro a la Authentication Request de OAuth2, que es el protocolo sobre el que se construye este estándar, como era el caso tanto de OpenID Connect como de SIOPv2. Este es el parámetro `presentation_definition`, que posibilita indicar al SIOP qué Credenciales Verificables o qué partes de estas, si se usa *selective disclosure*, se están pidiendo exactamente. Selective disclosure es un proceso que permite escoger determinados atributos o partes de información de una o más credenciales para generar una especie de Credencial sintetizada en una Presentación a partir de estas que siga siendo verificable y que permita entregar exactamente la información necesaria para el RP y nada más que no necesite. Este concepto de *Presentation Definition* se basa en un estándar de la *Digital Identity Foundation* o *DIF* [16]. Una Presentation Definition permite que el RP pueda definir la información que necesita obtener del End-User para poder solicitársela a su wallet y que esta pueda determinar qué información de toda la que contiene es la que debe presentar. Precisamente por eso se le denomina Presentation Definition, porque define qué información debe contener la presentación del usuario al RP.

Esta especificación introduce un nuevo token adicional al ID Token de OAuth2 que ya se ha tratado en apartados anteriores. El nuevo token se denomina *VP Token*, Este contiene Verifiable Presentations con las credenciales que se hayan elegido desde la wallet y que coincidan con las que han sido solicitadas por el RP correspondiente. Se devuelve junto al ID Token o en solitario en caso de que solo se pretenda utilizar para intercambio de información

mediante credenciales y no para autenticación/autorización. Por tanto en la Authentication Response se devuelve tanto el ID Token como el VP Token, como respuesta a la Authentication Request en la que el RP ha enviado una presentation definition para solicitar determinadas Credenciales. Junto al VP Token también se entrega una `presentation_submission`, también definida por la DIF en el mismo trabajo que la Presentation Definition [16], que contiene información relativa al formato y estructura que sea necesaria para procesar el propio VP Token y, por tanto, no pueda ir directamente en el mismo.

El flujo combinando SIOPv2 y OIDC4VP sería el que se muestra en la Figura 3.6.

3.8. *OpenID for Verifiable Credentials*

A la combinación de SIOPv2 con OIDC4VCI y OIDC4VP se le conoce como OIDVC, u OpenID for Verifiable Credentials. Como se puede deducir a partir de las secciones anteriores, con la solución SIOPv2 combinada con OIDC4VCI y OIDC4VP se consigue aislar las llamadas RP-Issuer a la hora de la identificación del usuario. Esto evita que el OP sea capaz de trazar la actividad de los usuarios que utilizan los servicios de un determinado RP que, en el caso de OIDC, confía plenamente en este OP. En este caso aunque quisiera no podría confiar excesivamente en el OP ya que es el propio usuario y, si así fuera, de hecho el usuario saldría beneficiado.

Como se comentó en secciones anteriores es necesario establecer de alguna manera la confianza entre Issuer-End-User y RP-Issuer. La forma más básica de confianza puede ser que cada usuario decida libremente en qué Issuers confía y en qué RP confía, al igual que el RP, que también decidirá qué Credenciales son fiables para sus procesos y cuáles no. Esta forma de confianza es muy democrática y correcta pero se empieza a tambalear cuando los RP deciden admitir únicamente las Credenciales de un Issuer concreto o algún otro mecanismo para corromperlo a su favor. En muchos procesos de contratación digital actuales dentro del *European Single Market* es de obligado cumplimiento ceñirse al reglamento para la identificación electrónica, conocido como *eIDAS*, que establece una serie de procedimientos a ejecutar para asegurar la confianza en el entorno de la contratación digital. Con este nuevo modelo de Identidad Digital se espera que se apruebe la nueva normativa *eIDAS 2* que vendrá en forma de enmienda a la normativa eIDAS (EU No 910/2014). El objetivo de esta es doble, arreglar la anterior normativa que se ha observado escasa en muchos ámbitos e incluir nuevos procedimientos que permitan establecer un nuevo

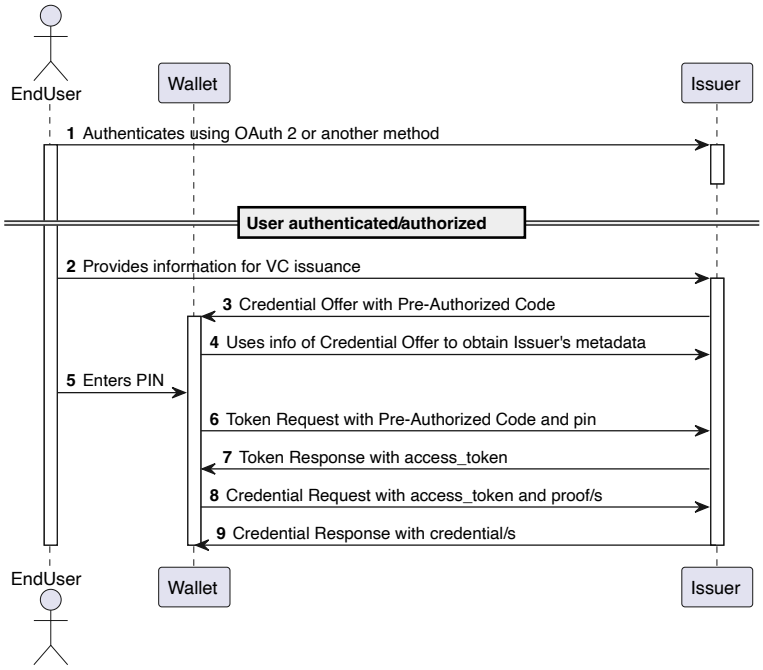


Figura 3.5: Flujo del protocolo OIDC4VCI [13]. Imagen de autoría propia.

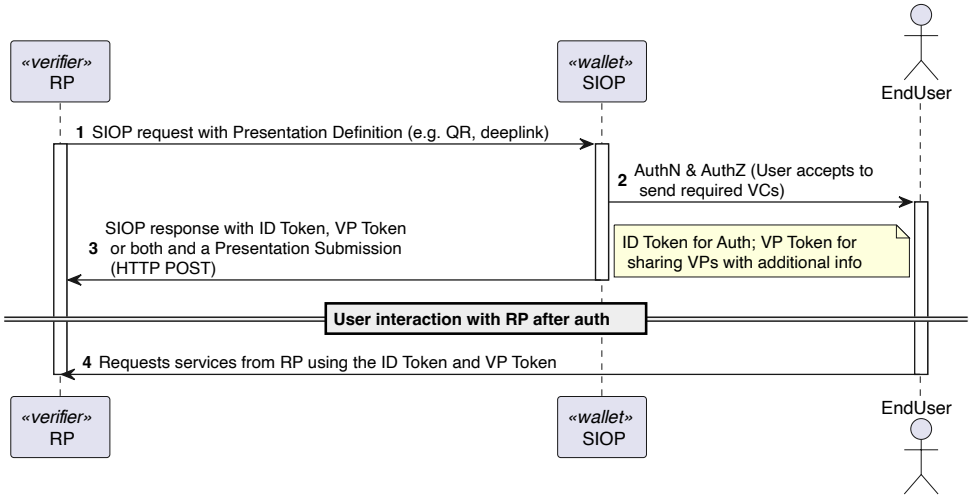


Figura 3.6: SIOPv2 combinado con OIDC4VP [14]. Imagen de autoría propia.

marco de confianza o *Trust Framework* que regule la confianza en las interacciones bajo la nueva Identidad Digital Autosoberana (SSI).

3.9. Normativa eIDAS

La normativa eIDAS [17] de la Unión Europea define los métodos y procedimientos necesarios para permitir interacciones seguras entre empresas y entre empresas y usuarios particulares. Busca proteger los intereses de ambas partes al realizar sus interacciones en el mundo digital. Para ello define los siguientes conceptos clave:

- *Firma electrónica*. Habilita una forma reconocida legalmente para firmar datos electrónicos como persona física.
- *Sello electrónico*. Habilita una forma reconocida legalmente para firmar datos electrónicos como persona jurídica.
- *Sello de tiempo electrónico*. Habilita una forma de firmar asegurando que la firma se produjo en una fecha concreta reconocida legalmente.
- *Documentos electrónicos*. Habilita formas de firmar electrónicamente distintos documentos digitales reconocidas legalmente.
- *Servicios de entrega electrónica certificada*. Habilita formas de garantizar el reconocimiento de la recepción de notificaciones electrónicas amparadas por la legalidad.
- *Servicios de certificación para autenticación de sitios web* (*QWACS*). Se permite autenticar las identidades de los sitios web para asegurar que el usuario puede confiar en que representan a la empresa que dicen representar.

Los requisitos que se establece para todos estos estándares bajo eIDAS son:

- Que su uso pueda ligarse de manera unívoca al firmante.
- Que pueda identificarse al firmante.

- Que las firmas realizadas en todos estos procesos sean realizadas con un dispositivo que esté en posesión únicamente del firmante, con mayor o menor nivel de confianza dependiendo del grado de seguridad que pueda tenerse de que esto se esté cumpliendo.

La firma o sello electrónico puede ser de varios tipos:

- *Simple*: Si no se ajusta a *AdES* (*Advanced Electronic Signature*). AdES (Advanced Electronic Signature) es una firma electrónica que permite identificar al firmante y detectar cambios en los datos firmados, garantizando integridad y vinculación exclusiva al firmante. Solo se considera como tal si se cumplen los requisitos funcionales y legales definidos en la normativa eIDAS para AdES.

- *Avanzada*: Si es realizada conforme a AdES.

- *Cualificada*: Equivalente a la firma/sello manuscrita. Se basa en la firma avanzada pero añade requisitos técnicos y legales referentes a una vinculación más estricta entre el firmante y la firma realizada.

Cabe destacar que la firma o sello cualificado tiene la ventaja de que se presume el origen fidedigno de la información firmada y, por tanto, invierte la carga de la prueba hacia la parte que pone en duda su veracidad en caso de procedimiento judicial. El sello de tiempo cualificado tiene la ventaja de que se presume que se firmó exactamente en la fecha y en el momento que aparecen reflejados, invirtiendo la carga de la prueba al igual que en el caso de la firma/sello cualificado.

A los proveedores de cualquiera de los servicios anteriormente citados, que se denominan servicios de confianza o *Trust Services*, se les denomina *Trust Services Providers*. Si proveen servicios cualificados pasan a llamarse *Qualified Trust Services Providers*.

Además la normativa eIDAS regula el uso de los medios de identificación electrónica nacionales, tales como el DNIe español, haciendo de obligado cumplimiento su admisión como método de identificación equivalente a la presentación física del mismo en todas las administraciones públicas de los estados miembro.

3.10. Normativa eIDAS 2

Es la evolución de la actual normativa eIDAS [18]. Actualmente se encuentra en proceso de discusión por la Comisión Europea. Pretende incorporar como nuevo servicio de confianza las *QEAA* o *Qualified Electronic Attribute Attestations*, que puede traducirse como testimonios de atributos electrónicos cualificados. La definición de este nuevo servicio encaja conceptualmente con lo que consigue el concepto de Credenciales Verificables que se ha expuesto a lo largo de este trabajo, que es poder expresar aserciones sobre propiedades o atributos de uno o más sujetos con mayor o menor grado de confianza.

Por otro lado, aunque relacionado con lo anterior, se establece la obligación de los Estados Miembro a proveer a sus ciudadanos de una Cartera de Identidad Digital Europea, o *European Digital Identity Wallet (EUDI Wallet)*, que sirva como medio de identificación electrónica. Esto no necesariamente implica que esta wallet vaya a ser la identidad de los ciudadanos, tal y como lo es el DNI, solo implica que se les podrá identificar con ella en lugar de con el DNI o DNIe. Para que esto sea posible se almacenarán Datos de Identificación Personal, o *Person Identification Data (PID)*, además de QEAA en dicha wallet. Los PID son como los QEAA pero con semántica ligeramente distinta ya que permiten la identificación unívoca del portador de la cartera. También se permitirá la firma electrónica usando la wallet mediante sus *QES*, también conocidos como *Certificados Digitales Cualificados*. El Certificado Digital es el medio para ejercer la firma electrónica como tal.

En la propuesta legislativa hay muchos más cambios pero esta sección se ha centrado en los más relevantes para este caso concreto, que es establecer un modelo teórico para implantar la Identidad Digital Autosoberana.

3.10.1. *Architecture and Reference Framework* (ARF)

El documento Architecture and Reference Framework [19], que está siendo elaborado por el grupo de expertos eIDAS de la Comisión Europea. Este establece los requisitos y algunos de los posibles casos de uso, tanto de la EUDI Wallet como de todo el ecosistema necesario para mantenerla en funcionamiento. En siguientes evoluciones irá profundizando más en la arquitectura del ecosistema y se acercará más a la implementación en el sentido de que se propondrán los estándares a utilizar. La intención es ser estándar respecto a este ecosistema y utilizar la EUDI Wallet como ”caja negra” en el caso concreto de la situación propuesta en este trabajo, pero de momento

no se definen en el documento. No se va a profundizar en los requisitos específicos de este ecosistema ya que se partirá de requisitos específicos para el caso concreto propuesto, aunque se incorporarán algunos de ellos, que serán mencionados en su caso. Para los QEAA es interesante sobretodo el Trust Model, del que tampoco se dan muchos detalles todavía.

3.10.2. EBSI

EBSI es la *European Blockchain Services Infrastructure* y se considera el brazo implementador de la Comisión Europea en cuanto al ecosistema de la EUDI Wallet y la cartera en sí. Es una iniciativa conjunta de la Comisión con la *EBP* o *European Blockchain Partnership* por lo que sus elecciones en cuanto a implementación son interesantes, ya que cabe esperar que encajen con el ARF y eIDAS 2 así que en este trabajo se ha buscado el máximo alineamiento con EBSI todo lo posible para asegurar la compatibilidad con este otro modelo que se prevee que sea el estándar en Europa en un futuro cercano. EBSI sigue todos los estándares cubiertos hasta ahora y aporta alguna característica extra que se utilizará en este trabajo.

3.10.2.1. DID

En EBSI se usan dos tipos de DID:

- Un primer tipo cuyo material criptográfico, su DID Document, se escribe en la red blockchain de EBSI.
- Un segundo tipo cuyo DID Document no se escribe en la blockchain y simplemente se reconstruye a partir de material criptográfico intercambiado del usuario hacia la parte que debe comprobar una firma hecha con el material criptográfico asociado al DID. A este se le denomina como *peer to peer*.

Es necesario hacer uso de estos dos métodos diferenciados ya que la blockchain es pública y no se pueden publicar datos personales, ni siquiera encriptados, que es como están consideradas las claves públicas asociadas a los DID de personas físicas. Es ilegal ya que infringiría la *GDPR* europea. Por ello para personas físicas se usa el segundo DID y para personas jurídicas el primero, aunque ambos tienen el mismo "prefijo", aunque distinta longitud y funcionamiento.

El DID del primer tipo descrito requiere acceso a través de API Key a un API de la Comisión, que de momento es privado.

3.10.2.2. Modelo de datos

EBSI define los modelos de datos para las Credenciales Verificables de ciertos casos de uso concretos que pretende estandarizar pero también cambia ligeramente el modelo de datos de lo que considera como Credencial Verificable, que según EBSI y de acuerdo con el texto legal de eIDAS2 pasa a llamarse *Verifiable Attestation*. Usaremos estos términos de manera intercambiable. También redefine el de la Verifiable Presentation, aunque se mantiene la terminología en este caso. En todo caso estos modelos de datos siguen siendo acordes al modelo del W3C.

Específicos para determinados casos de uso pero bastante generales son también el *Verifiable ID for Natural Person* y el *Verifiable ID for Legal Person*. El primero es básicamente lo que se entiende como PID en eIDAS 2 y el segundo sería la identificación de personas jurídicas. Es especialmente interesante el primero para el caso concreto de este trabajo.

3.10.2.3. Formatos de firmado de Credenciales

El formato de firma más común para Credenciales, al que se hace referencia como ejemplo en el modelo del W3C, es mediante LD-Proofs, que es la forma de firmar JSON-LD. Si se modelan las Credenciales como JWT, cosa que el modelo permite, entonces se firmarían con JWS. EBSI va un paso más allá y propone la firma *JSON Advanced Electronic Signatures (JAdES)* para Credenciales emitidas por entidades legales, que son prácticamente todas. Una firma JAdES es una firma avanzada mediante Certificado Digital, en este caso usando Sello de empresa. Es JWS pero con cabeceras adicionales por lo que se considera una extensión del mismo. Para realizarla es posible utilizar la herramienta *DSS* o *Digital Signature Standard* de la UE. Si se hace con sello cualificado entonces ya estaríamos ante los QEAA de los que habla eIDAS 2, aunque también se puede usar solamente firma avanzada y tener un *EAA*. Un EAA es un QEAA pero sin firma cualificada. En el Anexo B se puede ver un ejemplo de un Verifiable ID firmado con JAdES.

3.10.2.4. Trust Model

El concepto de marco de confianza o modelo de confianza aplica tanto a la Identidad Digital actual como a la Identidad Digital Autosoberana basada en Credenciales. Por ello la UE también dispone de un marco de confianza concreto para eIDAS que se basa en emitir jerárquicamente certificados digitales desde entidades a nivel europeo hasta entidades nacionales que habilitan a los *Proveedores de Confianza Cualificados* o *Qualified Trust Services Provider (QTSP)* en el ámbito nacional y estos ya se encargan de emitir certificados y sellos a personas físicas y jurídicas. Estos proveedores de confianza cualificados son los organismos habilitados a nivel nacional para poder emitir certificados cualificados de firma o sello. No se explicará con detalle este marco de confianza si no que se centrará el trabajo en el nuevo marco de confianza que pretende establecerse para eIDAS 2 y la nueva Identidad Digital, que es el que realmente es de interés. Si que es relevante conocer que se hace referencia a este marco de confianza jerárquico como *PKI* o *Public Key Infrastructure*.

En la Figura 3.7 se puede observar un esquema de las posibles formas de gobernanza para un marco de confianza. Estas son las siguientes:

- *Centralizado*, si hay una entidad concreta que regula las entidades que son de confianza y las que no. Para verificar si una entidad es o no de confianza es necesario contactar con esta entidad que las regula.
- *Federado*, si hay un conjunto de entidades que regulan las entidades que son de confianza y las que no. Para verificar si una entidad es o no de confianza se pueden consultar listas expuestas por cada entidad reguladora que especifican las entidades que son de confianza. A esto se le denomina como una *Trusted List*. Este es el modelo que se sigue en eIDAS. En la actualidad suelen ser ficheros XML en la web del ministerio correspondiente de cada Estado Miembro.
- *Distribuido*, si es como una gobernanza Federada pero con mucho mayor grado de distribución en cuanto al sistema que habilita y permite verificar a los proveedores de servicios de confianza.

El Trust Model distribuido es el que se pretende utilizar para eIDAS 2. Tratará de imitar las relaciones jerárquicas de confianza del modelo federado actual pero ayudándose sobre una red blockchain. A esta nueva estructura de PKI se le viene denominando como *DPKI* o *Decentralized Public Key Infrastructure*. A grandes rasgos consistirá

en tener Trusted Lists almacenadas en la red blockchain de EBSI, mantenida por uno o más nodos de cada Estado de la UE, almacenar las claves públicas de los Issuers de Electronic Attestation of Attributes, las estructuras o *schemas* de EAA que se establezcan por la entidad pertinente y, casi con toda seguridad, las revocaciones de las EAA. Los schemas detallan la estructura de un tipo determinado de EAA. Llevar una lista de revocaciones en blockchain es algo controvertido ya que se tiene miedo de que, al ser la blockchain una red pública, si se almacenan las revocaciones de las credenciales de los ciudadanos podría darse el caso de que puedan correlacionarse para inferir la actividad del usuario y poder trazarla. De esta manera podrá comprobarse que el usuario presenta Credenciales emitidas por un tercero de suficiente confianza (Cualificado o no) como para dar seguridad al caso de uso correspondiente, que podría ir de modelar simples entradas a un concierto, que si se falsifican no ocasionan daños elevados, hasta datos de salud, que podrían contribuir al fraude de medicamentos si no hay seguridad de que el emisor de esas Credenciales esté habilitado para hacerlo.

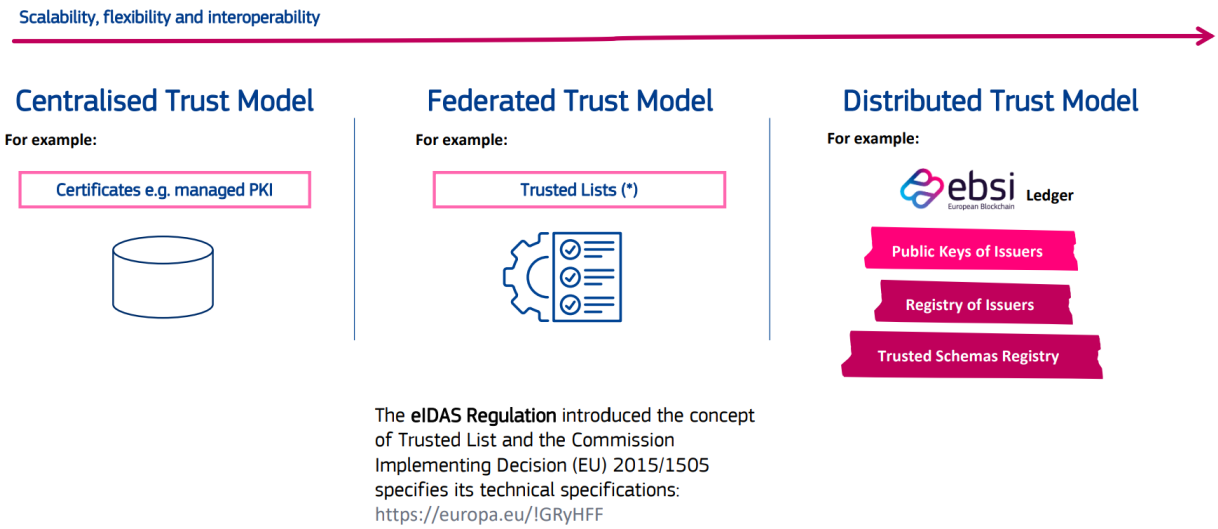


Figura 3.7: Diferentes formas de gobernanza de un Trust Model. Imagen tomada de [20].

3.10.3. Riesgos y desafíos del modelo de Identidad Digital Autosoberana

Aunque el modelo de Identidad Digital Autosoberana (SSI) presenta claras ventajas en cuanto a privacidad, descentralización y control del usuario sobre sus datos, también introduce una serie de desafíos y riesgos que deben ser tenidos en cuenta para su adopción generalizada y segura:

1. **Correlación de actividad**: la reutilización de un mismo *DID* en múltiples interacciones permite a terceros

correlacionar dichas actividades y, por tanto, reconstruir parte del perfil del usuario.

2. **Gestión de la privacidad por parte del usuario**: el modelo SSI traslada la responsabilidad de gestionar la identidad y la privacidad al propio usuario. Esto implica que debe entender qué credenciales posee, cuándo presentarlas y ante quién. En la práctica, si la experiencia de usuario no está cuidadosamente diseñada, esto puede generar errores humanos, cesión indebida de datos o incluso falta de adopción.

3. **Pérdida o compromiso de la *wallet***: a diferencia de los sistemas centralizados en los que el proveedor de identidad puede asistir en caso de pérdida de credenciales, en un modelo autosoberano la pérdida del acceso a la *wallet* puede significar la pérdida de acceso a múltiples servicios o identidades digitales. Además, si una *wallet* es comprometida (por ejemplo, por malware), el atacante podría utilizar las credenciales presentables del usuario de forma ilegítima.

4. **Confianza en los emisores y el marco de gobernanza**: aunque el modelo propone un marco descentralizado, la confianza en la veracidad de las credenciales depende de los emisores y del marco de confianza adoptado (*Trust Framework*). Un ecosistema sin mecanismos de reputación, auditoría o control puede derivar en la emisión de credenciales falsas o poco fiables.

Capítulo 4

Requisitos

En este capítulo se cubre la especificación de requisitos de un sistema que permita a la Universidad de Valladolid emitir sus titulaciones de manera electrónica con la misma validez que en su formato físico haciendo uso de credenciales verificables. Además, el sistema permite la presentación a un tercero de las titulaciones emitidas para acceder a puestos de trabajo con requisitos sobre determinadas titulaciones.

4.1. Introducción

Este sistema cumple con todos los requisitos necesarios que establece la propuesta para el futuro reglamento europeo para la Identidad Digital eIDAS 2 [19] y las indicaciones tanto del ARF como de los desarrollos de EBSI, aunque no será posible en todos los casos ya que algunos de los requisitos requieren clarificación en futuras revisiones y otros son demasiado ambiciosos como para pretender abarcarlos en este trabajo.

El sistema requiere que cada estudiante posea una wallet que cumpla con los requisitos necesarios para ser considerada una EUDI Wallet según el reglamento europeo citado anteriormente. De manera muy simplificada estaríamos hablando de un sistema software instalado en un dispositivo inteligente que permita el almacenamiento de datos acerca del portador de la wallet en forma de Credenciales Verificablesque serán emitidas por diferentes Issuers, como serán en este caso las emitidas por la UVa, con la seguridad criptográfica que corresponda. Se

supondrá que dicha wallet es una aplicación Android totalmente nativa o híbrida. De esta manera el estudiante podría presentar una Credencial correspondientes a una titulación de la UVa que almacena en su wallet para poder justificar su titulación ante una empresa que se lo requiera para poder acceder a cierto puesto de trabajo, de una forma más cómoda, menos burocrática, más segura ante falsificaciones, más auditable y cumpliendo con el futuro reglamento eIDAS 2. El sistema de la wallet se tomará como un sistema ya desarrollado y no se tratará y el modelado de las titulaciones será una versión simplificada de las titulaciones reales. También se supondrá a la UVa como la entidad firmante o Issuer de las titulaciones digitales emitidas sin interponer a un tercero de confianza.

4.2. Descripción detallada del sistema

Para entender el sistema desarrollado en esta sección se incluye una descripción en lenguaje natural, con una serie de ejemplos de uso de este.

Un **Estudiante** ha terminado el Grado en Filosofía y ahora necesita acreditar su titulación ante una empresa para postularse a cierto puesto de trabajo. Para ello se dirige al **sistema de emisión y presentación de titulaciones digitales**, ya que le permite recibir una versión digital de su titulación de manera telemática para poder acreditar que la ha finalizado con un alto grado de confianza de que esa información es verídica. Como ya se ha aprobado la nueva propuesta para Identidad Digital de la Unión Europea [19], el Estudiante conecta su **EUDI Wallet**, que tiene instalada en su *smartphone*, en la que almacena **credenciales** y **datos de identificación personal**, o **PID**, que se consideran credenciales, al Sistema **para poder identificarse** en él. Al identificarse de esta forma se evita tener que recordar datos de acceso de ningún tipo más allá de los datos de acceso o biometría necesarios para hacer uso de la wallet. Esto también evita que haya un tercero que maneje y presente por él los datos requeridos por el Sistema, ya que él mismo maneja la wallet que ha sido emitida por terceros de confianza establecidos por el Estado Español que permiten que la mayoría de datos del usuario permanezcan de manera local en el dispositivo y sean gestionados por el propio usuario. El sistema solicita al Estudiante que presente un PID formado por su **nombre**, **apellidos**, **fecha de nacimiento**, **dirección** y **nif**. Estos datos permiten identificarle de manera unívoca y son necesarios para disponer de información básica sobre él. Este PID se utilizará tanto a la hora de registrarse en el sistema como a la hora de identificarse ante él. Este modo de proceder no es lo mejor ya que una vez utilizado en el registro se podría presentar información menos sensible en posteriores identificaciones, adecuando el sistema

de mejor manera a la minimización de datos que se exige en la normativa GDPR. En principio deberían utilizarse los datos estrictamente necesarios para identificaciones posteriores al registro. Este PID en concreto se espera que haya sido precargado en la wallet en el momento de la emisión de esta, como se contempla en la propuesta de la Unión Europea. Como el Estudiante puede tener diferentes Credenciales es necesario que el Sistema le indique qué espera recibir exactamente para que la *wallet* pueda entregar lo que se le solicita. Para ello el sistema realiza una **Solicitud de Presentación** en la que especifica la estructura y semántica del PID que espera obtener. Para poder asegurar una **autenticación mutua**, como se requiere en la propuesta europea, el Sistema firma digitalmente esta Solicitud de Presentación antes de enviarla a la wallet del usuario, de manera que se puede verificar la integridad de la solicitud y que realmente está comunicándose con el Sistema y no con un impostor de este. Esto permite a la wallet autenticar al Sistema. La *wallet* encapsula el PID solicitado, la wallet lo encapsula en una **Presentación de Credenciales**, la cual se encarga de firmar para que el Sistema pueda a su vez autenticar a la wallet y asegurar la integridad de la información intercambiada. Una vez recibido el PID también se comprueba la firma de este, que tiene que ser del emisor. En este caso, como los PID se emiten por **entidades certificadas** por la UE, también se comprobará que el emisor del PID se encuentra en la lista de emisores cualificados en el **Trusted Registry** o Registro de Confianza europeo. Este registro se encuentra en la *blockchain* de EBSI y se puede acceder a él mediante un API concreto.

Una vez identificado el **Usuario**, queda comprobar si este está autorizado a acceder al sistema, cosa que solo ocurrirá si este es un estudiante de la UVa. Para ello se realiza una conexión con el sistema de la UVa para comprobar si un usuario con los datos contenidos en el PID, que lo identifican de manera unívoca, existe como estudiante en el **Sistema de la UVa**. Finalmente el **Estudiante** está autenticado y autorizado en el Sistema y podría solicitar la emisión de la titulación digital de cualquier grado que haya finalizado.

Una vez identificado, el Estudiante **solicita la emisión de una credencial** que acredite que dispone de la titulación correspondiente. Para permitir una retrocompatibilidad con el sistema tradicional de emisión de títulos se exigirá que se haya hecho la emisión de la titulación de la manera tradicional antes de que pueda solicitarse su versión electrónica. El sistema le muestra una lista de todas aquellas titulaciones que ha terminado y cuya titulación haya sido ya solicitada físicamente para que seleccione la titulación que desea emitir en formato digital. Para poder mostrar esta lista el sistema ha de conectar con el sistema de la UVa para realizar las comprobaciones mencionadas anteriormente. Una vez que el Estudiante ha escogido el Grado en Filosofía, el Sistema solicita al sistema de la UVa

información adicional sobre la titulación que aparecerá en la titulación digital emitida. Una vez generada la credencial de titulación digital se firma en nombre de la UVa y se envía a la *wallet*, donde se verificará la autenticidad de la firma y la integridad de la credencial. Tras finalizar este proceso el usuario dispondrá de la credencial en su *wallet* para poder presentarla de manera autónoma, sin depender de tercerlos, en aquellas entidades que así se lo requieran, bien para obtener autenticación y autorización o bien simplemente como método para aportar información requerida.

Una vez que tiene la titulación digital en la wallet, el Estudiante accede a la web de la empresa en la que pretende postularse para un puesto que requiere el Grado en Filosofía. El **sistema de RRHH de la empresa** delega en el sistema desarrollado el intercambio de datos necesario para que el Estudiante pueda aplicar al puesto ofrecido, que en este caso consistirá en llegar a comprobar que el Estudiante dispone de la titulación correspondiente. El sistema desarrollado solicita a la **EUDI Wallet** del Estudiante que le presente una **titulación digital de la UVa** por medio de una **Solicitud de Presentación** en la que especifica su estructura y su semántica. Para asegurar una **autenticación mutua** entre la *wallet* y el sistema desarrollado el segundo firma la Solicitud para que la *wallet* pueda comprobar la autenticidad y la integridad de la Solicitud. Para enviar la titulación digital, la *wallet* la encapsula en una **Presentación de Credenciales** firmada para que el sistema pueda verificar la autenticidad e integridad de dicha presentación, con lo que quedaría autenticada la propia wallet. También se comprobará que la firma de la titulación corresponda a la UVa. Al final de este proceso el sistema puede estar seguro de que el Estudiante dispone de la titulación correspondiente y por tanto se autoriza su acceso para poder postularse al puesto ofrecido.

Si debido a un error informático se han emitido titulaciones digitales erróneas y se quiere que queden invalidadas a efectos prácticos. Para ello el **Encargado de la UVa** comunica al sistema el **id** de la credencial que desea marcar como revocada y se lo comunica al **Trusted Registry** europeo a través de su API para que tanto la EUDI Wallet como otros terceros a los que se pueda presentar la titulación digital, como la empresa de RRHH en este caso, sean conscientes de que la credencial ha sido revocada y actúen en consecuencia.

4.3. Requisitos

4.3.1. Requisitos funcionales

En base a la descripción detallada del sistema en la sección 4.2 y el marco teórico fijado en el capítulo 3, se establecen los siguientes requisitos funcionales:

RF-1 El Estudiante podrá autenticarse ante el sistema haciendo uso de los Datos de Identificación Personal (PID) residentes en su EUDI Wallet.

RF-2 El sistema deberá contactar con el Trusted Registry europeo a la hora de autenticar al usuario mediante su PID para poder comprobar que este fue emitido por un emisor de confianza

RF-3 El Estudiante podrá solicitar telemáticamente a la Universidad de Valladolid la emisión en formato digital de la titulación académica finalizada que considere.

RF-4 El Encargado de emisión de titulaciones de la UVa deberá autenticarse ante el sistema haciendo uso de los Datos de Identificación Personal (PID) residentes en su EUDI Wallet.

RF-5 El Encargado de emisión de titulaciones de la UVa podrá revocar las titulaciones emitidas.

RF-6 El sistema deberá contactar con el Trusted Registry europeo para reflejar la revocación de una titulación digital por parte de un Encargado.

RF-7 Las titulaciones emitidas digitalmente deberán ser firmadas electrónicamente en nombre de la UVa.

RF-8 El sistema de RRHH de la empresa deberá poder solicitar al sistema que se verifique la posesión de la titulación digital que corresponda para permitir al portador de una wallet con titulaciones académicas digitales de la UVa aplicar a determinados puestos vacantes.

RF-9 El sistema deberá contactar con el Trusted Registry europeo para comprobar si una titulación ha sido revocada a la hora de verificar la posesión de la misma por un usuario.

4.3.2. Requisitos no funcionales

En base a la descripción detallada del sistema en la sección 4.2 y el marco teórico fijado en el capítulo 3, se establecen los siguientes requisitos no funcionales:

RNF-1 La funcionalidad del sistema dedicada a la emisión de titulaciones digitales deberá concretarse en una aplicación web.

RNF-2 La funcionalidad del sistema dedicada a la verificación de la posesión de titulaciones deberá concretarse en una API REST a disposición de terceros.

RNF-3 El sistema deberá enviar las Credenciales Verificables que representan a las titulaciones digitales a la wallet en formato JWT, encapsulando un JSON-LD que modela la credencial propiamente dicha.

RNF-4 El sistema deberá seguir el modelo de Verifiable Credentials del W3C.

RNF-5 El sistema deberá firmar las titulaciones digitales mediante JAdES para asegurar su autenticidad e integridad.

RNF-6 Las titulaciones deberán ir firmadas digitalmente por el Estudiante que las solicite.

RNF-7 El firmado digital de las titulaciones digitales deberá ser suficientemente fuerte en términos criptográficos como para asegurar su autenticidad y su integridad, determinado por su inclusión en el catálogo SOG-IS.

RNF-8 El sistema deberá realizar una autenticación mutua con la wallet del usuario antes de dar por válido el proceso de emisión de una titulación.

RNF-9 El sistema deberá realizar una autenticación mutua con la wallet del usuario antes de dar por válido el proceso de presentación de una titulación.

RNF-10 Es necesario que el sistema utilice un protocolo de autenticación mutua con la wallet común al del resto de sistemas que interactúan con la wallet.

RNF-11 El sistema utilizará OIDC4VCI como protocolo estándar para la emisión de las titulaciones digitales.

RNF-12 El sistema utilizará OIDC4VP como protocolo estándar para la presentación de las titulaciones a un tercero.

RNF-13 El sistema utilizará SIOPv2 como protocolo estándar para la autenticación/autorización mediante la titulación en sistemas de terceros.

4.3.3. Requisitos de información

En base a la descripción detallada del sistema en la sección 4.2 y el marco teórico fijado en el capítulo 3, se establecen los siguientes requisitos de información:

RI-1 El sistema deberá almacenar un histórico de las titulaciones emitidas en formato digital. Concretamente se almacenará el código de la titulación, si es de grado o máster, nombre de la titulación, nota media del expediente, fecha y hora de emisión, información relativa al registro físico de la titulación y la legislación vigente sobre el que se ampara la misma.

RI-2 El sistema deberá almacenar los datos de identificación personal (PID) de cada Estudiante. Concretamente se almacenará nombre, apellidos, dirección, fecha de nacimiento y DNI.

RI-3 El sistema deberá almacenar un histórico de las revocaciones de titulaciones realizadas por el Encargado de la UVa junto con su fecha y hora de realización.

4.4. Casos de uso

4.4.1. Modelo de casos de uso

En las siguientes secciones se especifican los actores y casos de uso del sistema identificados en el Modelo de Casos de Uso. La Figura 4.1 muestra el diagrama de casos de uso del sistema.

4.4.1.1. Actores principales

Los actores principales del sistema son:

- *Usuario*. Todo usuario del sistema relacionado con la emisión de titulaciones digitales, esto es, estudiantes y encargados de la UVa.
- *Estudiante*. Usuario del sistema que se acredita como estudiante de la UVa. Puede solicitar la emisión de sus titulaciones en formato digital.

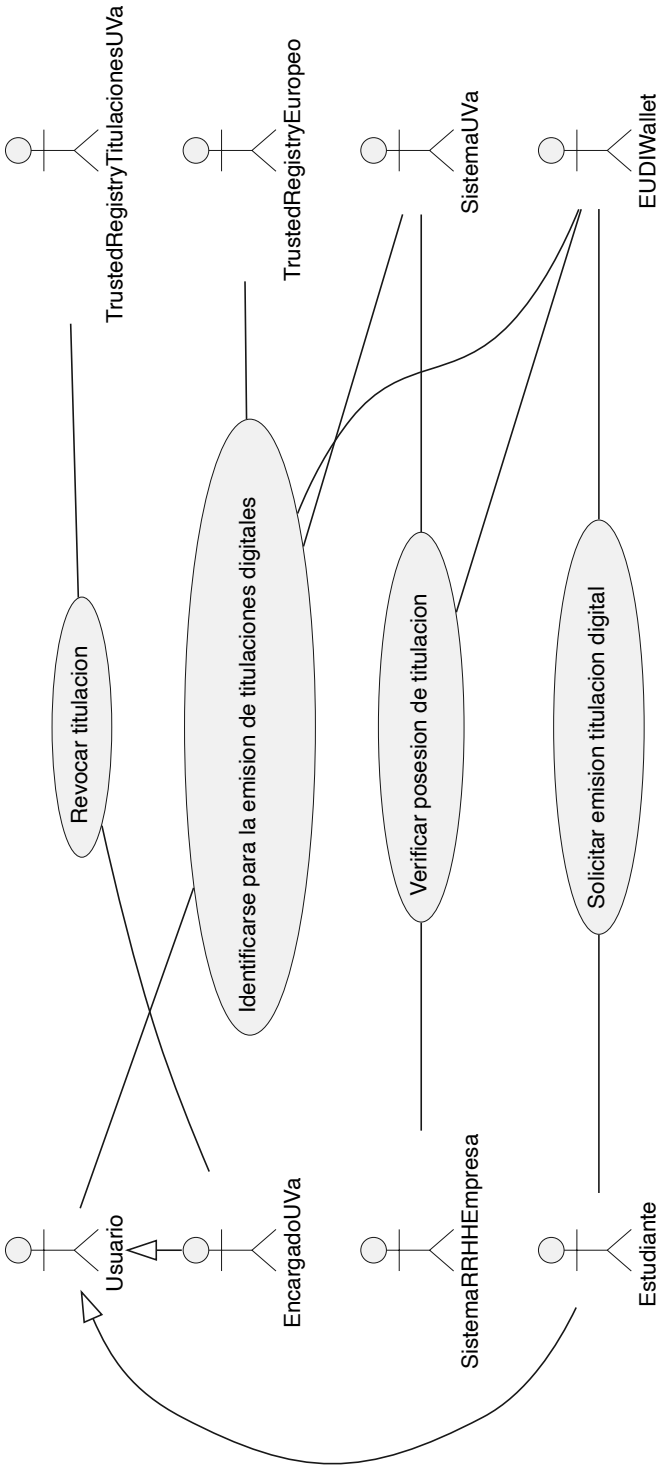


Figura 4.1: Diagrama de casos de uso.

- *EncargadoUVa*. Usuario del sistema que se acredita como encargado de la UVa. Puede revocar las titulaciones digitales emitidas.
- *SistemaRRHHEmpresa*. Sistema de RRHH de una empresa que haya delegado en el Sistema la verificación de posesión de una titulación o titulaciones en base a la cual permitir acceso autorizado a su sistema.

4.4.1.2. Actores secundarios

Los actores secundarios del sistema son:

- *TrustedRegistryEuropeo*. Sistema de EBSI que permite consultar la lista de emisores de confianza y si una titulación ha sido revocada o no.
- *SistemaUVa*. Sistema actual de la UVa del que se solicitará información tanto para autenticar al Estudiante como para la emisión de sus titulaciones digitales.
- *EUDIWallet*. Wallet de Identidad Digital conforme al reglamento eIDAS2. Almacenará Credenciales del usuario, como la titulación digital, y permitirá su presentación a terceros.
- *TrustedRegistryTitulacionesUVa*. Sistema en la blockchain de Ethereum que permite marcar titulaciones digitales emitidas por la UVa como revocadas.

4.4.2. Especificación de Casos de Uso

En este apartado se realizará la especificación de cada Caso de Uso del sistema. En la Tabla 4.1 se encuentra la descripción del CU Identificarse para la emisión de titulaciones digitales, en la Tabla 4.2 la del CU Solicitar emisión titulación digital, en la Tabla 4.3 la del CU Revocar titulación digital y en la Tabla 4.4 la del CU Verificar posesión de titulación digital. Esta especificación de casos de uso sigue la norma IEEE 29148 en la medida de lo posible, aunque se ha adaptado a las necesidades del sistema.

Caso de Uso:	**CU1. Identificarse para la emisión de titulaciones digitales**
Descripción:	El Sistema deberá comportarse tal y como se describe cuando el Usuario solicite identificarse ante él
Actores:	Usuario, TrustedRegistryEuropeo, SistemaUVa, EUDIWallet
Pre-condiciones:	El usuario posee una EUDI Wallet emitida por el organismo competente.
Post-condiciones:	El usuario queda identificado en el sistema.

Secuencia normal

1. El Usuario solicita identificarse ante el sistema.
2. El sistema conecta con la EUDI Wallet del Usuario para solicitar un PID compuesto por nombre, apellidos, fecha de nacimiento, dirección y nif.
3. El sistema recibe el PID.
4. El sistema comprueba que el PID ha sido firmado por el Usuario especificado en sus metadatos.
5. El sistema comprueba la integridad del PID recibido haciendo uso de sus metadatos.
6. El sistema registra el PID del Usuario.
7. El sistema conecta con el Trusted Registry europeo para comprobar que el PID ha sido firmado por un emisor que figura como emisor de confianza.
8. El sistema conecta con el Sistema de la UVa para comprobar que el usuario autenticado coincide con un Estudiante y/o un Encargado de la UVa.
9. El sistema consulta al Usuario si quiere identificarse como Estudiante o como Encargado de la UVa.
10. El Usuario elige identificarse como Estudiante.

Secuencia alternativa
5.a. El sistema informa de que no se ha podido comprobar que el usuario que dice estar presentando el PID sea quien realmente lo está presentando y el Caso de Uso finaliza quedando sin efecto.
6.a. El sistema informa de que el PID recibido no es íntegro y el Caso de Uso finaliza quedando sin efecto.
7.a. El sistema informa de que el PID no ha sido emitido por un emisor que figure en el Trusted Registry europeo y el Caso de Uso finaliza quedando sin efecto.
9.a.1. El sistema identifica al usuario como Estudiante.
9.a.2. El Caso de Uso finaliza correctamente.
9.b.1. El sistema identifica al usuario como Encargado de la UVa.
9.b.2. El Caso de Uso finaliza correctamente.
9.c. El sistema no identifica al usuario ni como Estudiante ni como Encargado y el Caso de Uso finaliza quedando sin efecto.
10.a. El Usuario elige identificarse como Encargado de la UVa.

Tabla 4.1: Especificación del CU Identificarse para la emisión de titulaciones digitales

Caso de Uso:	**CU2. Solicitar Emisión titulación digital**
Descripción:	El Sistema deberá comportarse tal y como se describe cuando el Usuario solicite la emisión de una titulación. digital
Actores:	Estudiante, EUDI Wallet.
Pre-condiciones:	El usuario se encuentra identificado en el sistema.
Post-condiciones:	El usuario tiene la titulación digital deseada en su EUDI Wallet.

Secuencia normal

1. El Estudiante solicita la emisión de una titulación digital.
2. El sistema conecta con el Sistema UVa para obtener las titulaciones finalizadas por el Estudiante cuyo certificado de finalización ya haya sido emitido físicamente.
3. El sistema presenta al usuario las titulaciones.
4. El Estudiante indica la titulación de la que quiere solicitar una emisión digital como prueba de que la ha finalizado.
5. El sistema utiliza la información de la titulación escogida para generar una titulación digital firmada en nombre de la UVa.
6. El sistema registra la titulación digital generada junto con su firma.
7. El sistema conecta con la EUDI Wallet del usuario para enviar la titulación digital generada.

Secuencia alternativa

2.a. El sistema informa de que ha habido un error interno que impide que el proceso continúe.

Tabla 4.2: Especificación del CU Solicitar emisión titulación digital

Caso de Uso:	**CU3. Revocar titulación digital**
Descripción:	El Sistema deberá comportarse tal y como se describe cuando el Encargado solicite la revocación de una titulación digital.
Actores:	Encargado, Trusted Registry Titulaciones UVa.
Pre-condiciones:	El usuario se encuentra identificado en el sistema como Encargado.
Post-condiciones:	La titulacion digital queda revocada, impidiéndose su uso.
Secuencia normal	

1. El encargado solicita la revocación de una titulación digital.
2. El sistema registra la revocación en el Trusted Registry Titulaciones UVa.
3. El sistema registra la credencial como revocada.

Tabla 4.3: Especificación del CU Revocar titulación digital

Caso de Uso:	**CU4. Verificar posesión de titulación**
Descripción:	El Sistema deberá comportarse tal y como se describe cuando el Sistema de RRHH de la empresa correspondiente solicite verificar la posesión de una titulación digital del Usuario aplicante.
Actores:	Sistema RRHH de la empresa, EUDI Wallet.
Pre-condiciones:	Ninguna.
Post-condiciones:	El sistema de RRHH de la empresa ha podido comprobar si el usuario correspondiente tiene una titulación digital de la UVa.

Secuencia normal

1. El Sistema de RRHH de la empresa solicita que se compruebe si el usuario que ha interactuado con sus sistemas dispone de cierta titulación digital de la UVa.
2. El sistema conecta con la EUDI Wallet del usuario solicitando que presente la titulación digital requerida.
3. El sistema recibe la titulación digital correspondiente.
4. El sistema comprueba que la titulación digital no figure como revocada en el Trusted Registry Titulaciones UVa.
5. El sistema comprueba los metadatos de la titulación digital para comprobar que ha sido firmada por el Estudiante que figura en los mismos.
6. El sistema comprueba la integridad de la titulación digital recibida haciendo uso de sus metadatos.
7. El sistema comprueba que la titulación digital ha sido firmada por la UVa.
8. El sistema envía al Sistema de RRHH de la empresa la titulación verificada.

Excepciones

4.a. El sistema informa de que la titulacion figura como revocada y el Caso de Uso finaliza quedando sin efecto.

5.a. El sistema informa de que no se ha podido comprobar que el usuario que aparece en los metadatos de la titulación digital sea quien realmente la ha presentado y el Caso de Uso finaliza quedando sin efecto.

6.a. El sistema informa de que la titulación digital recibida no es íntegra y el Caso de Uso finaliza quedando sin efecto.

7.a. El sistema informa de que la titulación digital no ha sido emitida por la UVa y el Caso de Uso finaliza quedando sin efecto.

8.a. El sistema notifica al Sistema de RRHH de la empresa que no se ha podido comprobar que el usuario disponga de la titulación digital requerida y el Caso de Uso finaliza quedando sin efecto.

Tabla 4.4: Especificación del CU Verificar posesión de titulación

Capítulo 5

Análisis

En este capítulo se se presenta el flujo de trabajo de Análisis todo el proceso de Análisis del sistema analizado en el capítulo 4. Se presentan los modelos de Dominio y Análisis.

5.1. Modelo del dominio

En la Figura 5.1 se muestra el Modelo de Dominio del sistema.

5.2. Modelo de Análisis

El diagrama de clases del Análisis puede observarse en la Figura 5.2. La firma del interesado sobre las titulaciones digitales emitidas desaparece ya que se considerará que la UVa se apoya en un QTSP al emitir sus titulaciones digitales para dotarlas de las mismas capacidades que si estas hubieran sido emitidas utilizando un sello de empresa o persona jurídica. Firmadas de esta forma se admite por defecto que el origen que se ha utilizado para obtener los datos acerca de la titulación correspondiente es correcto, lo cual es posible debido a que la UVa se puede mostrar ante el QTSP correspondiente como un organismo con acceso a la *fuente de verdad* de esos datos, ya que es información que genera la propia UVa. Esto por si solo no implica que la firma del interesado sea prescindible.

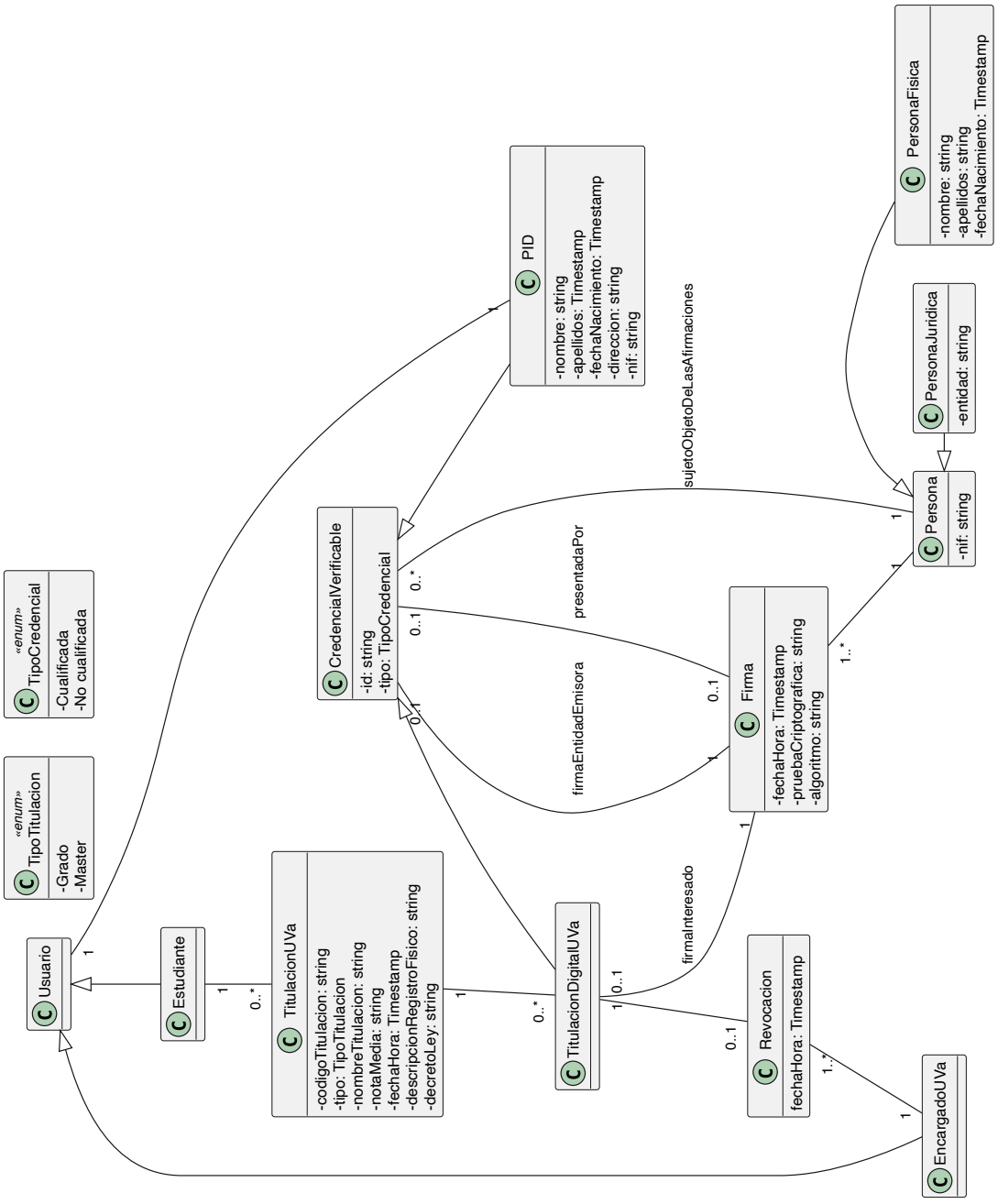


Figura 5.1: Modelo del dominio.

Como la titulación lleva asociada a la persona interesada, en este caso persona física, y esta está asociada a su vez con un GestorFirmaVerificacion que tiene su clave pública habiendo establecido la sesión del usuario bajo un LoA High se puede afirmar que la UVa afirma conocer que está interactuando con quien de verdad es el usuario con un nivel de confianza elevado y por tanto tiene capacidad de poder afirmar que el usuario está conforme con la emisión de la titulación y se presupone firmada por el mismo.

Respecto al modelo de dominio ha sido necesario añadir una nueva clase que encargue de emitir y verificar firmas sobre las Credenciales Verificables. Como la parte que verifica las firmas no necesariamente tiene que tener la clave privada de la entidad que las realiza hay un gestor específico para esa función, GestorFirmaVerificacion, mientras que GestorFirma puede realizar ambas acciones. También ha sido necesario introducir una clase que modele los datos necesarios para poder solicitar determinada Credencial Verificable a una EUDI Wallet, que es la clase SolicitudPresentacion. Otra clase que se ha introducido ha sido la de PresentacionCredenciales, para permitir entregar un conjunto de credenciales que pueda estar firmado por el Holder una única vez al ser esta una firma realizada de manera digital.

5.2.1. Especificación de las clases

En la Figura 5.3 se puede ver una visión detallada de la clase Usuario. En la 5.4 y 5.5 se puede ver la del Estudiante y EncargadoUVa respectivamente. En 5.6 se tiene la clase PID, en 5.7 la de la Titulacion Digital y en 5.8 la de la Credencial Verificable. En 5.9 y 5.10 se tiene la Solicitud de Presentación y la Presentación Verificable respectivamente. En 5.12 se tiene la clase GestorFirma y en 5.13 la GestorFirmaVerificación mientras que en 5.11 se tiene la clase Firma como tal. En 5.14 se tiene a la clase Persona, junto a la clase PersonaFisica en 5.16 y la de PersonaJuridica en 5.15. Por último en 5.17 se tiene la clase Revocacion y en 5.12 la clase GestorRevocacion.

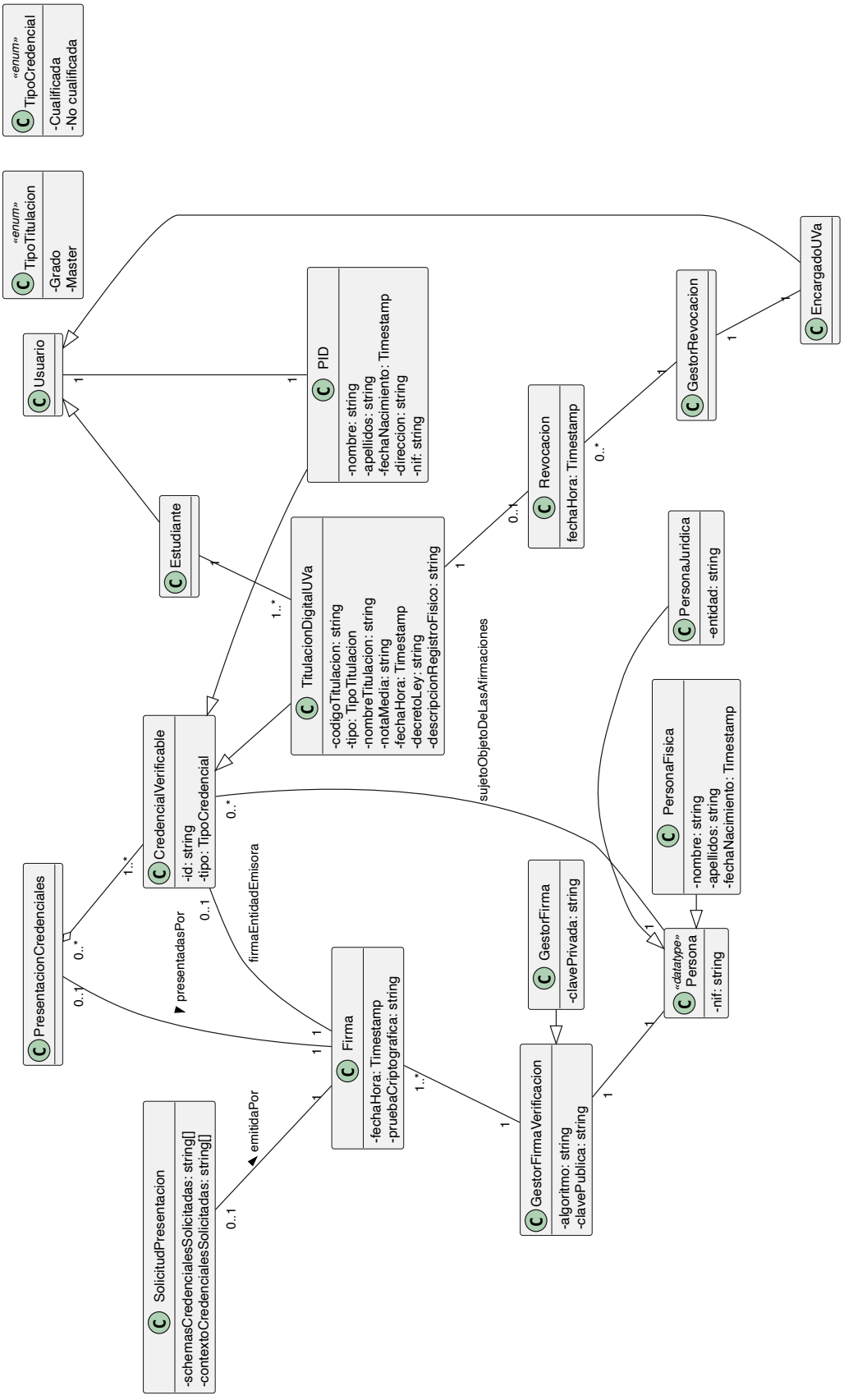


Figura 5.2: Diagrama de Clases del Análisis.

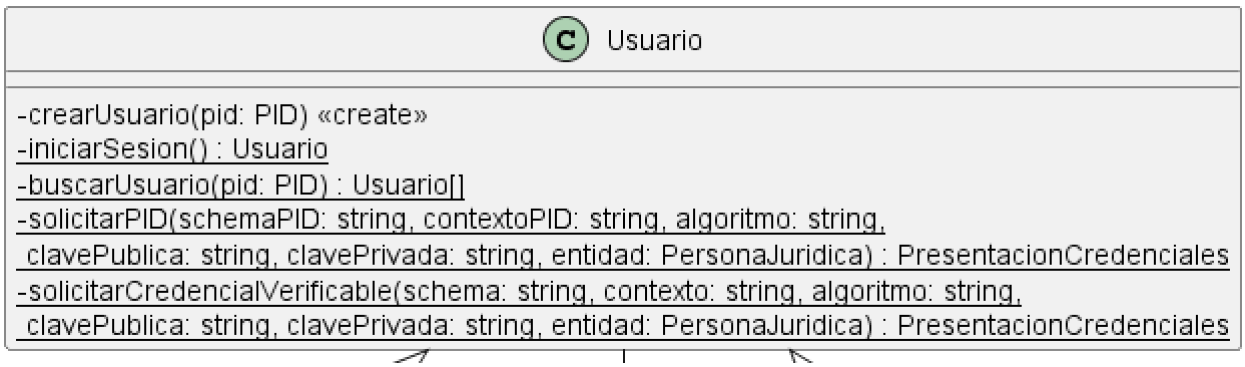


Figura 5.3: Detalle de la clase de Análisis Usuario.

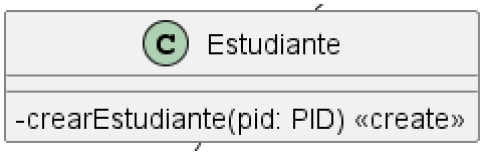


Figura 5.4: Detalle de la clase de Análisis Estudiante

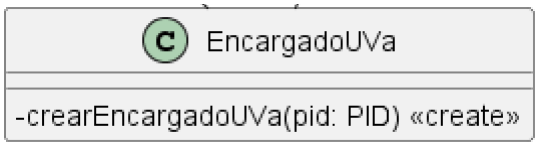


Figura 5.5: Detalle de la clase de Análisis EncargadoUVa.

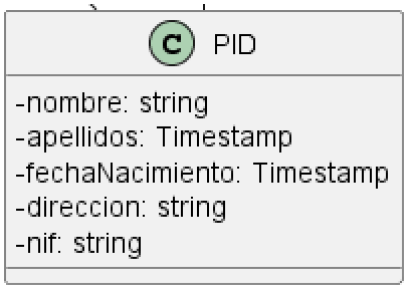


Figura 5.6: Detalle de la clase de Análisis PID.

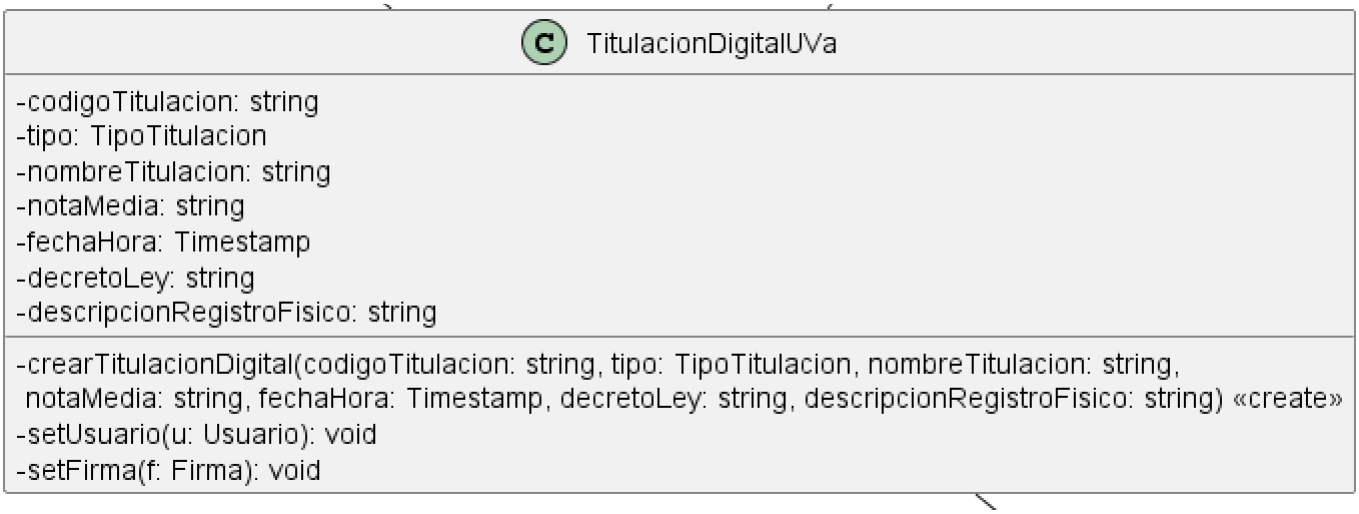


Figura 5.7: Detalle de la clase de Análisis TitulaciónDigitalUVa.

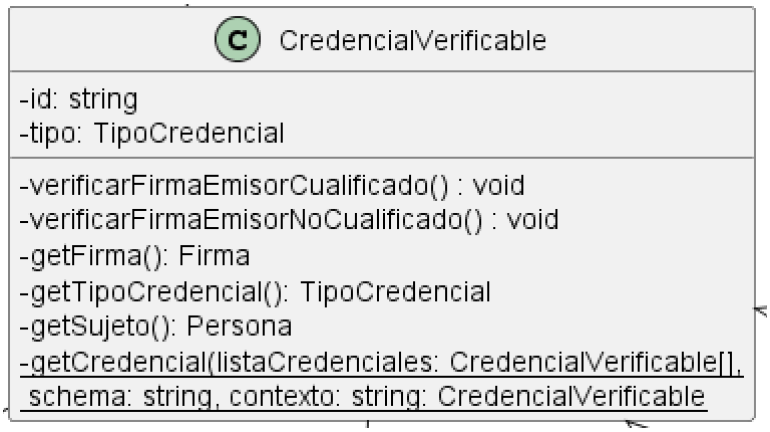


Figura 5.8: Detalle de la clase de Análisis CredencialVerificable.

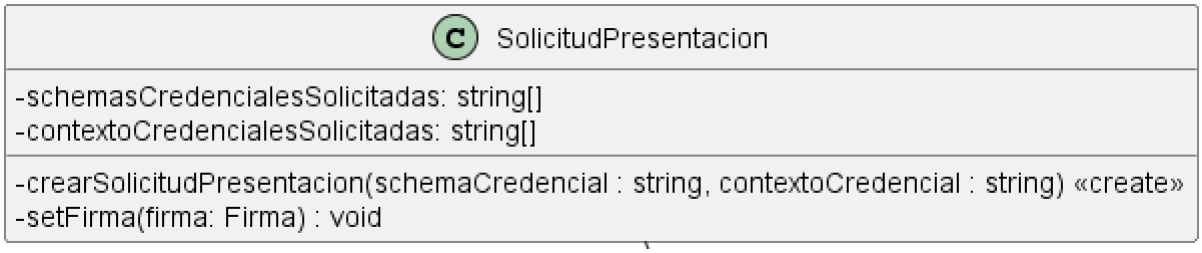


Figura 5.9: Detalle de la clase de Análisis SolicitudPresentacion.

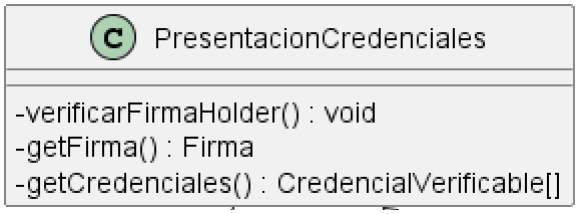


Figura 5.10: Detalle de la clase de Análisis PresentacionCredenciales.

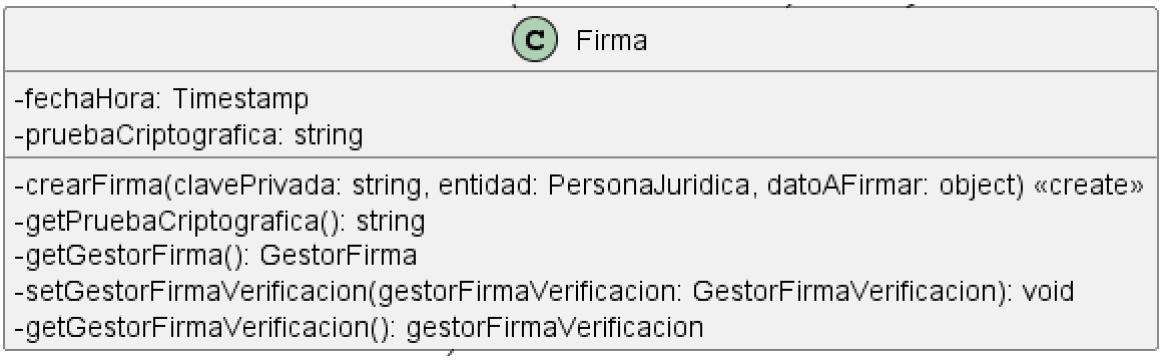


Figura 5.11: Detalle de la clase de Análisis Firma.

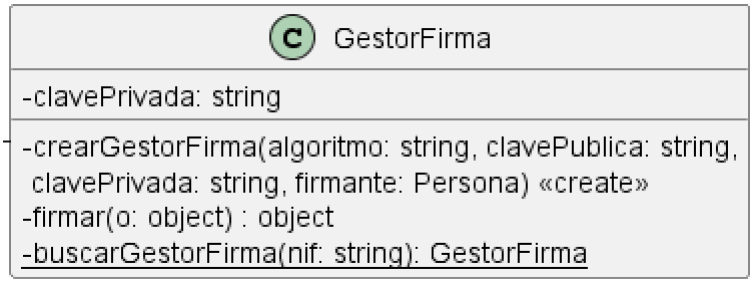


Figura 5.12: Detalle de la clase de Análisis GestorFirma.

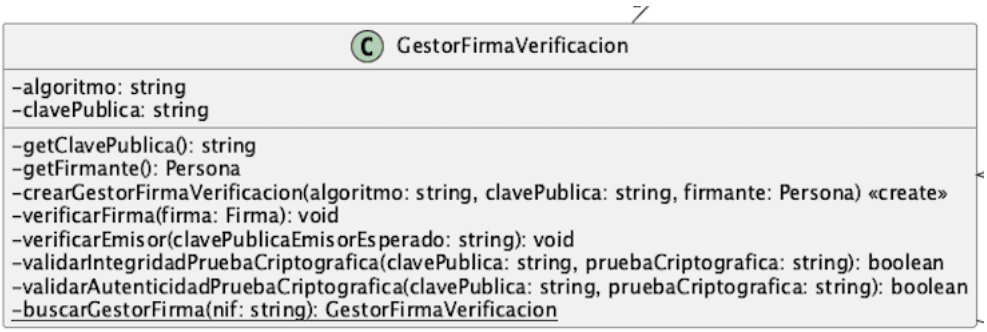


Figura 5.13: Detalle de la clase de Análisis GestorFirmaVerificacion.

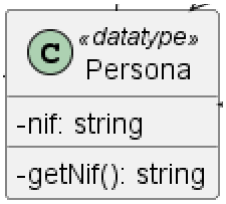


Figura 5.14: Detalle de la clase de Análisis Persona.

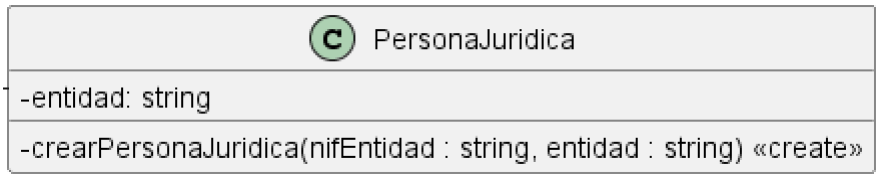


Figura 5.15: Detalle de la clase de Análisis PersonaJuridica.

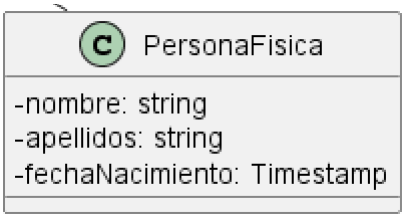


Figura 5.16: Detalle de la clase de Análisis PersonaFisica.

Figura 5.17: Detalle de la clase de Análisis Revocacion.

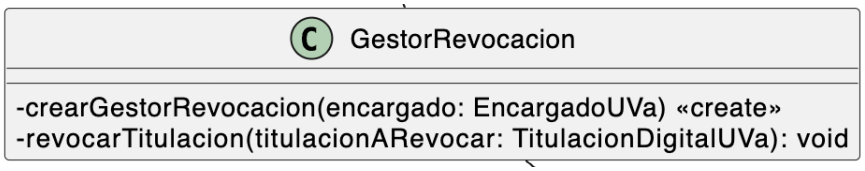


Figura 5.18: Detalle de la clase de Análisis GestorRevocacion.

5.3. Realización en Análisis de los Casos de Uso

En este apartado se presenta la Realización en Análisis de cada uno de los Casos de Uso del sistema.

5.3.1. CU1 Identificarse ante el Sistema de Emisión de Titulaciones Digitales

En la Figura 5.19 se puede observar el diagrama de secuencia de la identificación ante el sistema de emisión de titulaciones digitales. Como se puede observar hay llamadas hacia un actor secundario, situación que se repite en los diagramas de secuencia de varios de los CUs. En este caso se contacta con el sistema de la UVa para comprobar si un usuario es estudiante, encargado de la UVa o ambos. También se puede apreciar el uso de un fragmento *ref* para dividir el diagrama de secuencia y que pueda comprenderse mejor evitando que crezca en exceso. Nuevamente esto tiene lugar en varios de los diagramas presentados posteriormente. Para realizar esa identificación se realiza la solicitud del PID a la wallet del usuario, como se puede observar en la Figura 5.20. A la hora de solicitar el PID al usuario se realiza un proceso que es común para todo tipo de Credencial Verificable, como se puede ver en la Figura 5.21 La verificación de la firma de la presentación se encuentra en la Figura 5.22. La verificación de la firma del emisor del PID se encuentra en la Figura 5.23.

El PID contenido por la wallet desde el momento de su creación y firmado por una entidad cualificada según la Unión Europea se solicita a la hora de identificarse contra el sistema para poder autenticar al usuario. Esta autenticación será de un *Level Of Assurance* (*LoA*) high, el máximo nivel de confianza establecido por la UE, conseguido en la actualidad mediante, por ejemplo, el certificado electrónico del DNIe. Es posible este nivel de confianza máximo debido, en primer lugar, a la firma cualificada del PID que contiene la wallet y en segundo lugar a su almacenamiento seguro en el dispositivo como establece la UE en la propuesta en la que se está basando el sistema. Se verifica que la entidad firmante está cualificada por la UE haciendo una consulta al Trusted Registry europeo, que se menciona tanto en el ARF como en la propuesta eIDAS 2. Este Trusted Registry podrá ser una blockchain (DLT) u otra tecnología que cumpla los requisitos que establece la propuesta pero, ni entra dentro del sistema definido ni correspondería establecerlo en Análisis, se establecería en la fase de Diseño en todo caso. Por otro lado, al verificar la firma del PID se puede asegurar su integridad y autenticidad. La solicitud del PID se realiza mediante una Solicitud de Presentación, en la que se describe qué credenciales son exactamente las solicitadas y la wallet se encarga de interpretarlo y filtrar las credenciales del usuario de acuerdo a la descripción para posteriormente enviar únicamente las que cumplen con los requisitos establecidos. Se indica tanto la estructura del PID solicitado como el significado o semántica de esta.

Es importante destacar que la wallet entrega el PID al sistema encapsulado en lo que denominaremos una

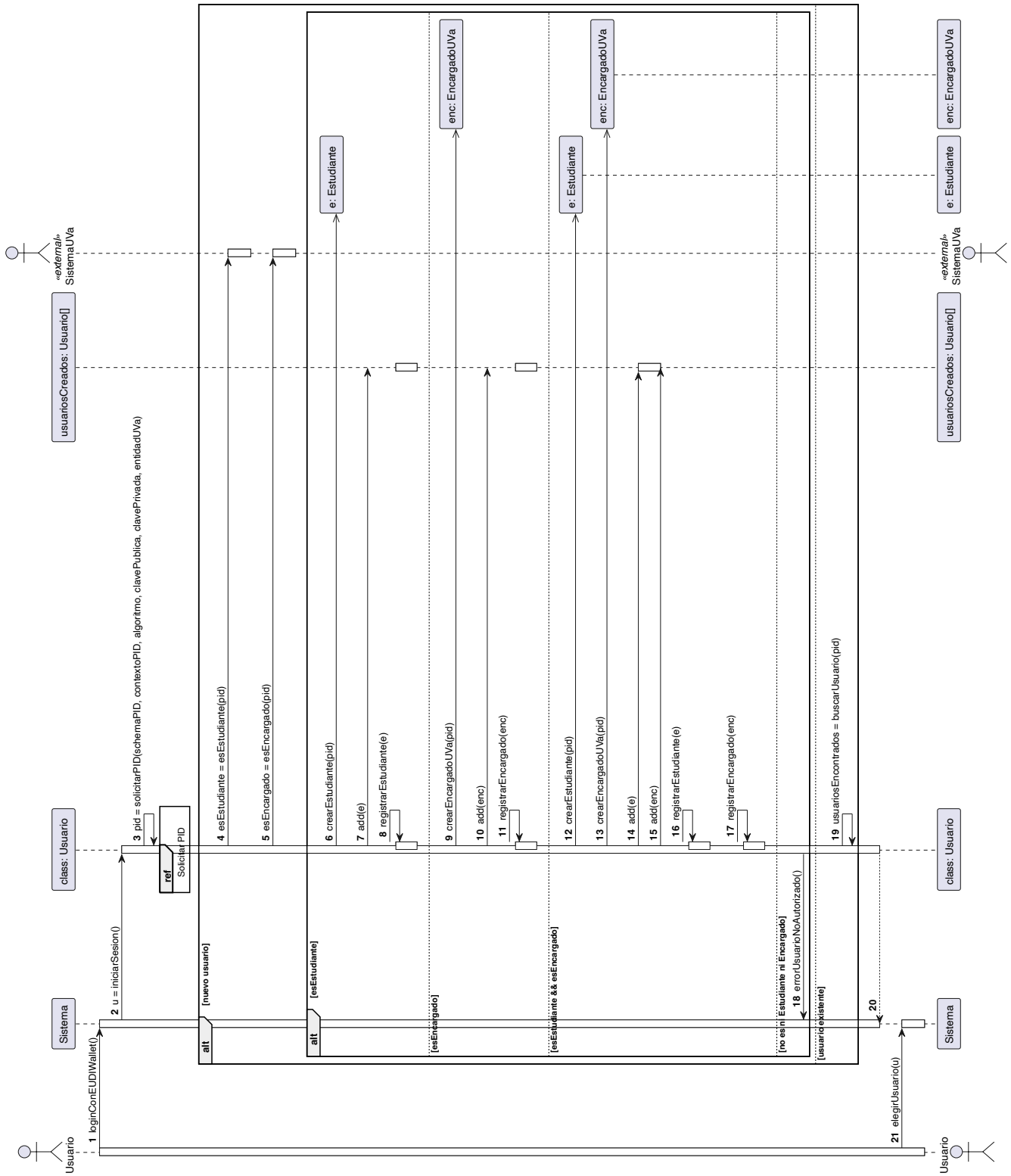


Figura 5.19: Diagrama de secuencia del CU Identificarse para la emisión de titulaciones digitales.

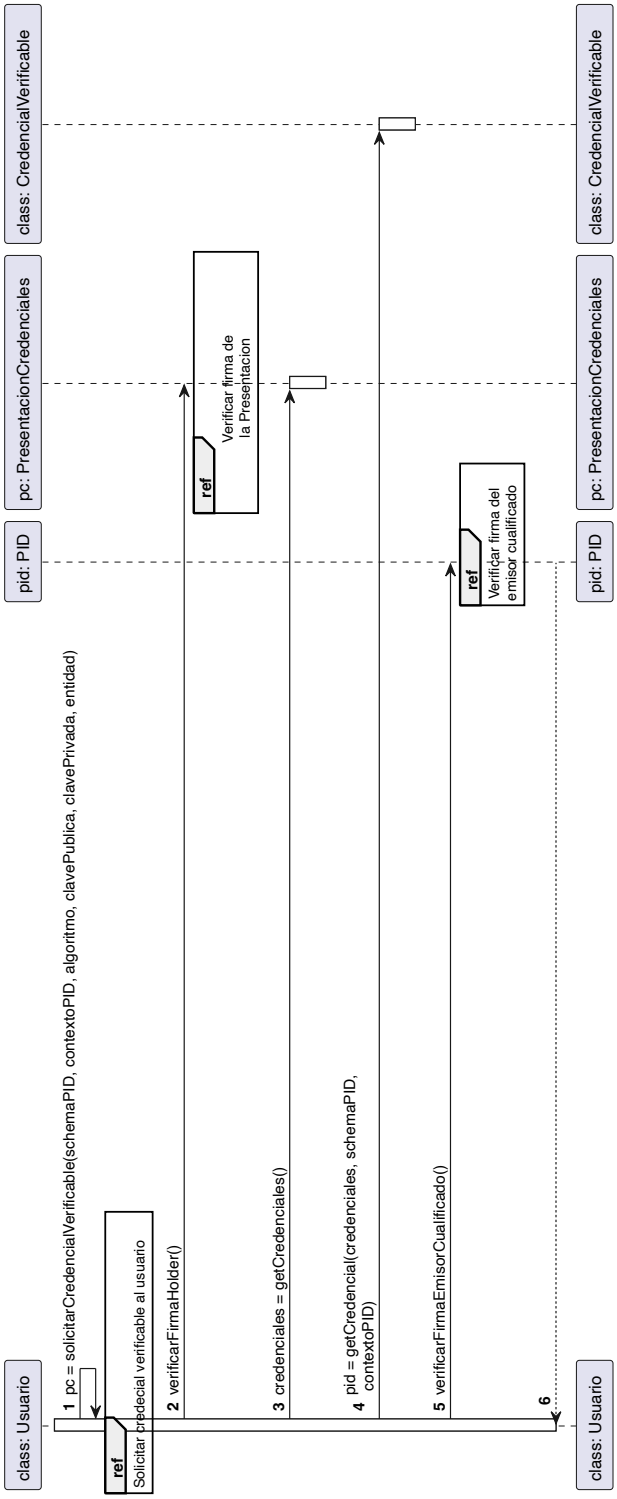


Figura 5.20: Diagrama de secuencia de la solicitud del PID a la wallet del usuario.

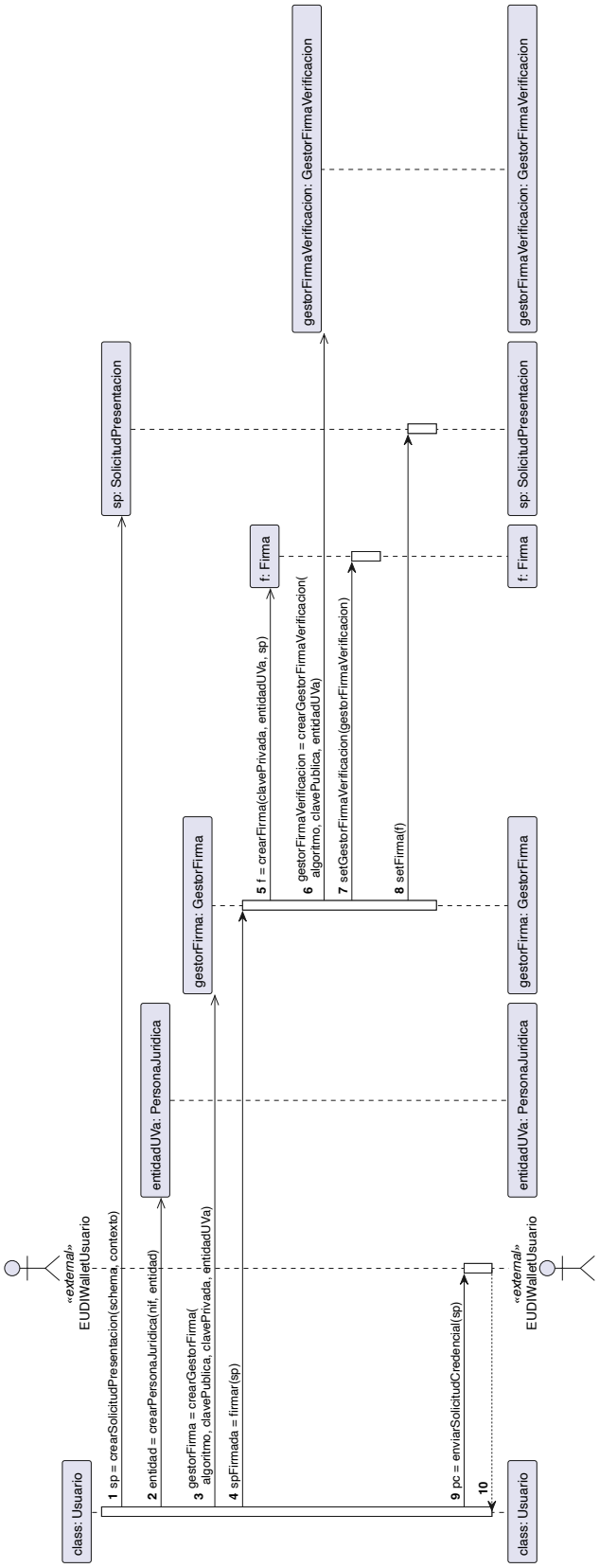


Figura 5.21: Diagrama de secuencia de la solicitud de Credenciales Verificables a la wallet del usuario.

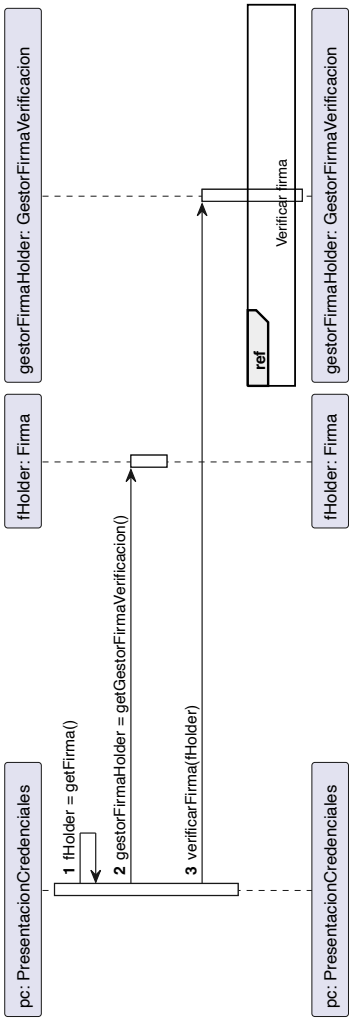


Figura 5.22: Diagrama de secuencia de la verificación de la firma de una Presentación Verificable.

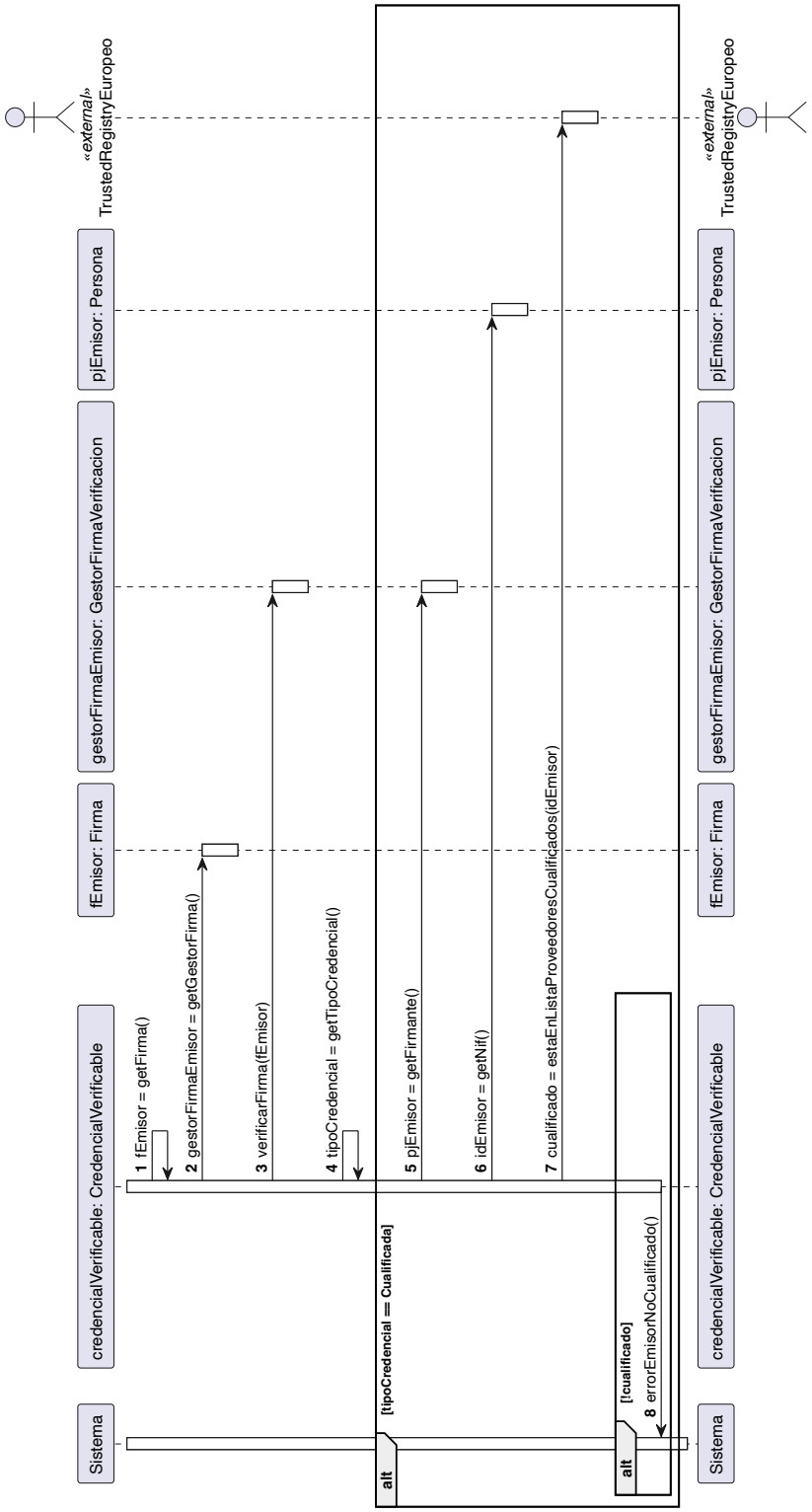


Figura 5.23: Diagrama de secuencia de la verificación de la firma del emisor Cualificado de una Credencial Verificable.

Presentación de Credenciales, que está firmada a su vez por el propietario de la wallet. Esto permite que se pueda autenticar al Holder de la wallet a la vez que la propia wallet autentica al sistema al verificar la firma que había realizado en la Solicitud de Presentación previamente enviada. Esto es a lo que el ARF europeo se refiere como **autenticación mutua** entre la wallet y el emisor de credenciales.

Tras verificar el PID recibido de la wallet se comprueba la autorización de acceso tanto al sistema de la UVa como al sistema de emisión de titulaciones del usuario identificado, comprobando si este es estudiante, encargado, ambas o ninguna de ellas. En base a esto se distingue si se está ante una identificación exitosa o no.

5.3.2. CU2 Solicitar Emisión Titulación Digital

En la Figura 5.24 se puede ver el diagrama de secuencia del Caso de Uso de Solicitar Emisión de la Titulación Digital. Cuando el Usuario solicita la emisión de una titulación digital a partir de una titulación de las que ha terminado y cuyo proceso de emisión física se ha completado se le muestra un listado de las titulaciones que cumplan esos dos requisitos. Tras emitir la titulación en formato digital se registra y se envía a la EUDI Wallet del usuario.

5.3.3. CU3 Revocar Titulación Digital

En la Figura 5.25 se puede ver el diagrama de secuencia de la revocacion de una titulacion digital.

5.3.4. CU4 Verificar Posesión de Titulación Digital

En la Figura 5.27 se puede ver el diagrama de secuencia del Caso de Uso de verificar posesión de la Titulación Digital. En la Figura 5.21 se puede ver el proceso de solicitud realizado para pedir la Titulación al usuario. En la Figura 5.22 se muestra la verificación de la firma de la presentación entregada por el usuario. La verificación de la firma del emisor de la Titulación Digital, esto es, la UVa, se encuentra en la Figura 5.28.

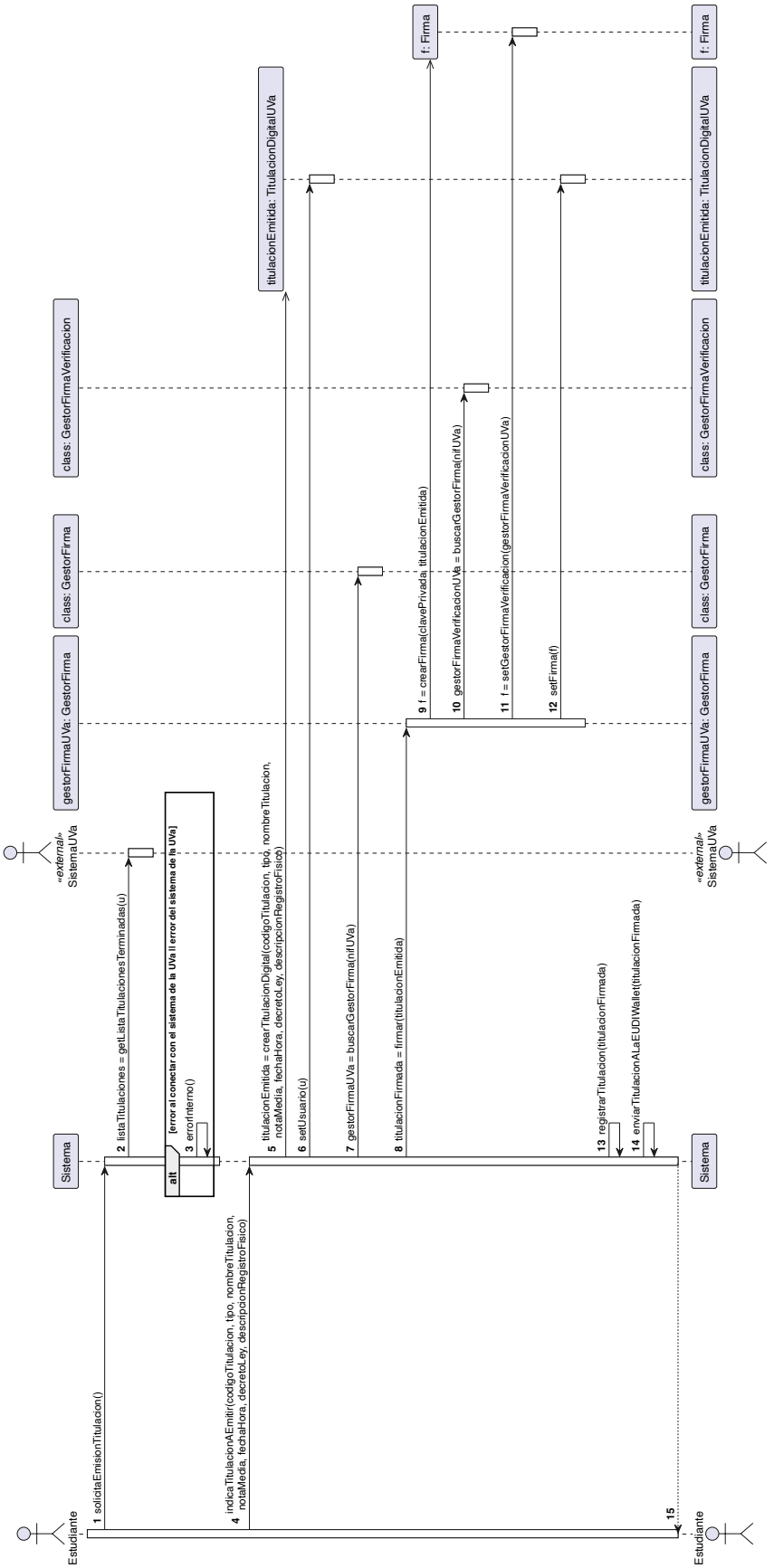


Figura 5.24: Diagrama de secuencia del CU solicitar emisión de la Titulación Digital.

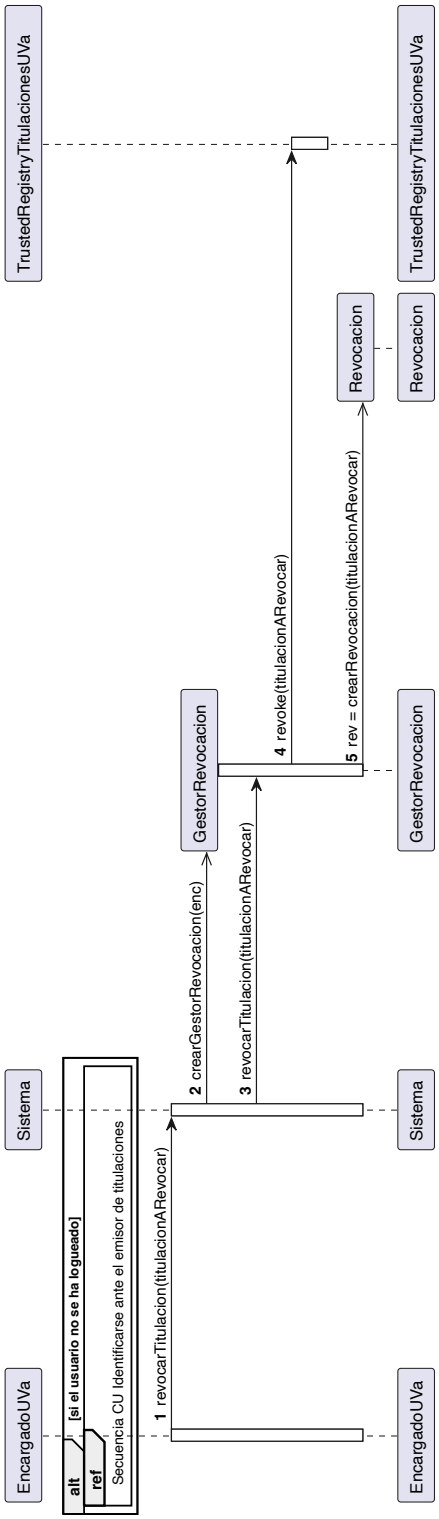


Figura 5.25: Diagrama de secuencia de la revocacion de una Titulación Digital.

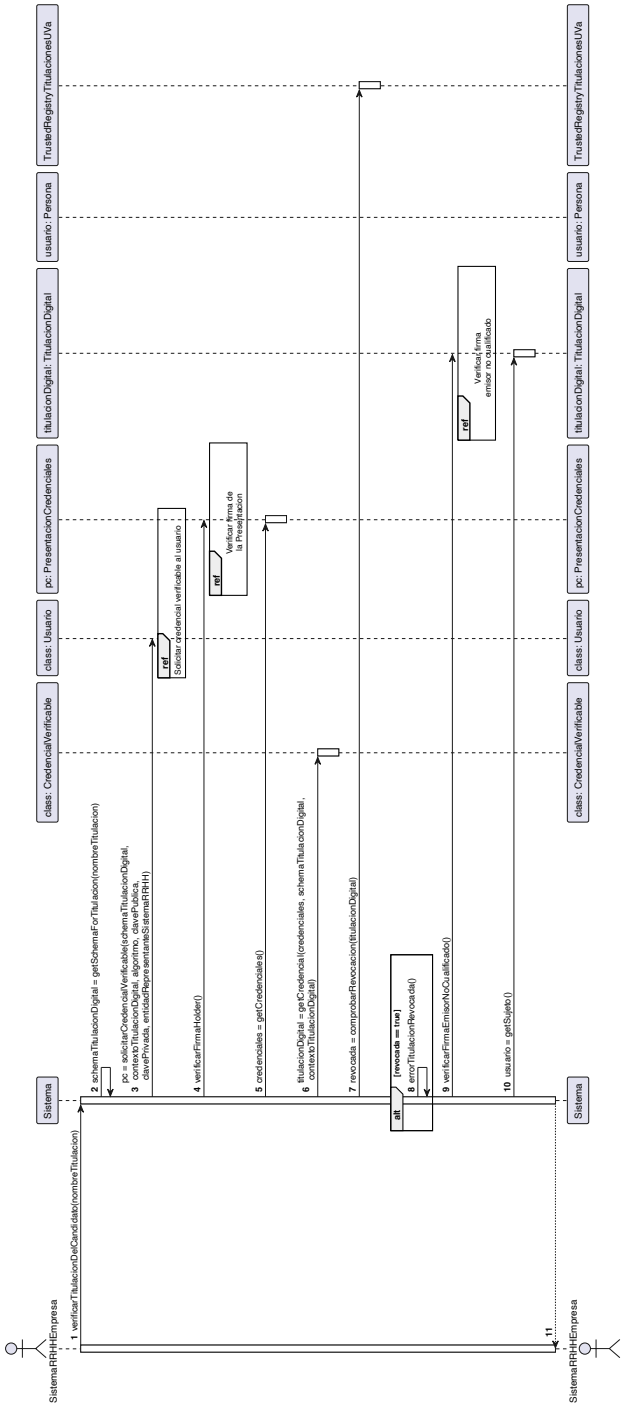


Figura 5.26: Diagrama de secuencia del CU Verificar Posesión de Titulación Digital.

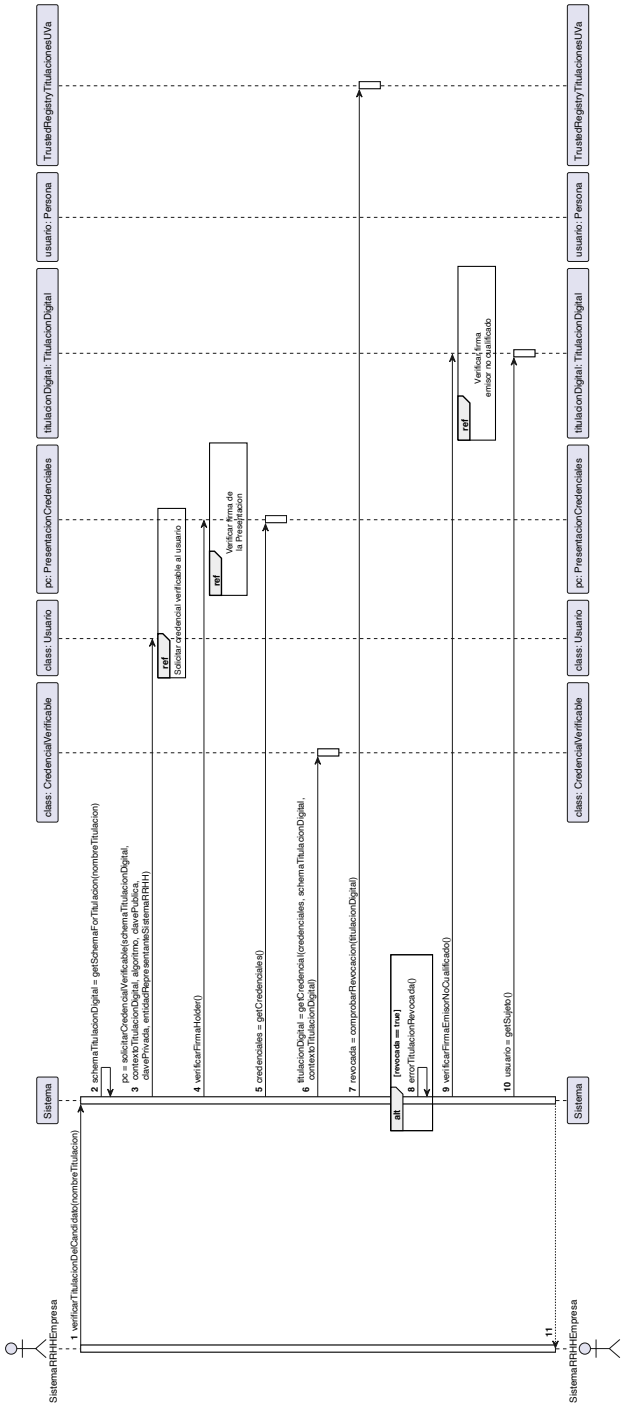


Figura 5.27: Diagrama de secuencia del CU Verificar Posesión de Titulación Digital.

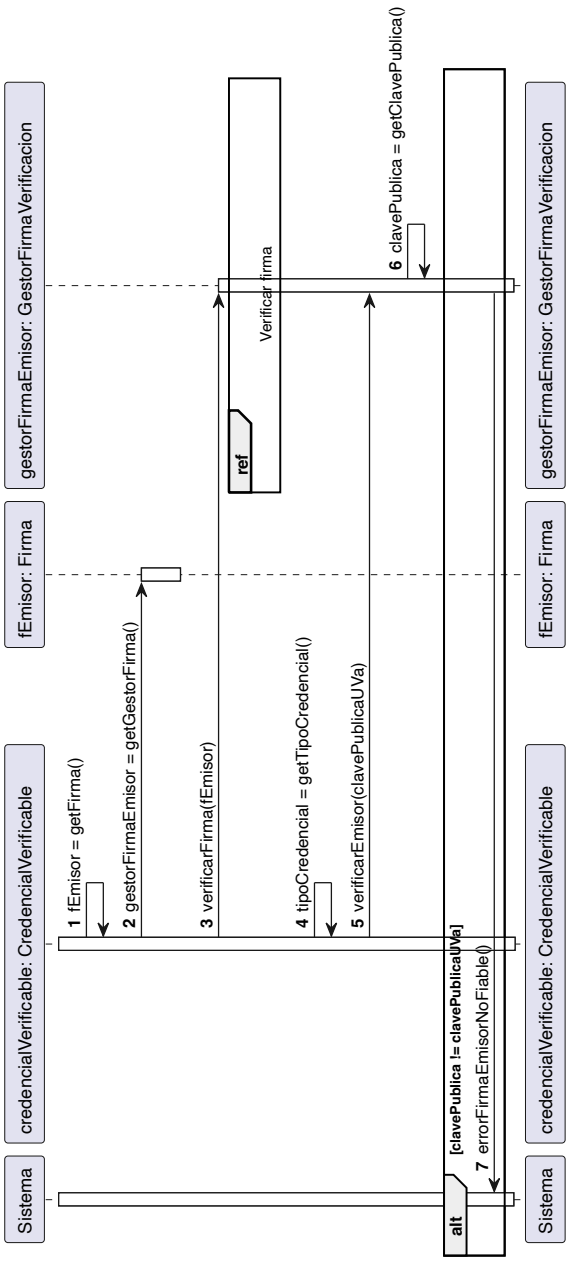


Figura 5.28: Diagrama de secuencia de la verificación de la firma del emisor no cualificado de la Titulación Digital.

Capítulo 6

Diseño

En esta sección se presenta el diseño del sistema propuesto como solución.

6.1. Arquitectura Lógica del Sistema

En la Figura 6.1 se puede ver el diagrama de paquetes UML donde se establece la arquitectura lógica del Sistema. Se ha optado por distinguir dos subsistemas: el subsistema del emisor de titulaciones digitales y el del verificador de titulaciones digitales. En ambos subsistemas se sigue una *Patrón Arquitectónico de Capas* dividida en capa de presentación, en el caso del subsistema del emisor, negocio y persistencia para limitar el acoplamiento al mínimo entre las responsabilidades de cada uno de los ámbitos que representan. A nivel de la interfaz se ha decidido utilizar el patrón *Model View ViewModel* o *MVVM* dado que es el modelo que más encaja al utilizar Vue como framework para la capa de interfaz.

6.1.1. Subsistema Emisor Titulaciones Digitales

En la Figura 6.2 se puede ver el Decomposition y Uses Style general del subsistema de emisión de titulaciones digitales. En la Figura 6.3 se puede ver el Inheritance Style.

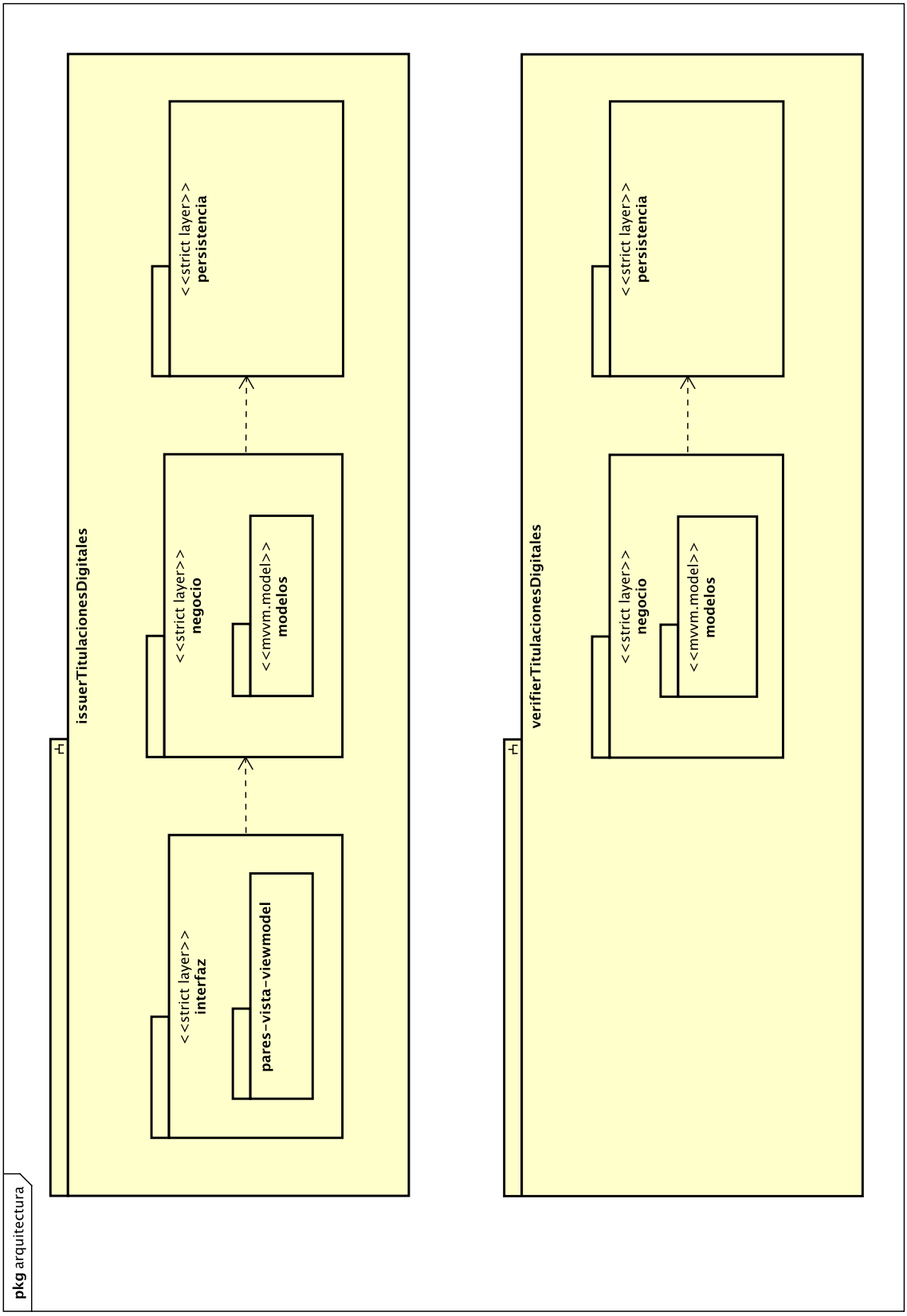


Figura 6.1: Diagrama de paquetes de la Arquitectura Lógica del Sistema.

En la Figura 6.4 se tiene el diagrama de clases del diseño del emisor de titulaciones digitales. En la Figura 6.5 se tiene las clases del diseño del emisor de titulaciones digitales con sus operaciones.

6.1.1.1. Capa de Presentación

La capa de presentación contiene las vistas, que son parte de una aplicación web desarrollada con Vue.js, y sus respectivos *ViewModel*, que se comunican con la capa de negocio para actualizar la vista en consecuencia a los cambios que ocurren en dicha capa. Como permite el patrón MVVM y a su vez el uso de Vue.js se hace uso de la reactividad, observando las propiedades correspondientes de la capa de negocio para poder actualizar la vista desde el ViewModel en cuanto cambian. Las vistas de la capa de presentación se encargan de mostrar al usuario las opciones disponibles, como solicitar la emisión de una titulación digital o revocar una titulación digital ya emitida.

6.1.1.2. Capa de Negocio

La capa de negocio contiene las clases que implementan la lógica del sistema, concretamente la emisión de titulaciones digitales y la revocación de las mismas. Es importante destacar la clase *ServicioAuthPID*, que se encarga de manejar la autenticación del usuario ante el emisor de credenciales mediante la credencial de tipo PID. Para ello hace uso de la librería de terceros que implementa el protocolo OIDC4VP, *OIDC4VPSphereonLibrary*.

Por otro lado también es relevante la clase *IssuerTitulacionesDigitales*, que se encarga de manejar la emisión de las titulaciones digitales. Para ello hace uso de la librería de terceros que implementa el protocolo OIDC4VP, *OIDC4VCISphereonLibrary*.

6.1.1.3. Capa de Persistencia

La capa de persistencia se encarga de almacenar los datos necesarios para el funcionamiento del sistema, como los datos de los usuarios, las titulaciones digitales emitidas y las revocaciones de las mismas. Para ello se utiliza una base de datos relacional en el caso del almacenamiento de los datos de los usuarios, que se utiliza desde la capa de negocio a través de *DAOs* o *Data Transfer Objects*. En el caso de las titulaciones digitales emitidas son almacenadas en una base de datos no relacional, MongoDB, también a través de DAOs. El patrón DAO se utiliza

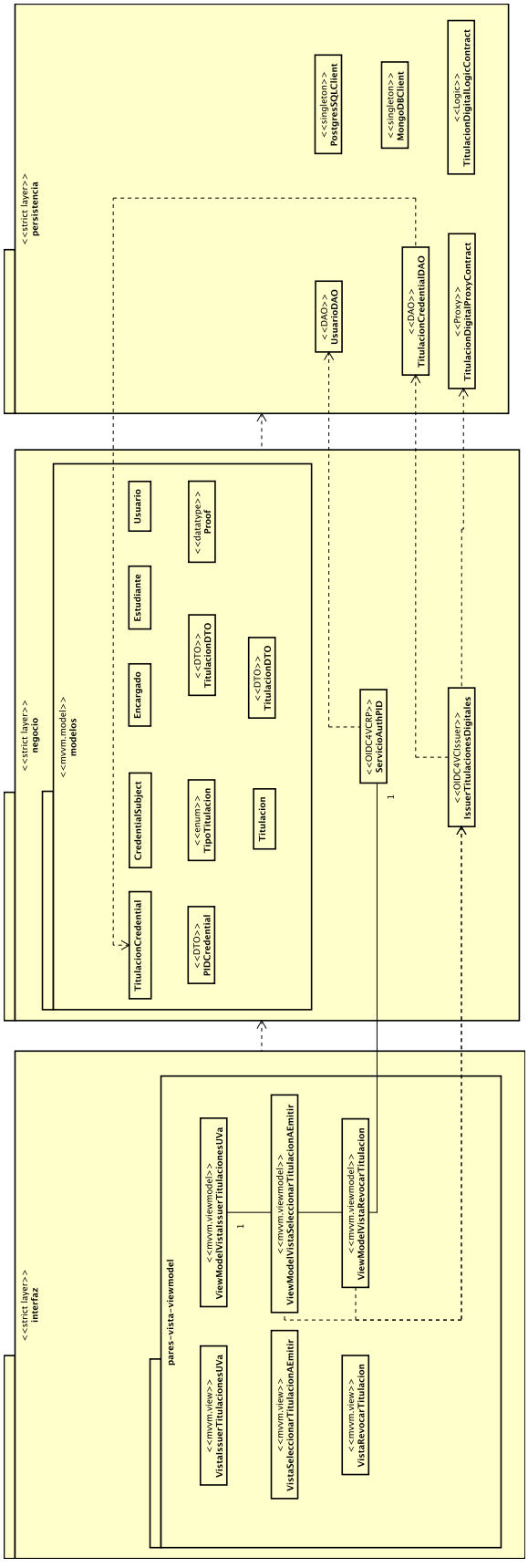


Figura 6.2: Decomposition y Uses Style general del Issuer.

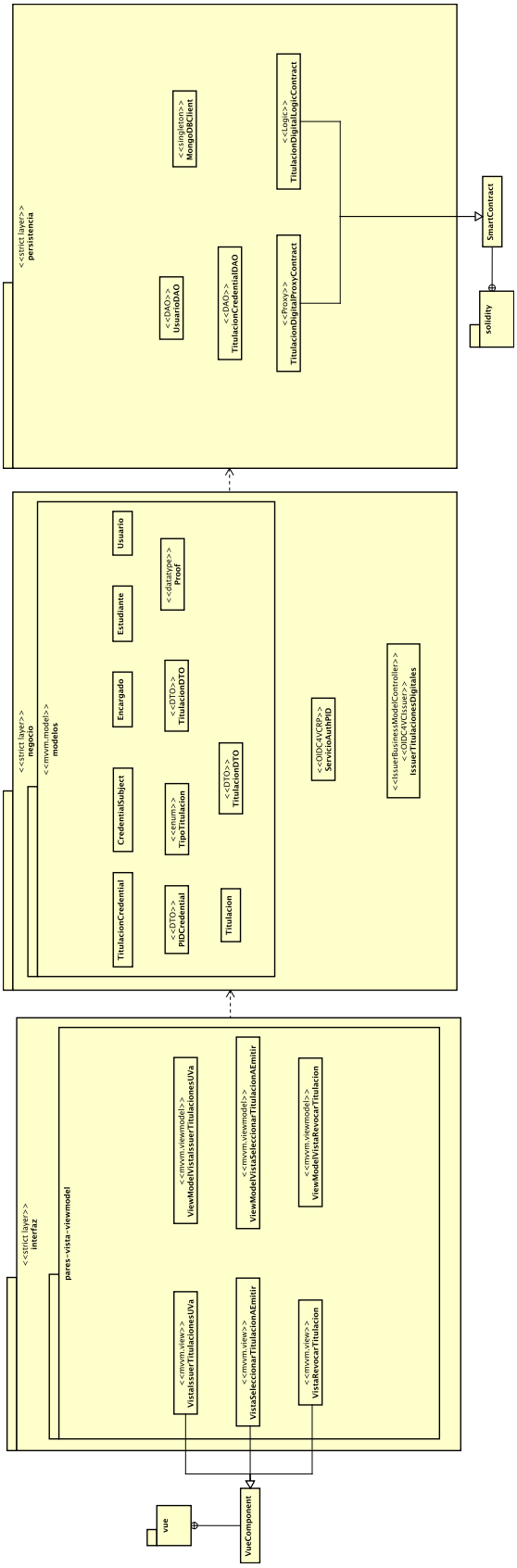


Figura 6.3: Inheritance Style del Issuer.

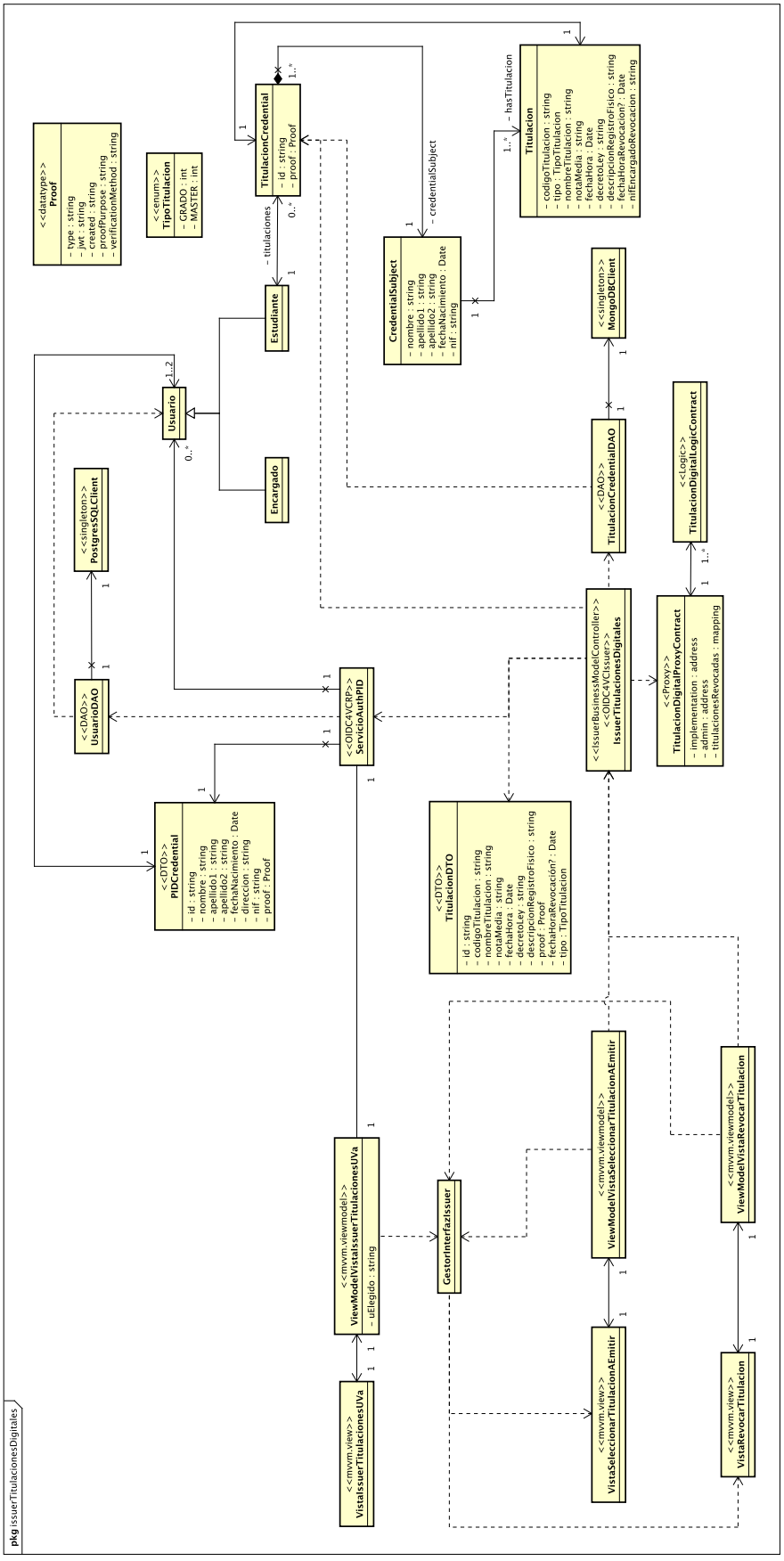


Figura 6.4: Diseño detallado del Issuer.

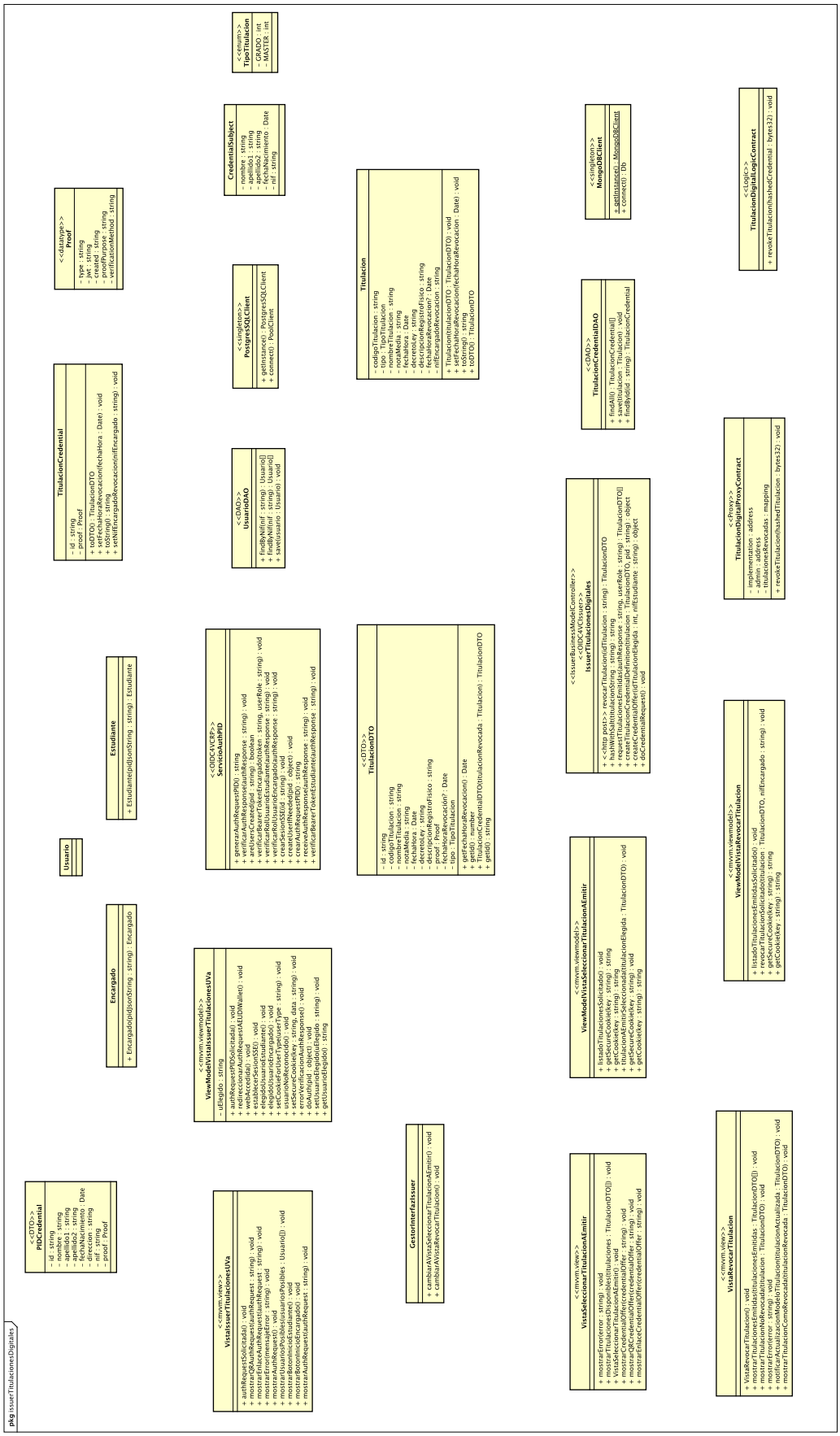


Figura 6.5: Diseño detallado con operaciones del Issuer.

para separar la lógica de acceso a los datos persistidos de la lógica de negocio, permitiendo un acoplamiento más débil entre las capas y facilitando el mantenimiento y la evolución del sistema. Para ello se crea una clase DAO por cada entidad de la lógica de negocio que se quiere persistir, dotada de métodos parea crear, leer, actualizar y eliminar (*CRUD*) los datos de dicha entidad. Por otro lado, se utilizan dos clases para manejar la conexión con la base de datos, una para la base de datos relacional, *PostgresSQLClient*, y otra para la base de datos no relacional, *MongoDBClient*. En ambos casos se utiliza el patrón Singleton para garantizar que solo haya una instancia de la conexión a la base de datos en toda la aplicación, lo que permite un acceso eficiente y controlado a los datos persistidos.

En cuanto a las revocaciones de las titulaciones digitales, estas se almacenan en una blockchain, que es la que permite garantizar la integridad y verificabilidad públicas de las revocaciones. Para ello se utiliza un contrato inteligente que se encarga de gestionar las revocaciones de las titulaciones digitales emitidas por la UVa. Se puede observar que la clase *TitulacionDigitalProxy* está estereotipada como *Proxy*, pero no porque se aplique un Patrón Proxy "tradicional"de la Ingeniería de Software. En este caso se está hablando del Patrón Proxy en el contexto de los Smart Contracts de una blockchain tipo Ethereum. Este consiste en tener un Smart Contract denominado Proxy que delega su funcionalidad en otro denominado Logic, que usa el almacenamiento asociado al Proxy para posibilitar cambiar del Smart Contract de Logic y cambiar, por tanto, la lógica del contrato pero sin perder los datos ya almacenados en el mismo.

6.1.2. Subsistema Verificador Titulaciones Digitales

En la Figura 6.6 se puede ver el Decomposition y Uses Style general del subsistema de verificación de titulaciones digitales. En la Figura 6.7 se puede ver el Inheritance Style. Se puede observar que se tiene la clase ServicioAuthTitulacionCredential, estereotipada como OIDC4VCRP, que ejerce las funciones precisamente de lo que está estereotipada, es decir, de Relying Party de credenciales de tipo Titulacion.

En la Figura 6.8 se tiene el diagrama de clases del diseño del verificador de titulaciones digitales. En la Figura 6.9 se tiene el diagrama de clases del diseño del verificador de titulaciones digitales.

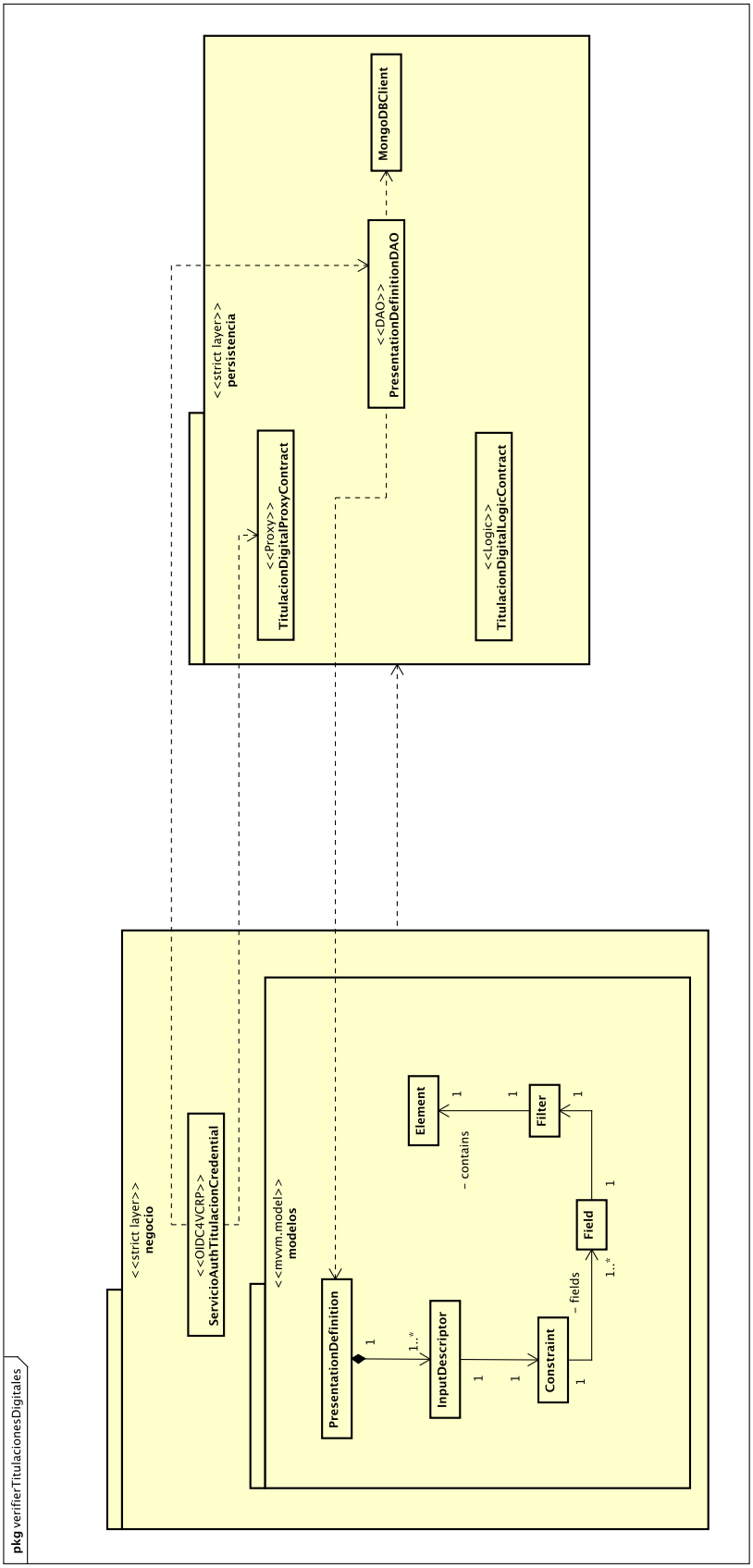


Figura 6.6: Decomposition y Uses Style general del Verifier.

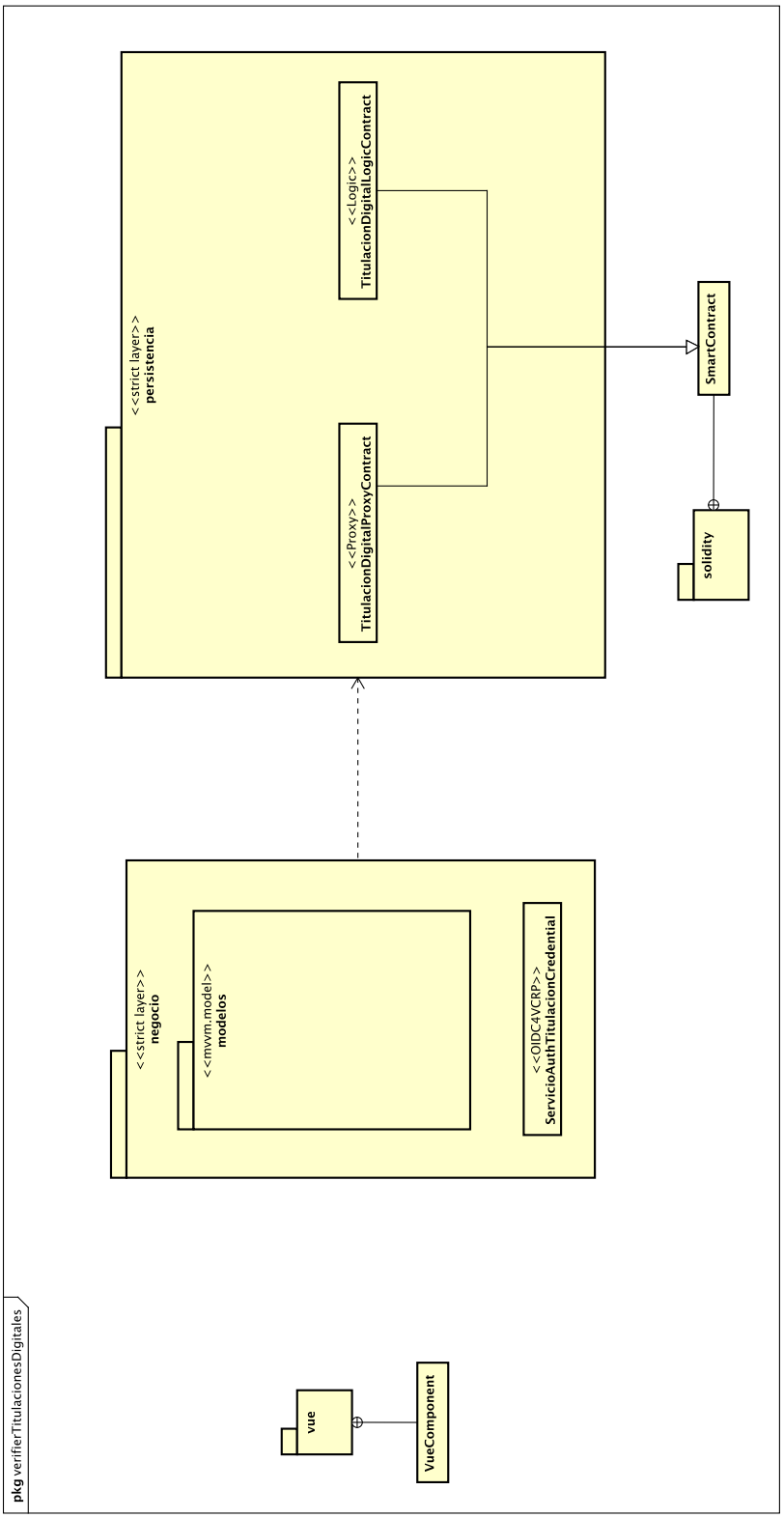


Figura 6.7: Inheritance Style del Verifier.

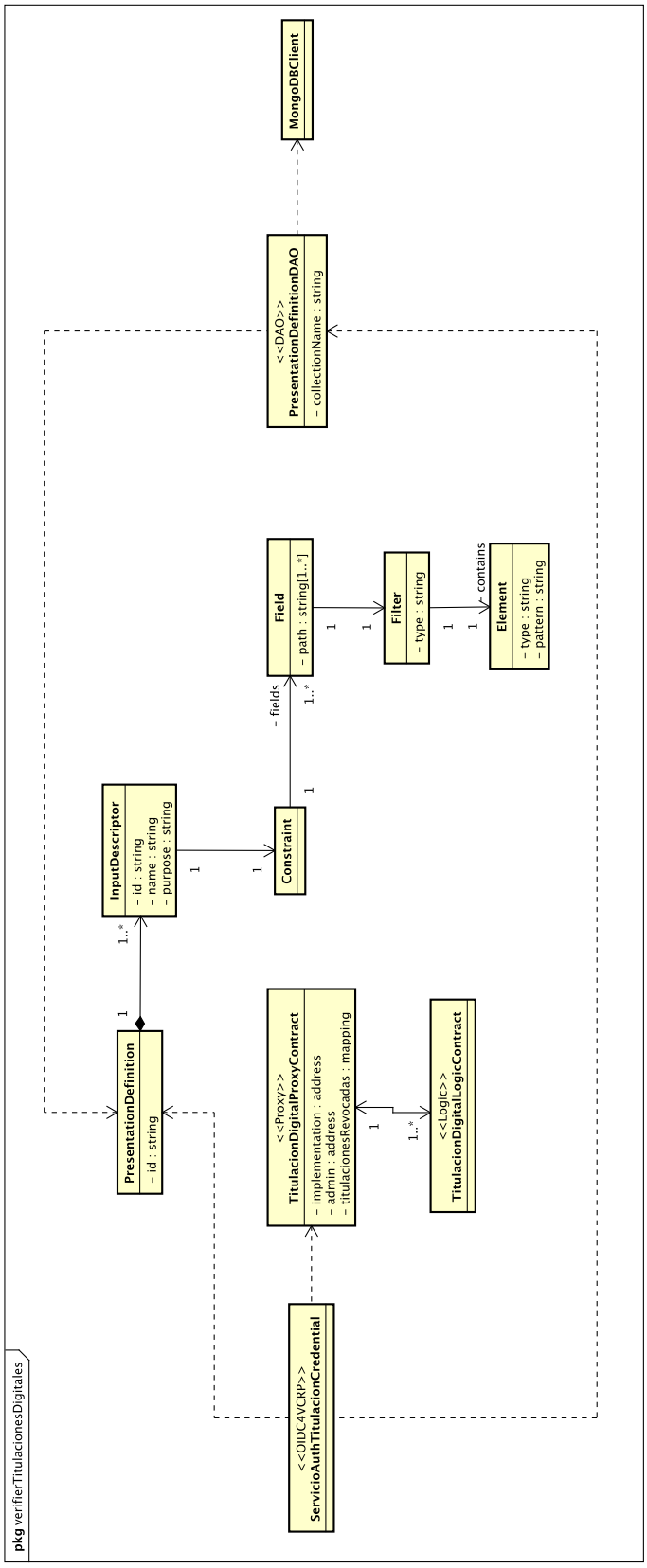


Figura 6.8: Diseño detallado del Verifier.

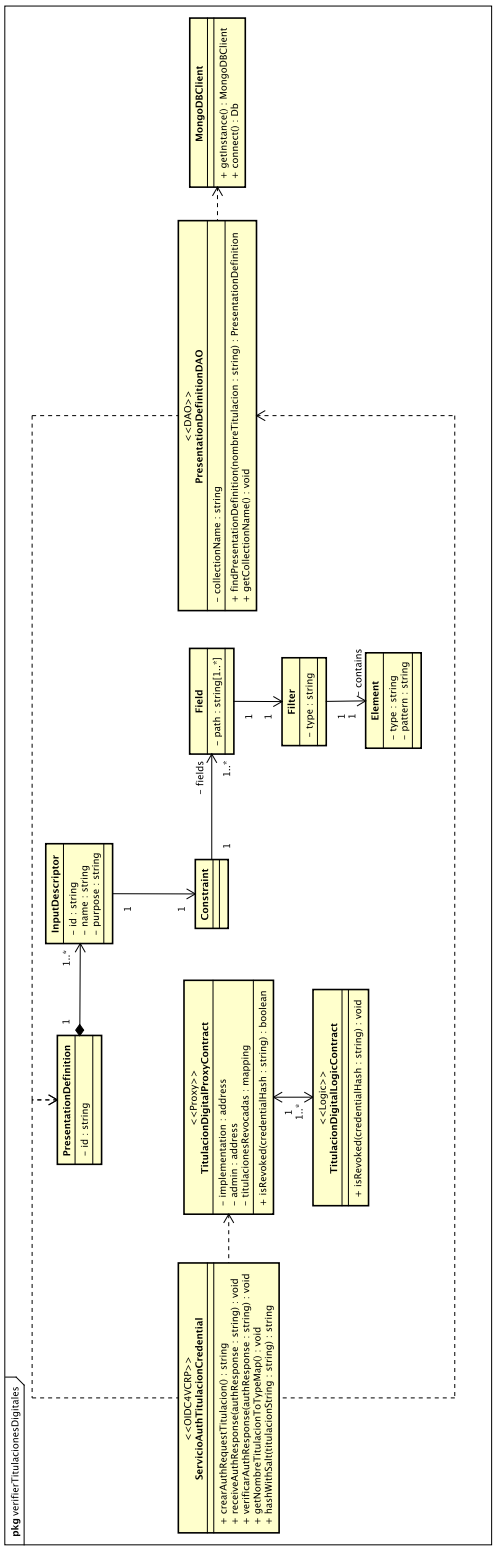


Figura 6.9: Diseño detallado con operaciones del Verifier.

6.1.2.1. Capa de Negocio

La capa de negocio contiene las clases que implementan la lógica del sistema, concretamente la verificación de titulaciones digitales para determinar si el usuario tiene una titulación válida para concederle o denegarle la autenticación de acceso al sistema de RRHH de la empresa correspondiente. Es importante destacar la clase *ServicioAuthTitulacionCredential*, que se encarga de manejar la autenticación del usuario ante el verificador de credenciales mediante una credencial de tipo titulación digital. Para ello hace uso de la librería de terceros que implementa el protocolo OIDC4VP, *OIDC4VPSphereonLibrary*.

6.1.2.2. Capa de Persistencia

La capa de persistencia se encarga de almacenar los datos necesarios para el funcionamiento del sistema, como las definiciones de las presentaciones para cada tipo de titulación digital posible y permitir el acceso a los datos de la blockchain acerca de las credenciales revocadas para revisar que la credencial que se presenta no se encuentre revocada. Es necesario almacenar las definiciones de presentaciones para que desde el sistema desde el que se llama pueda escoger exactamente el tipo de titulación que se requiere para verificar la autenticación del usuario. Como en el caso del emisor de titulaciones digitales, se utiliza una base de datos MongoDB para almacenar los datos de los usuarios, que se utiliza desde la capa de negocio a través de un DAO que conecta con la clase *MongoDBClient*, que aplica un patrón Singleton, y el patrón Proxy para acceder a los datos de la blockchain.

6.2. Realización en Diseño de los Casos de Uso

6.2.1. CU1 Identificarse ante el Sistema de Emisión de Titulaciones Digitales

En la Figura 6.10 se puede ver el diagrama de secuencia del CU Identificarse ante el Sistema de Emisión de Titulaciones Digitales. Como se puede observar en el diagrama, al entrar a la web del emisor de titulaciones digitales se establece una sesión *Server-Sent Events (SSE)* para poder notificar a la vista una vez se compruebe que la wallet dispone y presenta la credencial de tipo PID para identificar al usuario en el emisor de titulaciones digitales de la UVa. Este modelo de eventos establece un canal de los mismos entre el navegador y el backend, que sería la capa

de negocio, para que ante un evento emitido en el backend la vista sea capaz de poder responder en consecuencia para mostrar algo al usuario.

El uso de *Server-Sent Events* es necesario para poder reconocer cuando mostrar la selección de usuario, directamente la lista de titulaciones con posibilidad de ser emitidas o la lista de credenciales emitidas en caso de ser Encargado. También es necesario para permitir que se disponga de la *Authentication Response* desde el ViewModel de la vista una vez generado en la wallet del usuario para poder presentarlo contra el servicio de autenticación del emisor, *ServicioAuthPID*.

El proceso de obtención de la Authentication Response se puede observar en la Figura 6.11, tras haberse solicitado en un primer lugar la generación de una *Authentication Request* al entrar el usuario en la web del emisor de titulaciones. La Authentication Request se basa principalmente en la utilización de una Presentation Definition que acota la credencial o credenciales que se le solicitan a la wallet del usuario. En la Figura 6.13 se puede ver la Presentation Definition para el caso concreto planteado. Básicamente sirve para filtrar el tipo de credencial admitida para ser presentada de entre todas las presentes en la wallet del usuario. Esta Authentication Request se muestra al usuario a través de un QR y a través de un enlace, pudiendo el usuario elegir escanear el QR con su EUDI Wallet o bien pulsar sobre el enlace para ser redirigido a la misma en caso de encontrarse la wallet y el navegador con el que accede a la web en el mismo dispositivo. En el caso de pulsar sobre el enlace este invocará a la EUDI Wallet a través de lo que se denomina como *deeplink*, que es como una url tradicional pero diseñado para ser interceptado por una aplicación que está escuchando y esperando un prefijo concreto en lugar del tradicional "http://", como podría ser ”eudiwallet://”.

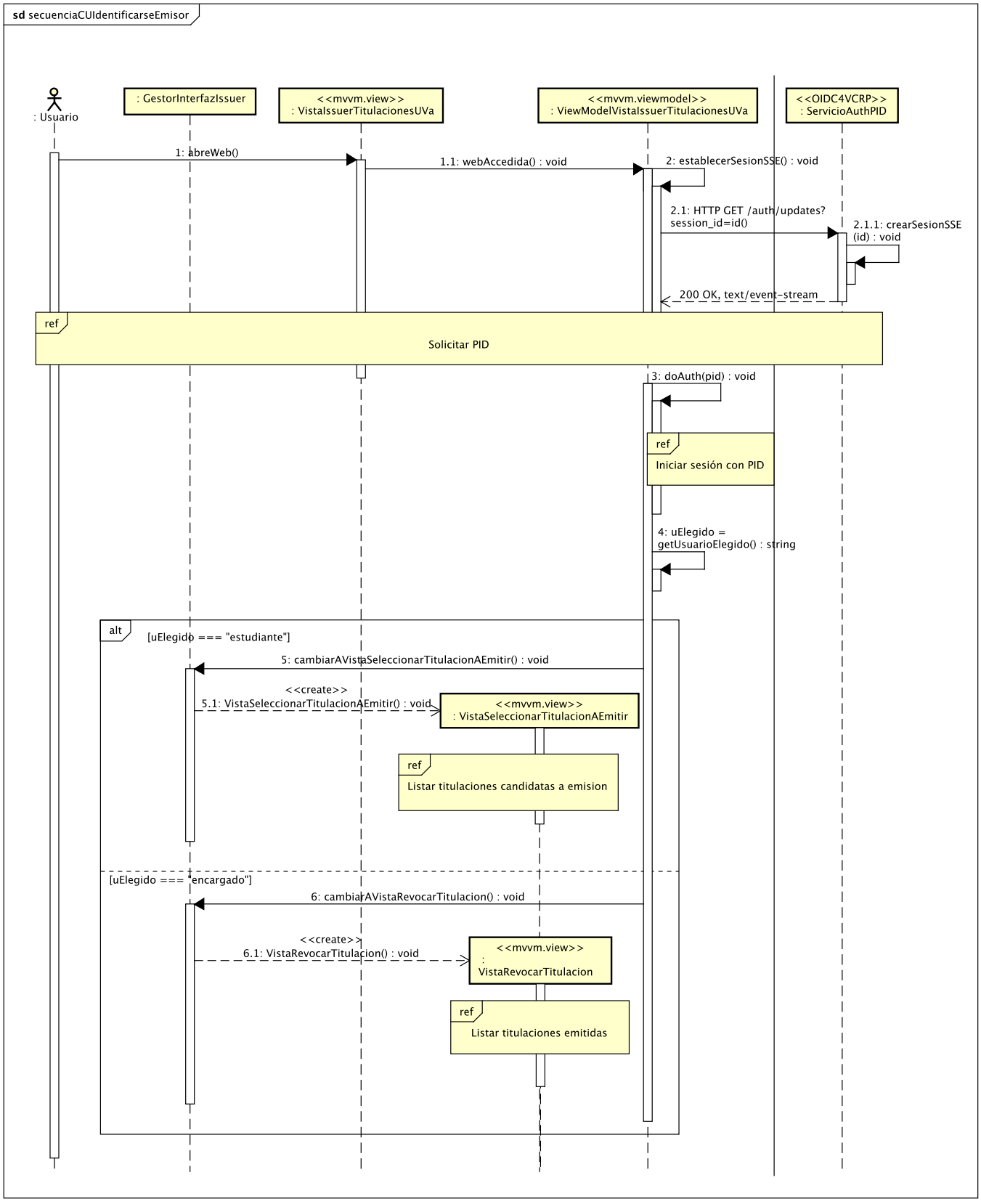


Figura 6.10: Realización en Diseño CU Identificarse ante el Sistema de Emisión de Titulaciones Digitales.

A partir de entonces el usuario deberá seleccionar la credencial a presentar que coincida con lo solicitado en la Authentication Request, caso que se sale de este sistema y queda en manos del sistema de la EUDI Wallet. Si se escanea el QR lo único que cambia es el medio por el que llega la Authentication Request a la EUDI Wallet, que es el componente que la interpreta y discierne para ver qué credenciales debe decidir presentar o no el usuario.

Tras la confirmación de presentar determinadas credenciales por el usuario desde su EUDI Wallet se invoca mediante un HTTP POST al endpoint encargado de procesar la Authentication Response para comprobar que sea válida, es decir, que la Presentación Verificable esté firmada por el dueño de la EUDIWallet, que también será el sujeto que figure en la credencial de tipo PID que contiene. Tanto la comprobación de la firma de la Presentación, como la de la entidad que ha emitido la Credencial se realiza a partir del DID en la red Ethereum que se asume asociado a ellas. No se trabaja con firmas cualificadas finalmente debido a que la librería escogida, que es de las más avanzadas, no lo soporta por el momento. La llamada de verificación hacia la librería de terceros se puede ver en la Figura 6.14. Una vez verificada se establece mediante una cookie para que pueda accederse a ella para pasarla como *Bearer Token* en siguientes peticiones que requieran acceso a recursos protegidos del issuer, como la propia emisión de titulaciones digitales o la revocación de titulaciones digitales ya emitidas, como veremos en los diagramas de secuencia correspondientes.

Tras la obtención de la Authentication Response es necesario comprobar si el usuario que refleja el PID contenido en ella debe ser creado o recuperarlo si ya existe en la base de datos. Como se puede observar en Figura 6.15 de la Auth Response se extrae la credencial PID para poder solicitar al ServicioAuthPID la creación del usuario correspondiente a dicho PID. Si el usuario ya existe en la base de datos entonces no se crea de nuevo y en cualquier caso se revisa contra el Sistema de la UVa si el usuario con determinado NIF se corresponde con un estudiante, un encargado o ambos. La conexión con la BD PostgreSQL se puede ver en la Figura 6.17.

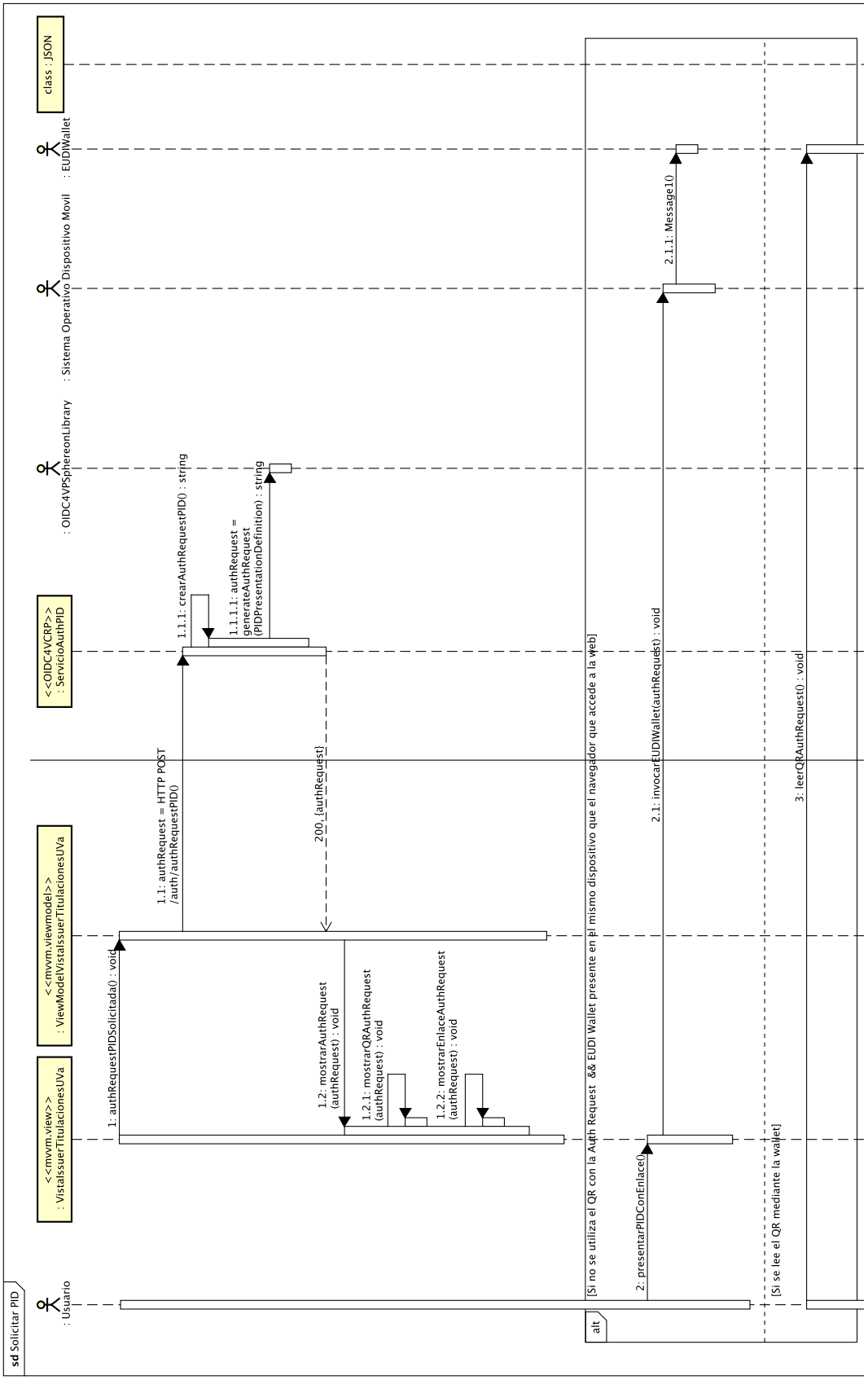


Figura 6.11: Diagrama de secuencia de la solicitud del PID. Parte 1.

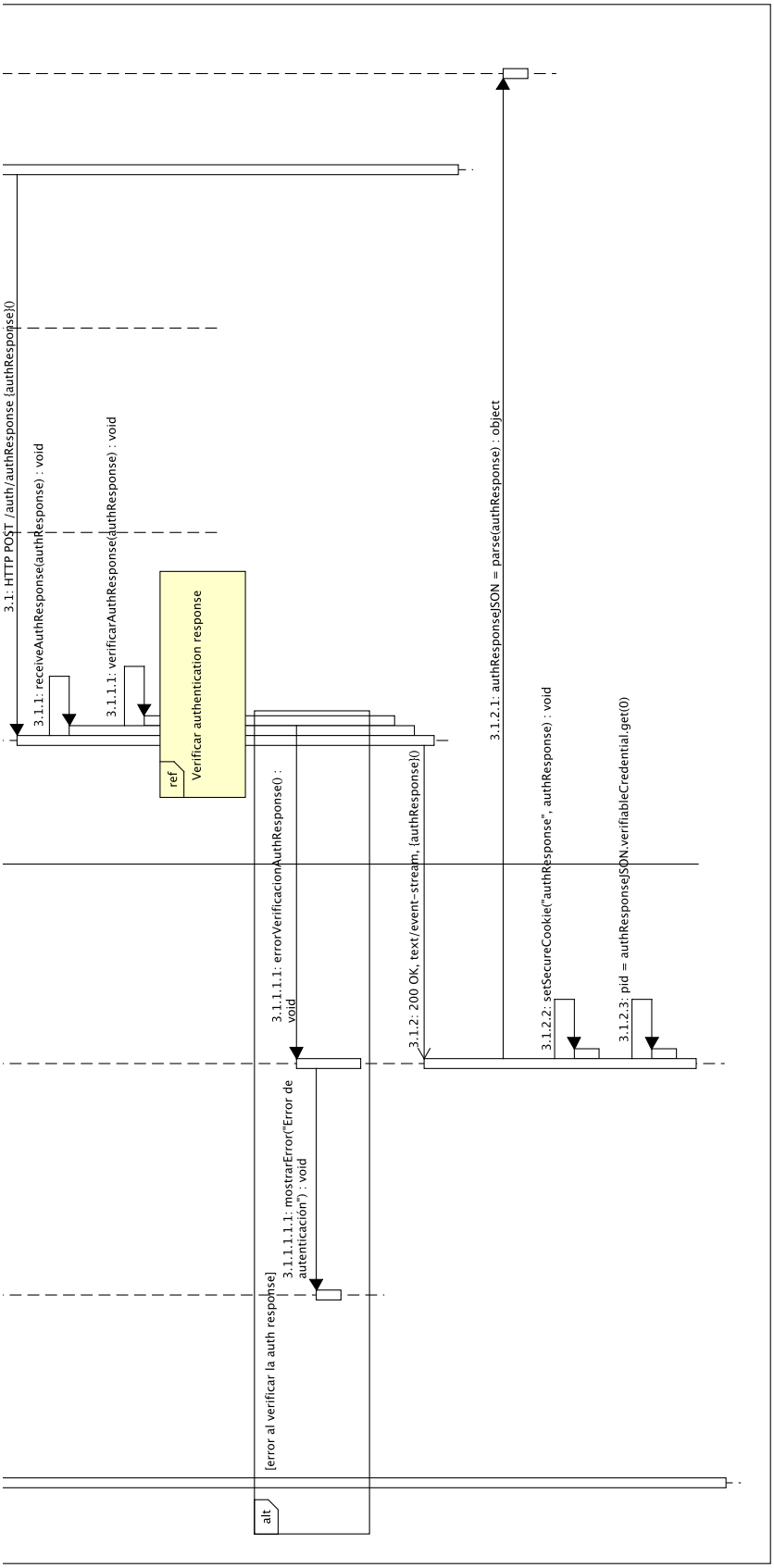


Figura 6.12: Diagrama de secuencia de la solicitud del PID. Parte 2.

```
{
  "presentation_definition": {
    "id": "PID Presentation Definition",
    "input_descriptors": [
      {
        "id": "PID",
        "name": "Personal Identification VC",
        "purpose": "Request presentation of PID",
        "constraints": {
          "fields": [
            {
              "path": [
                "$.type"
              ],
              "filter": {
                "type": "array",
                "contains": {
                  "type": "string",
                  "pattern": "^PIDCredential"
                }
              }
            }
          ]
        }
      }
    ]
  }
}
```

Figura 6.13: Presentation Definition para la credencial de tipo PID.

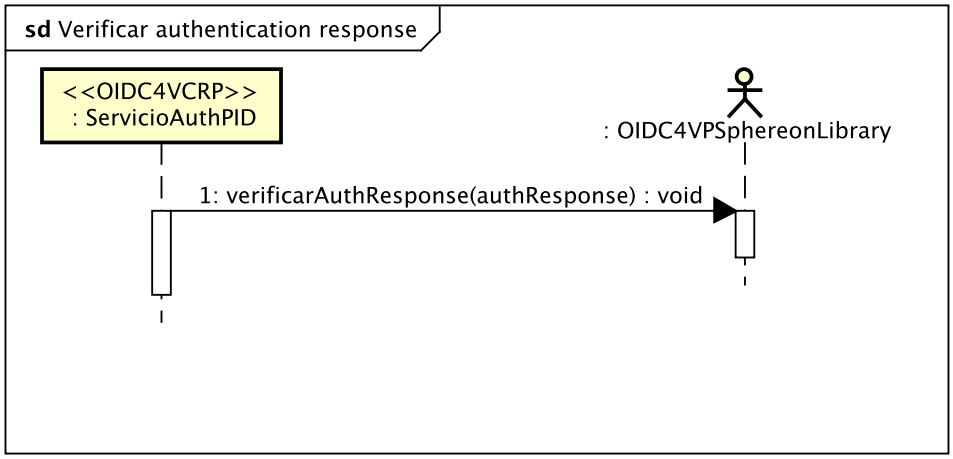


Figura 6.14: Diagrama de secuencia de la verificación de la Authentication Response por parte de la librería de Sphereon.

En el caso de tener ambos roles se le ofrece al usuario que elija con cual se quiere identificar (Figura 6.18). A la hora de verificar llamadas contra el emisor se verificará tanto la Auth Response como tal como si el usuario tiene realmente el rol que debe contra el sistema de la UVa.

Por último, según si se da el caso de estudiante o encargado se le muestra al usuario la VistaSeleccionarTitulacionAEmitir, con la lista de las titulaciones que el Sistema de la UVa indica que son aptas para su emisión en formato digital, que serán las asignaturas ya finalizadas y emitidas en formato digital (Figura 6.19), o bien la VistaRevocarTitulacion, con las titulaciones ya emitidas en formato digital, tanto las que ya han sido revocadas como las que pueden serlo si el Encargado así lo considera (Figura 6.20). En ambos casos se solicitan dichas acciones por el ViewModel correspondiente contra el *OID4VCIssuer*, pasando en la petición tanto la Auth Response, para que sea verificada por este contra el ServicioAuthPID, como la cookie que indica el rol con el que se identificó el usuario. La verificación de autenticación como estudiante y como encargado se pueden ver en la Figura 6.21 y en la Figura 6.22, respectivamente.

6.2.2. CU2 Solicitar Emisión Titulación Digital

En la Figura 6.23, 6.25 y 6.26 se puede ver el diagrama de secuencia en diseño de la solicitud de emisión de titulaciones digitales. El estudiante solicita a la *VistaSeleccionarTitulacionAEmitir* la emisión de la titulación que elija tras haberse identificado. El *ViewModelVistaSleccionarTitulacionAEmitir* se encarga de contactar con el *IssuerTitulacionesDigitales*, utilizando la *authResponse* presente en forma de cookie para autenticar la llamada y asegurar que se trata de un estudiante identificado, para obtener una *Credential Offer* de la titulacion escogida para emitir. Durante esta llamada se realizan verificaciones análogas a los casos anteriores, comprobando a mayores que el estudiante tiene la titulacion cuya emisión se solicita entre aquellas que están disponibles para ser emitidas en formato digital. Tras esto se inicia el proceso de emisión como tal solicitando a la librería externa que provee el protocolo de emisión, *OIDC4VCISphereon*, la creación de la Credential Offer. Para solicitar la creación de la Credential Offer se requiere una Credential Definition, en este caso de una credencial de tipo Titulación.

En la Figura 6.27 se puede ver la *Credential Definition* necesaria para solicitar la emisión de una Titulacion. Esta es un string JSON que contiene los datos de la Titulación y del Estudiante, que se devuelve al *ViewModel* y se manda mostrar a la Vista en forma de QR y como deeplink mediante un botón, para emitir la credencial hacia la

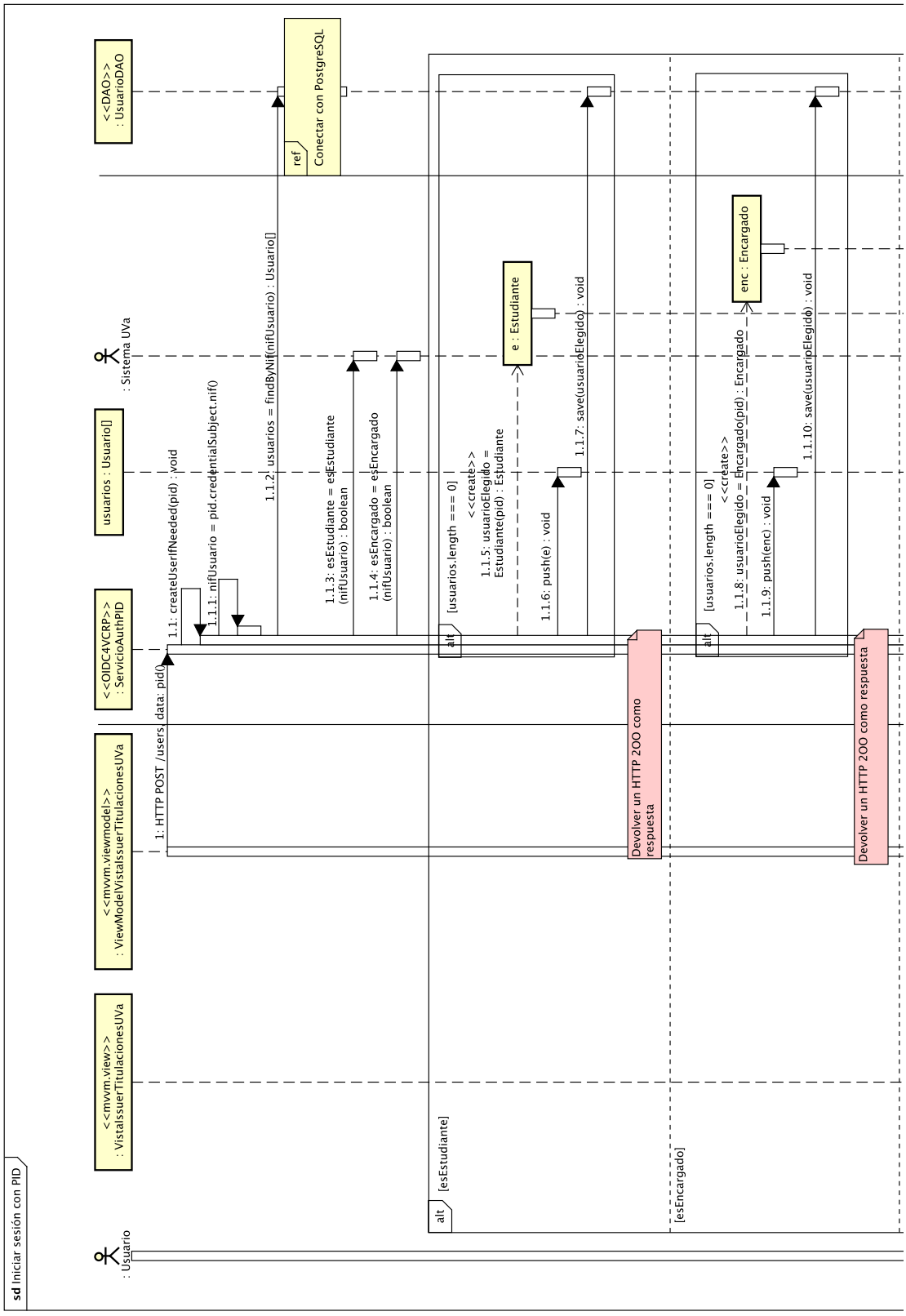


Figura 6.15: Diagrama de secuencia del inicio de sesión mediante un PID. Parte 1.

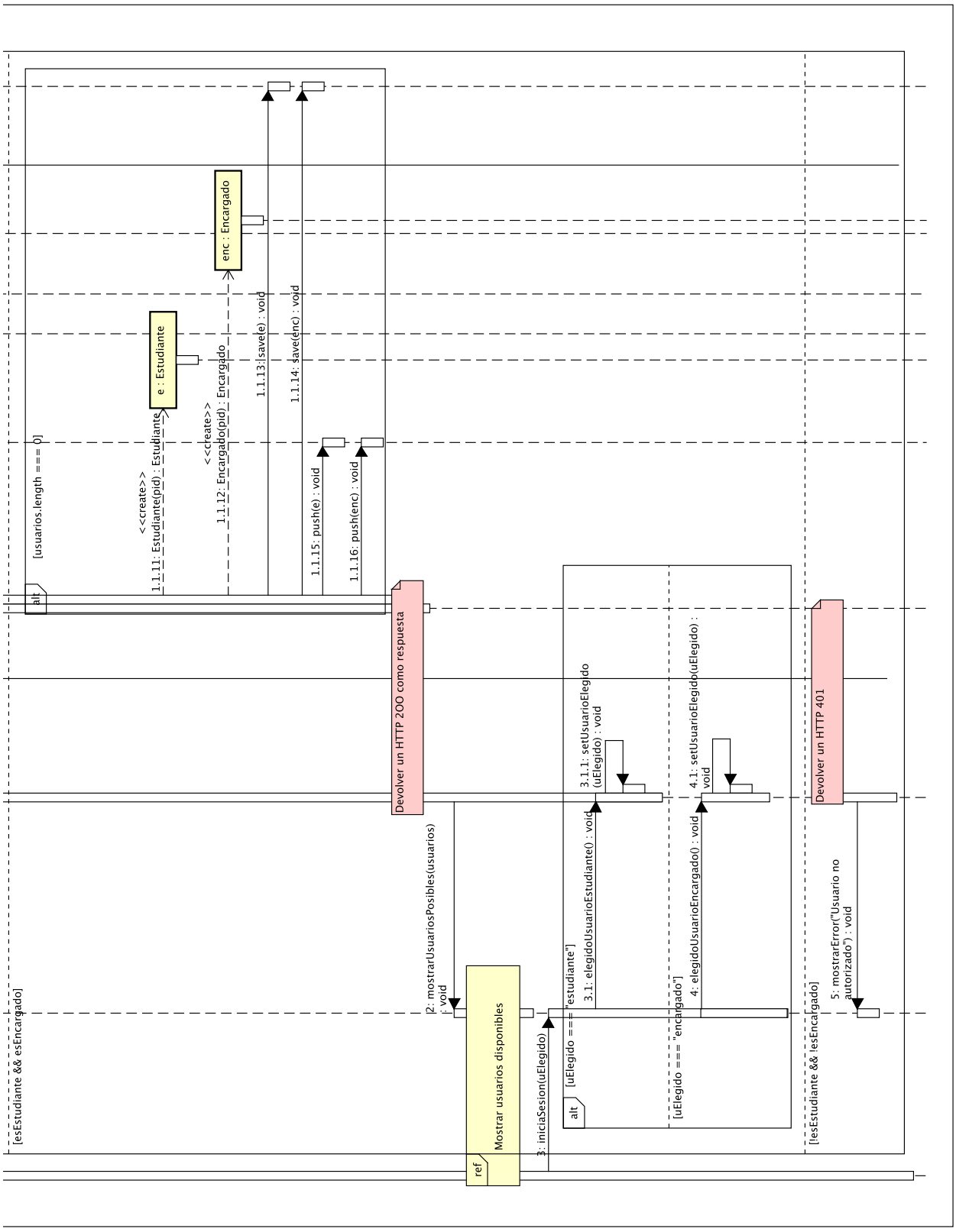


Figura 6.16: Diagrama de secuencia del inicio de sesión mediante un PID. Parte 2.

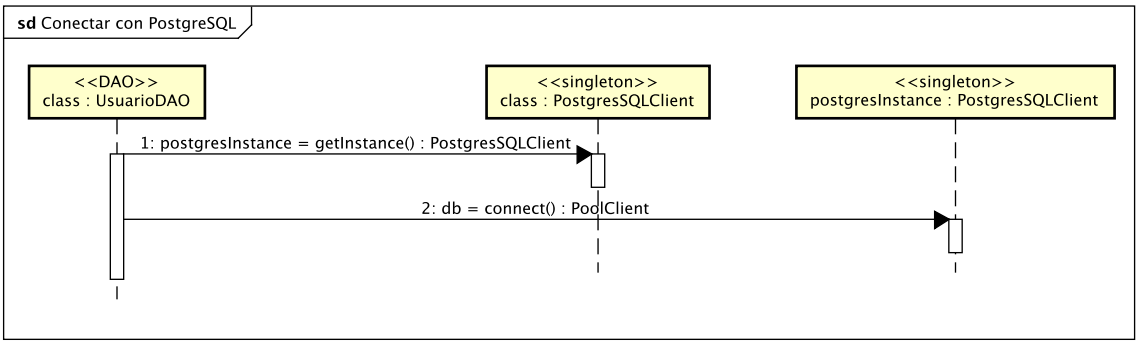


Figura 6.17: Diagrama de secuencia de la conexión con PostgreSQL.

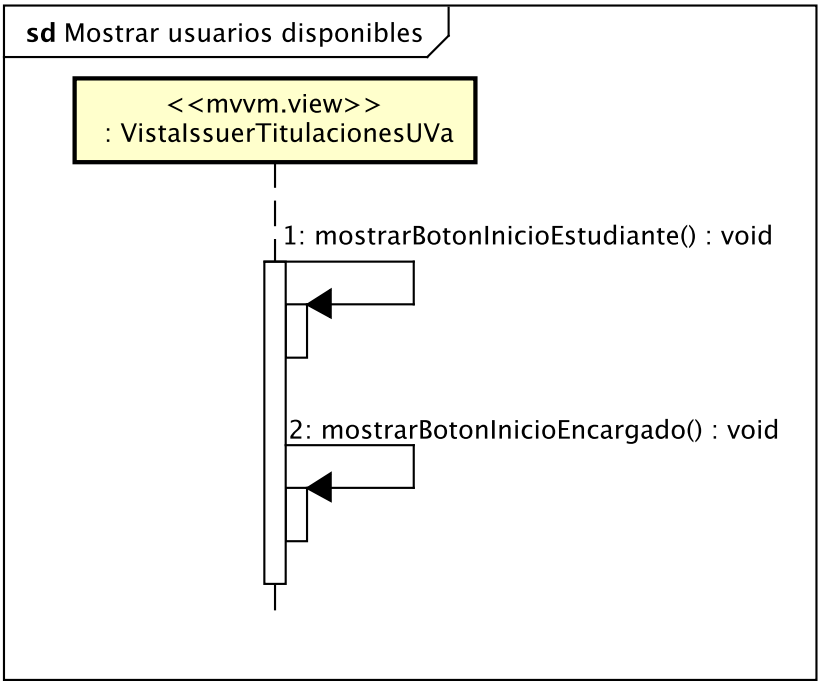


Figura 6.18: Diagrama de secuencia que muestra los perfiles de usuario disponibles al usuario.

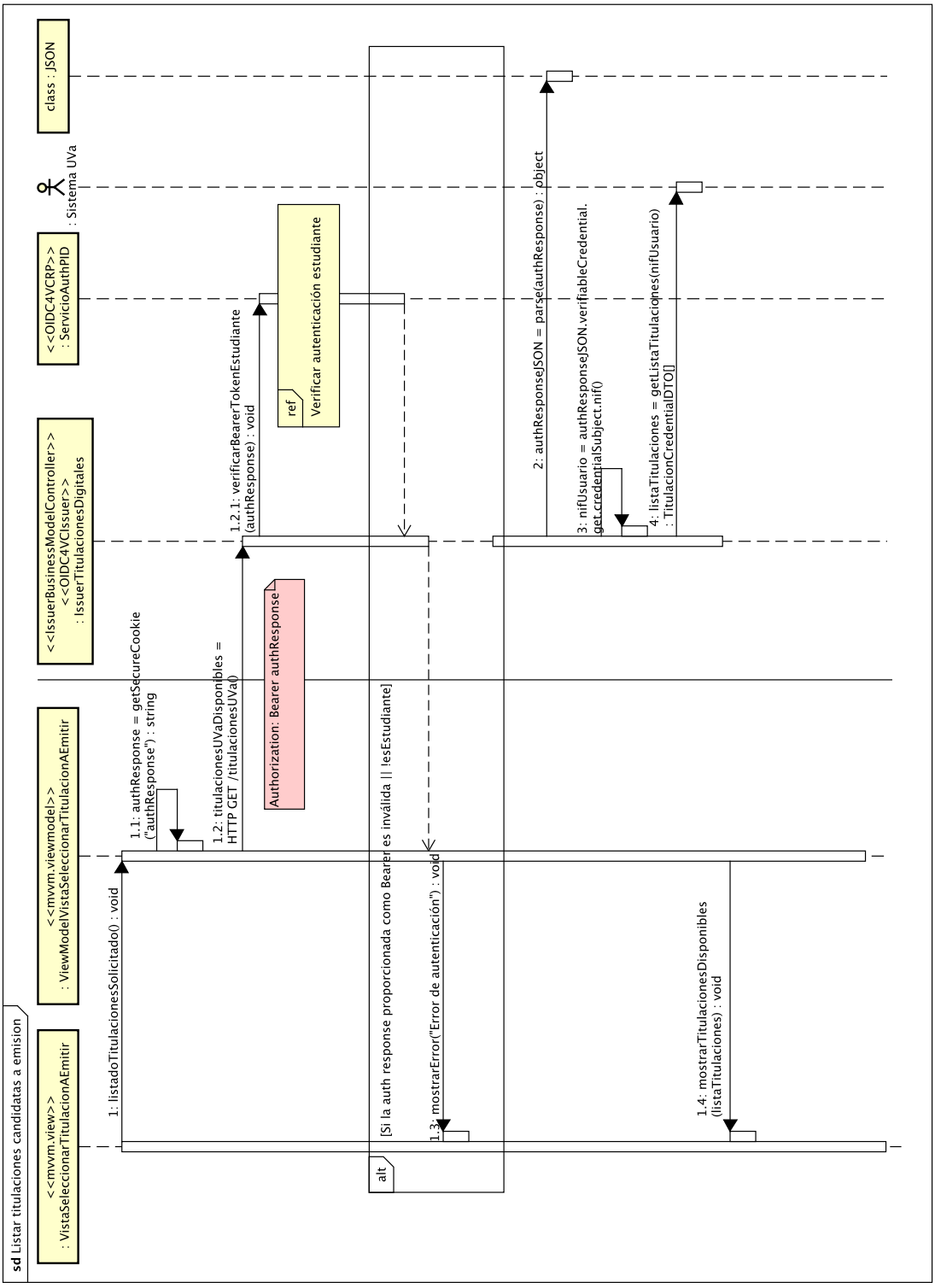


Figura 6.19: Diagrama de secuencia que muestra las titulaciones disponibles para ser emitidas.

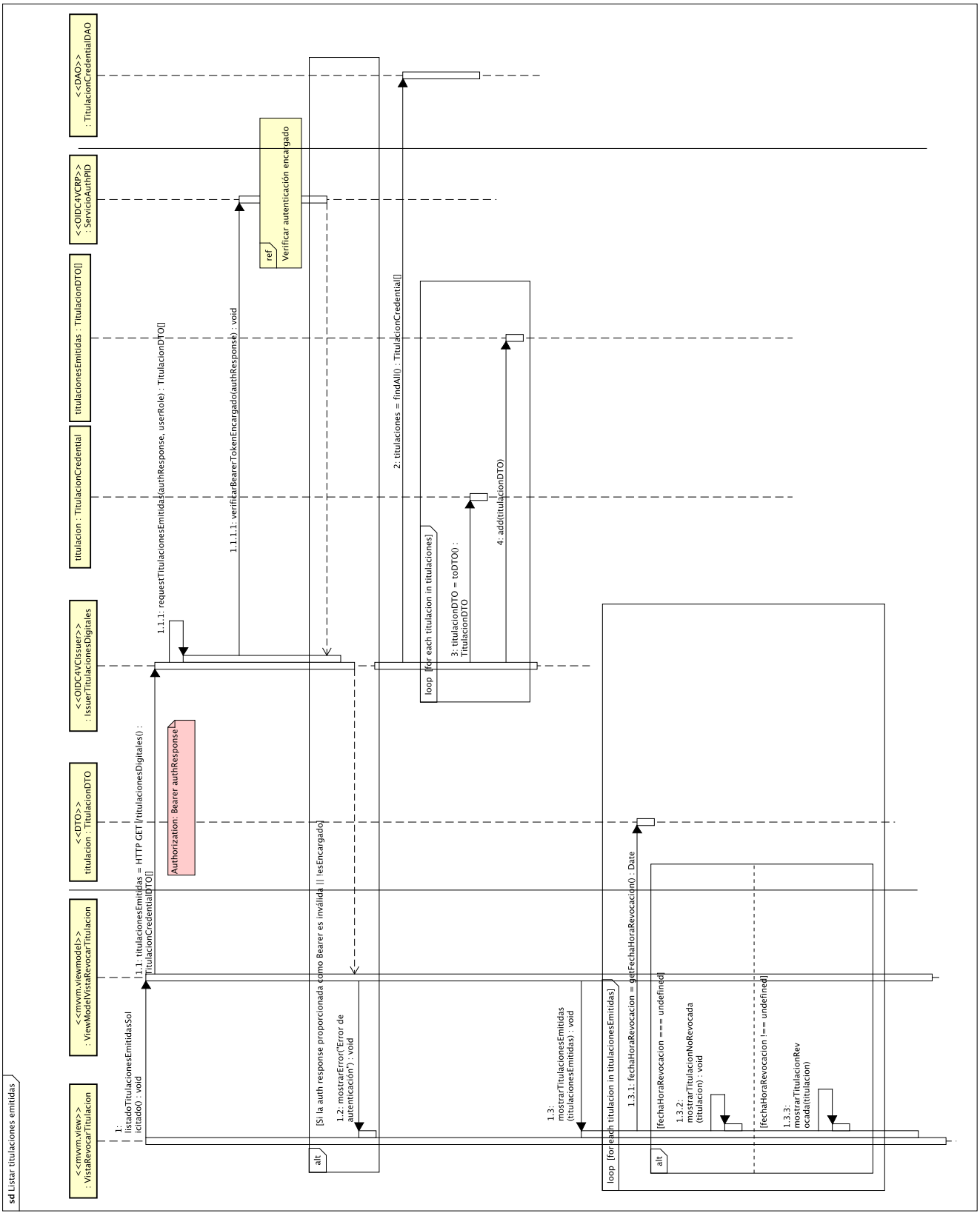


Figura 6.20: Diagrama de secuencia que muestra las titulaciones emitidas.

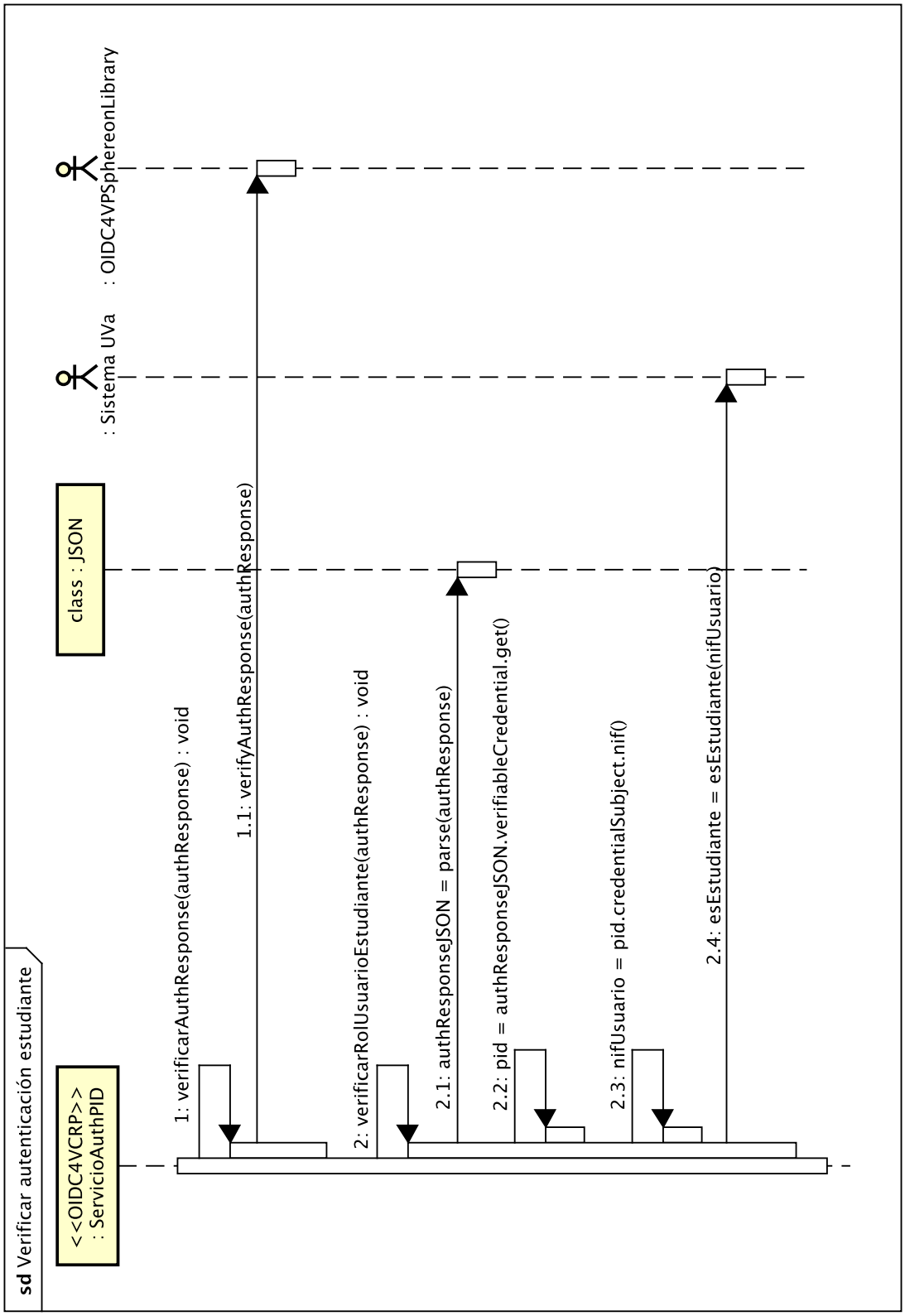


Figura 6.21: Diagrama de secuencia que verifica la autenticación como estudiante.

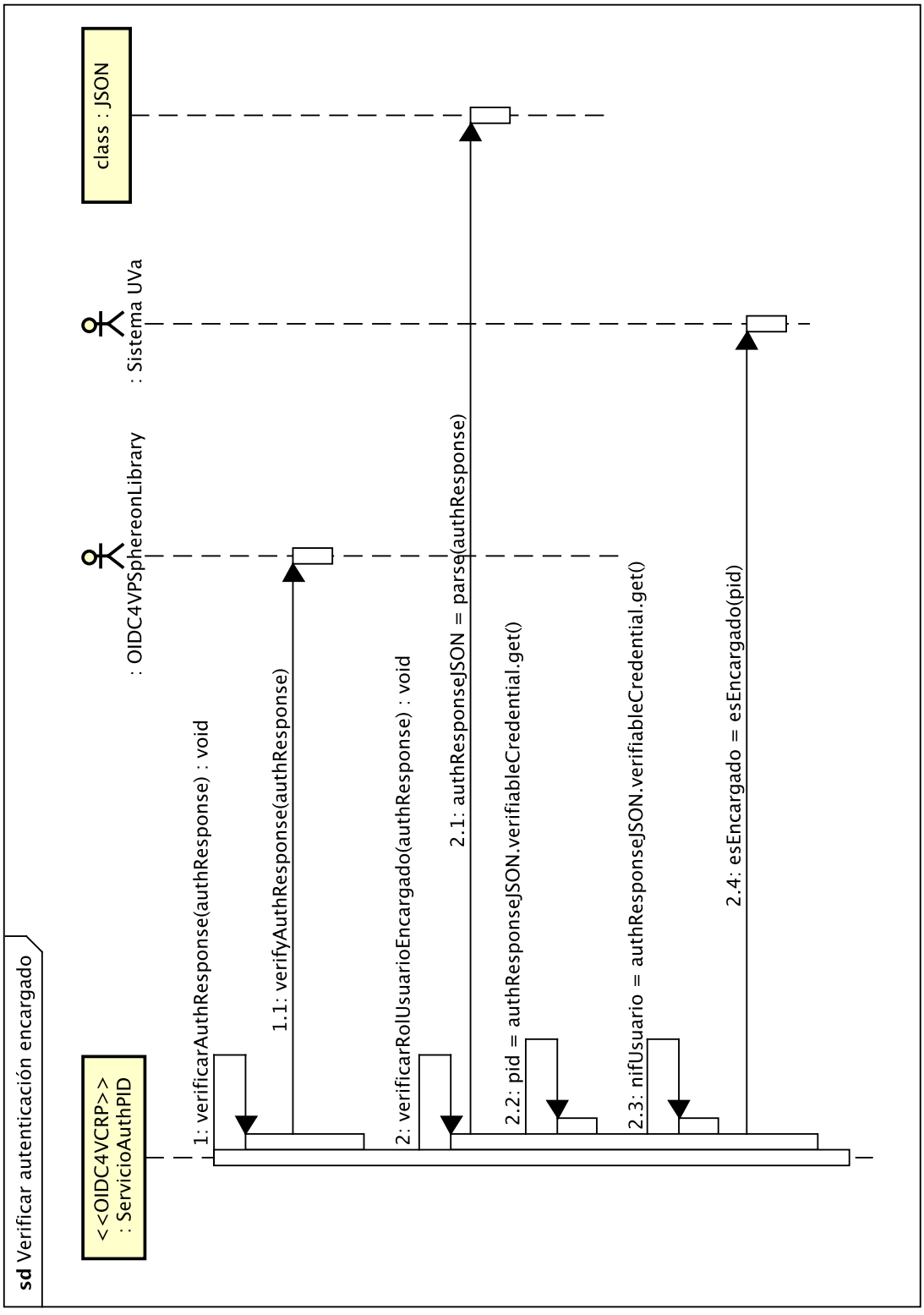


Figura 6.22: Diagrama de secuencia que verifica la autenticación como encargado.

wallet en un dispositivo distinto al que se está utilizando para solicitar la emisión o para hacerlo desde el mismo dispositivo, respectivamente. Junto con la Credential Offer, en esta implementación, se envía el pin a utilizar por el usuario desde la wallet, junto con el pre authorization code, para seguir con el proceso de emisión.

Ahora que ya se ha recibido la Credential Offer en la wallet y el pin proporcionado por el Issuer, y tras la confirmación por parte del usuario para proceder a emitir la credencial que se le ofrece, se realiza la petición de un token de acceso o *Access Token* utilizando el pin y el pre authorization code contra el endpoint correspondiente de la librería de terceros utilizada. Con este token ya se puede solicitar la emisión de la credencial como tal. Como se puede ver en el diagrama también se almacena la credencial emitida en la base de datos, para tener una trazabilidad de las titulaciones emitidas. La *EUDI Wallet* recibe y almacena la credencial emitida, lista para su uso.

6.2.3. CU3 Revocar Titulacion Digital

En la Figura 6.28 se puede ver el diagrama de secuencia en diseño de la revocacion de titulaciones digitales. En primer lugar se realiza una verificación análoga a la explicada anteriormente para comprobar que el usuario es un Encargado identificado.

Acerca de este proceso es relevante mencionar que se hace un hash con lo que se denomina un *salt* a mayores, que es simplemente un string secreto que le da una mayor aleatoriedad, de la representación en JSON de la titulacion digital a revocar para poder almacenar en el smart contract con el que se contacta que ese hash corresponde a una titulacion que ha sido revocada. A mayores se almacena en la base de datos MongoDB el estado revocado asociado a la credencial. La conexión con la BD se puede ver en la figura 6.30 Una vez se realiza el cambio al estado de revocada en el modelo, que es almacenado en el *OIDC4VCIssuer* y expuesto al viewmodel a través de un endpoint en el backend, este notifica a la *VistaRevocarTitulacion* para que se actualice reflejando el nuevo estado de revocación para la titulacion mostrada. Por este comportamiento, que es el típico al utilizar el framework *Vue* para la interfaz de usuario se ha escogido el patrón *ViewModel* y no un patrón *MVC* u otro. Por último destacar el uso de un *Data Transfer Object* o *DTO* a nivel de interfaz para trabajar sin tener que duplicar en esta el modelo de la titulacion que se encuentra en el backend anteriormente mencionado y evitar un mayor acoplamiento innecesario desde la capa de interfaz a la del modelo.

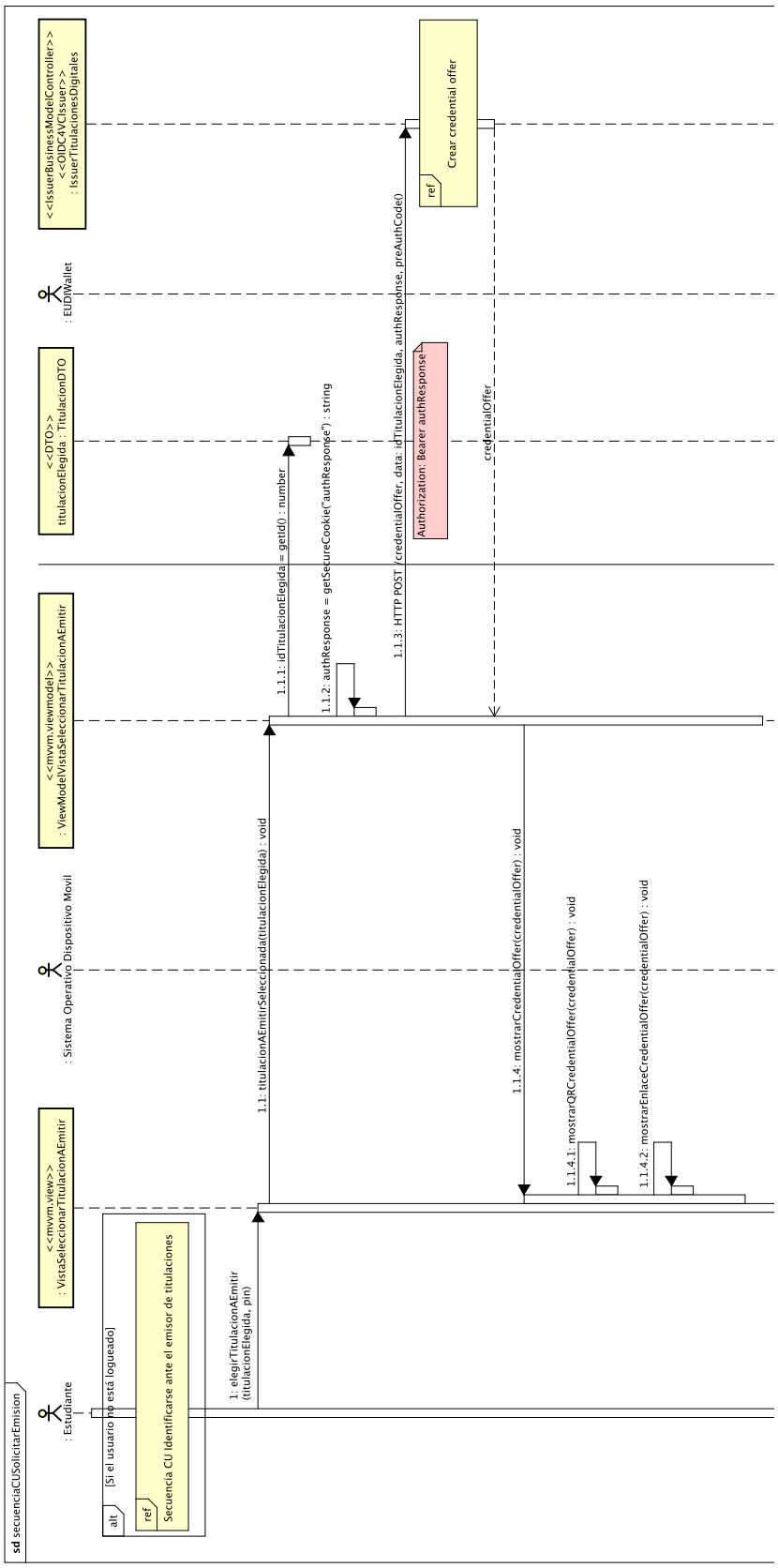


Figura 6.23: Diagrama de secuencia de la emisión de una titulacion digital. Parte 1.

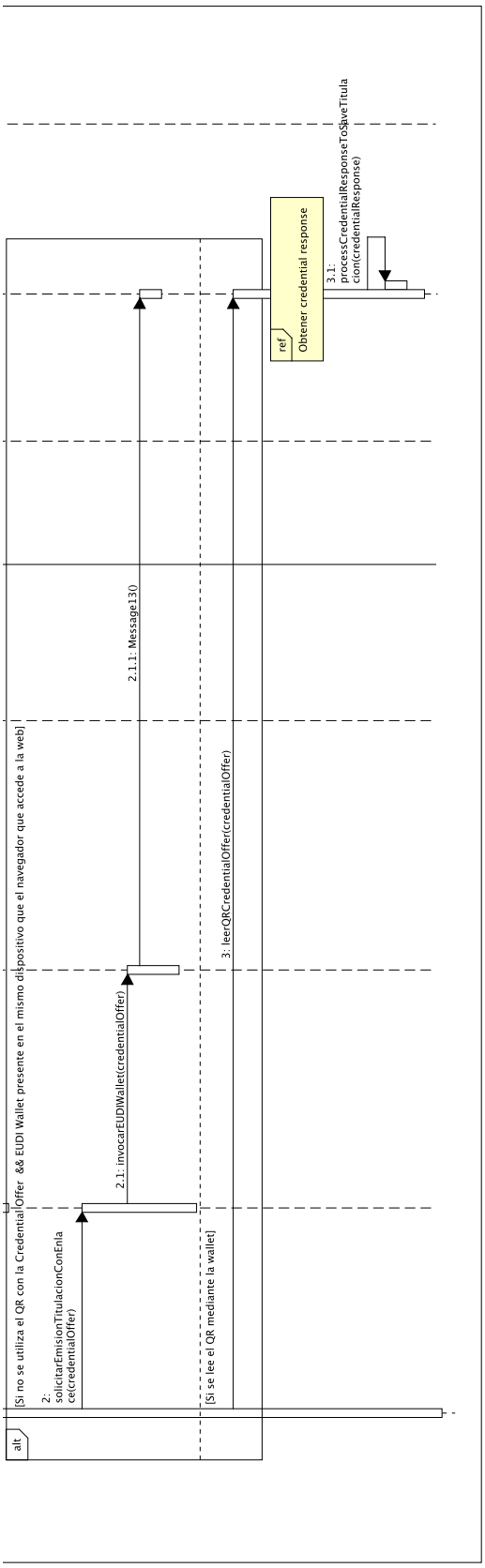


Figura 6.24: Diagrama de secuencia de la emisión de una titulacion digital. Parte 2.

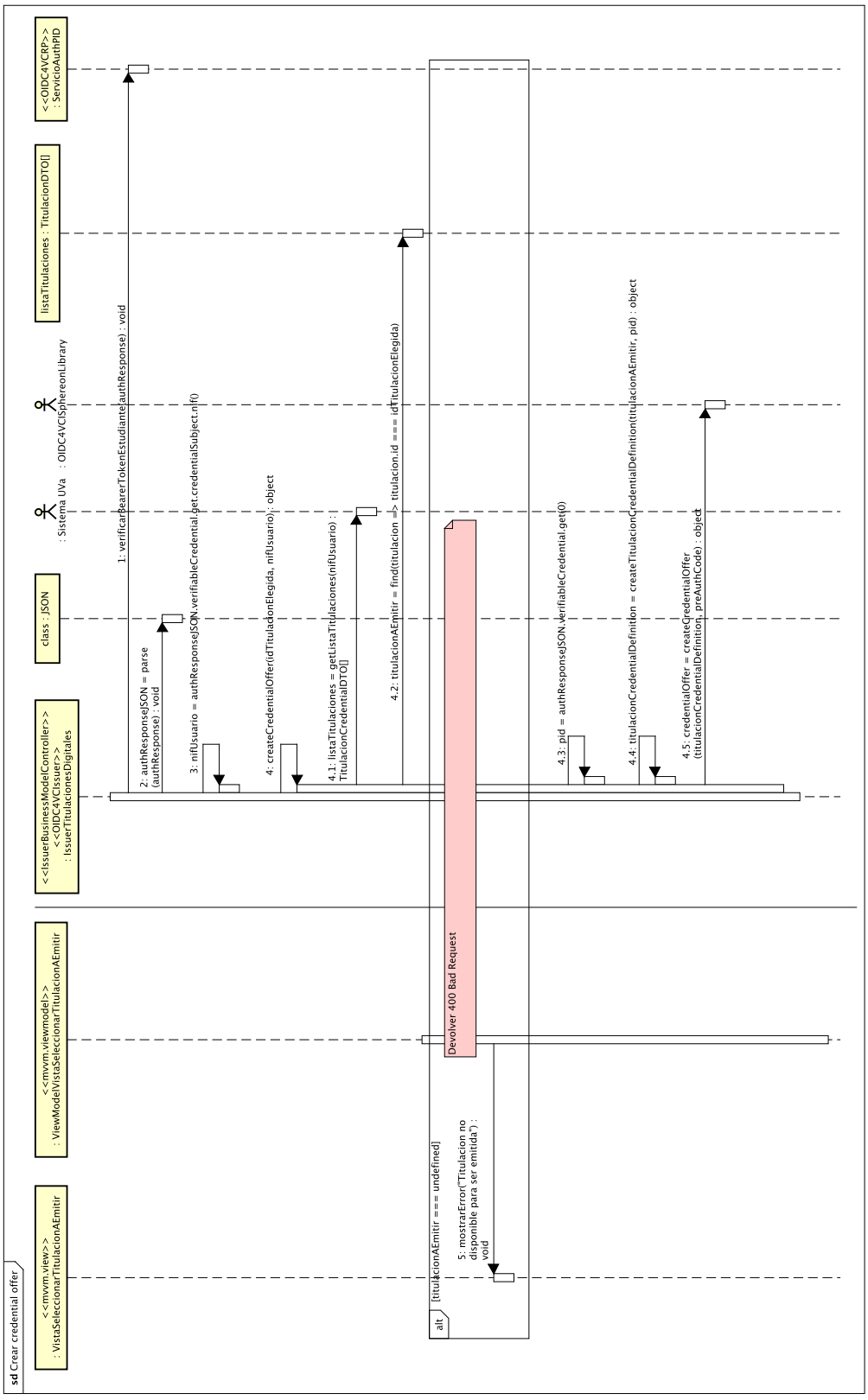


Figura 6.25: Diagrama de secuencia de creación de la *Credential Offer* durante la emisión de una titulacion digital.

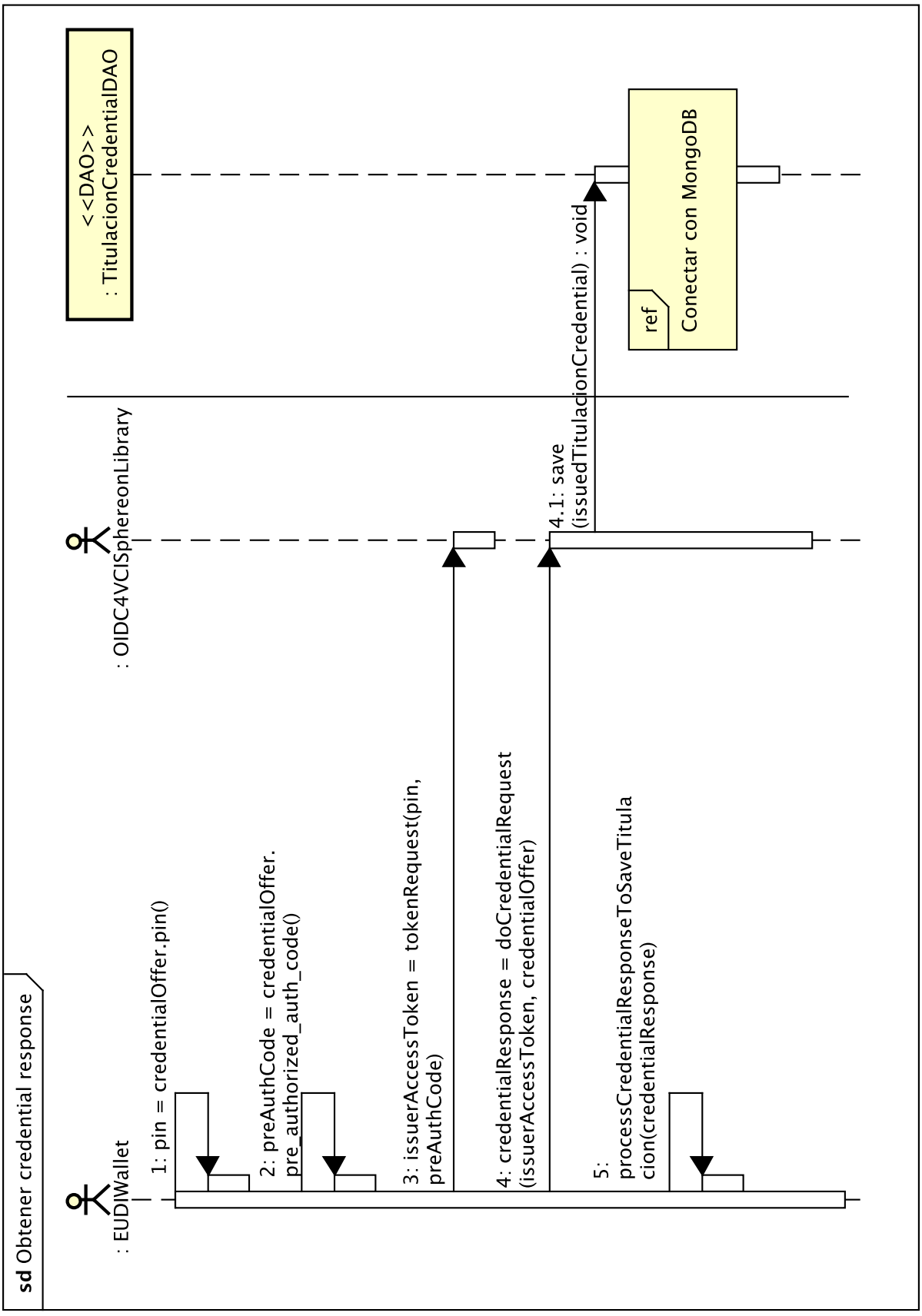


Figura 6.26: Diagrama de secuencia de la obtención de la *Credential Response* con la titulación digital emitida.

Figura 6.27: Credential Definition Titulación Digital.

6.2.4. CU4 Verificar posesión Titulación Digital

En la Figura 6.31 se puede ver el diagrama de secuencia en diseño de la verificación de la posesión de una titulacion digital a partir de la web de RRHH de una empresa de cara a que esta pueda comprobar la idoneidad del candidato con la certeza de que tiene determinadas titulaciones oficiales de la UVa. En primer lugar desde el *SistemaRRHHEmpresa* se produce una solicitud de generación de una *Authentication Request* para una titulacion concreta mediante su nombre. Internamente se resuelve el mombre proporcionado al tipo concreto de titulación digital que le corresponde para poder buscar la *Presentation Definition* correspondiente en la base de datos MongoDB de objetos de este tipo. En la Figura 6.33 se puede ver la presentation definition seleccionada en base al nombre de titulacion relacionado mediante el que se filtrarán las credenciales candidatas en la wallet del usuario para mostrar como opciones a presentar solo aquellas titulaciones que se correspondan con el tipo *TitulacionDigitalGradoFilosofia*, que es el que figura en este ejemplo. A nivel técnico lo que indica es que en campo *type* de las credenciales debe coincidir con el tipo *TitulacionDigitalGradoFilosofia* para poder ser presentada como credencial.

Una vez generada la auth request se devuelve al *SistemaRRHHEmpresa* para que sea mostrada mediante un QR o con un enlace con un deeplink que acabe siendo interpretado y utilizado desde la EUDIWallet del usuario. que participa en su proceso. Una vez se selecciona la titulación a presentar desde la wallet se recibe en el *ServicioAuthTitulacionCredential* la *Authentication Response* que contiene la credencial y, una vez verificada mediante la librería *OID4VPSphereonLibrary*, se envía al *SistemaRRHHEmpresa* para que disponga de los datos acerca del usuario y los datos de la titulación de la que ha demostrado disponer.

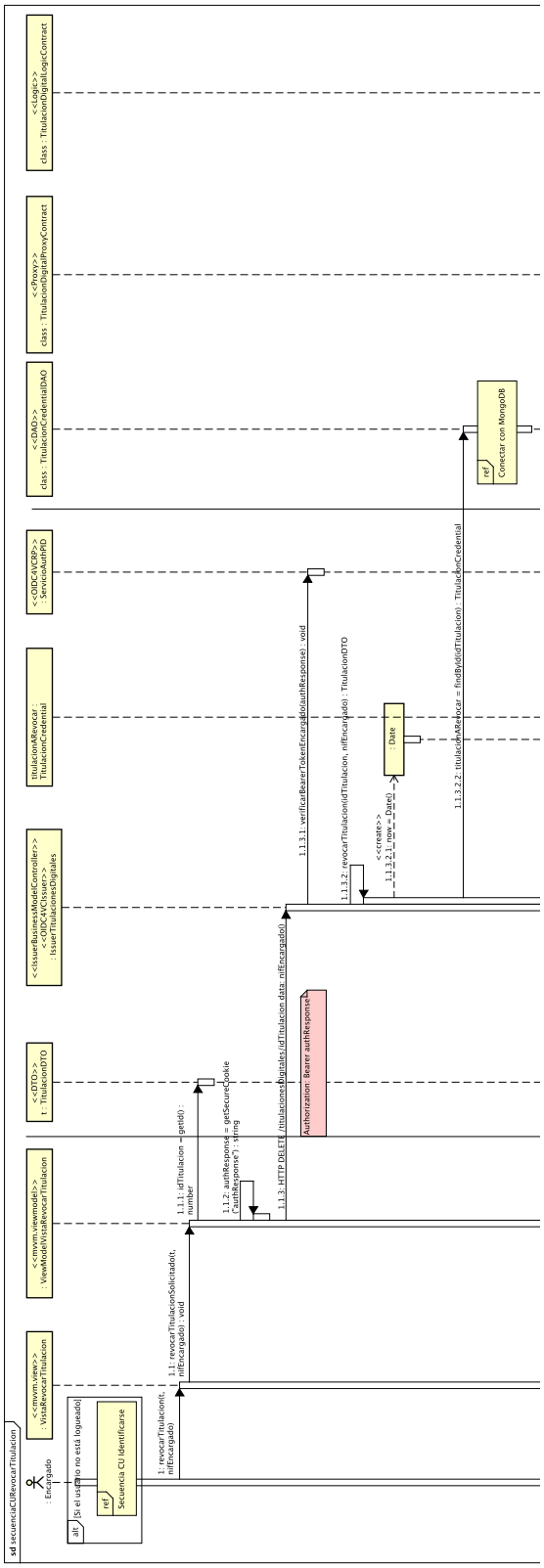


Figura 6.28: Diagrama de secuencia de revocación de una titulacion digital. Parte 1.

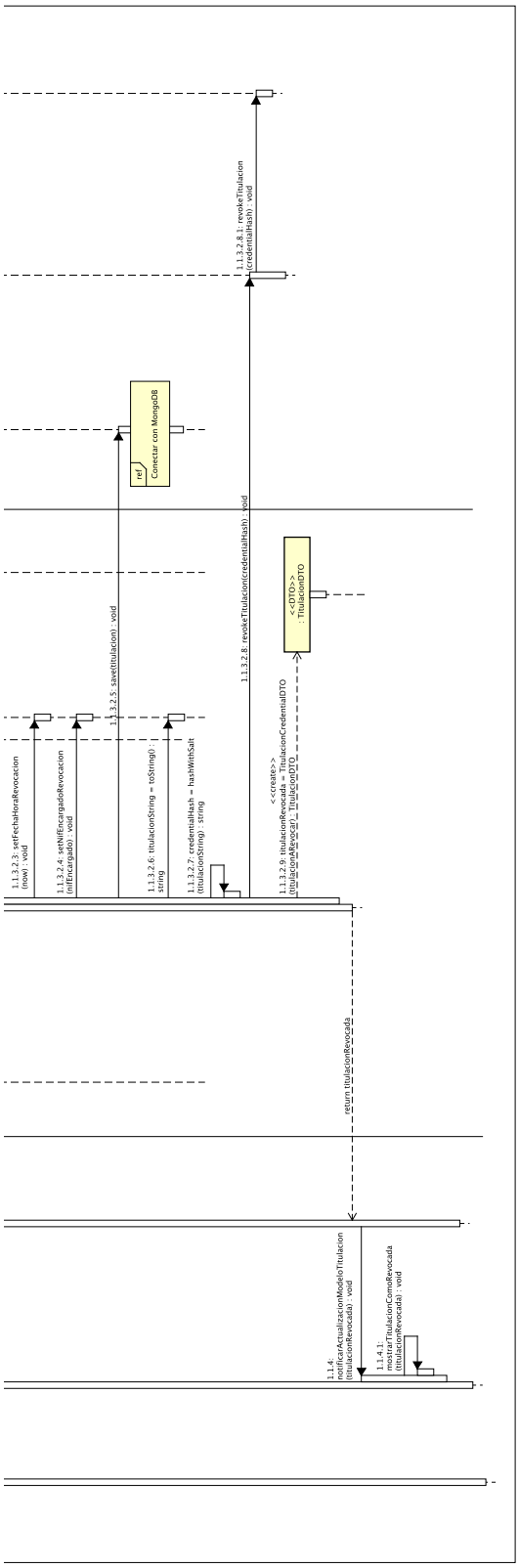


Figura 6.29: Diagrama de secuencia de revocación de una titulacion digital. Parte 2.

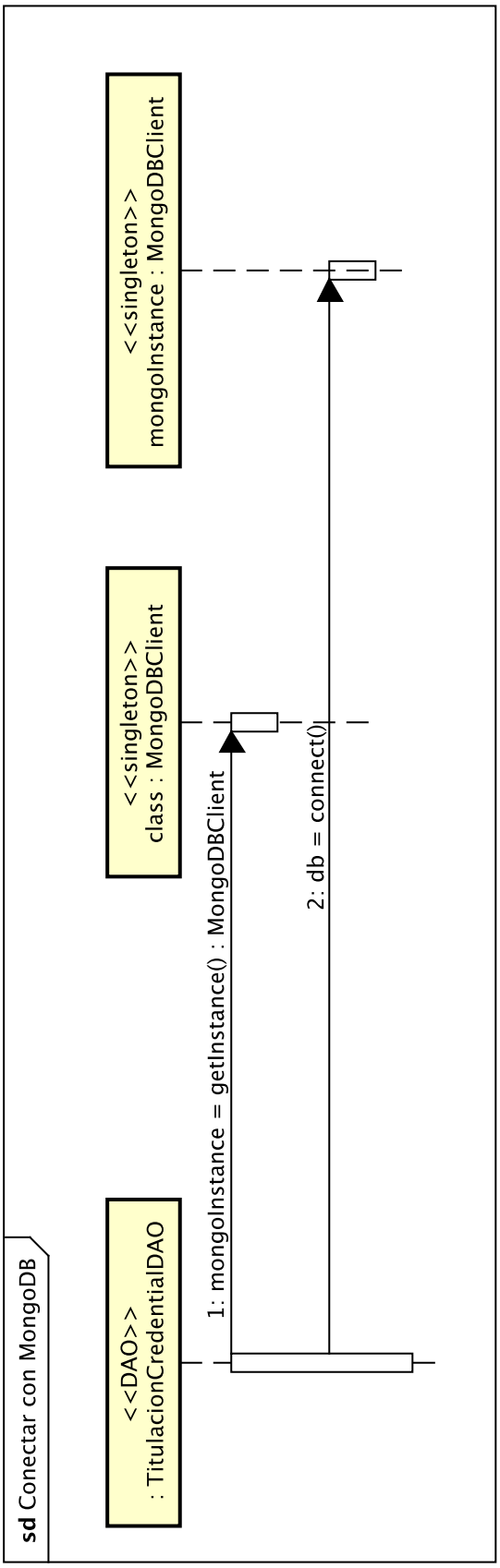


Figura 6.30: Diagrama de secuencia de la conexión con la BD MongoDB del issuer.

6.3. Diagrama de componentes

En la Figura 6.34 se puede ver el diagrama de componentes del emisor de titulaciones digitales. En la Figura 6.35 se puede ver el diagrama de componentes del verificador de titulaciones digitales. En ambos diagramas se pueden apreciar las conexiones con librerías de terceros. En el caso del emisor se puede ver que el componente *ServicioAuthPID* hace uso del componente de un tercero, *OID4VPSphereonLibrary* y el *OID4VCIssuer* hace uso del *OID4VCISphereonLibrary*.

6.4. Arquitectura Física del Sistema

En la Figura 6.36 se puede ver el diagrama de despliegue de la arquitectura física tanto del emisor de titulaciones como del verificador. Es importante destacar que ambos subsistemas conectan contra un nodo de Ethereum en Infura, un servicio online destinado a dar acceso a nodos de este tipo como servicio *SaaS* o *Software As A Service*. Esto es necesario tanto para que el emisor de titulaciones pueda almacenar el identificador hasheado de la titulaciones que son revocadas por un encargado a través del smart contract de *RevokationRegistry* como para que el verificador revise que no haya sido revocada a la hora de procesarla al haber sido presentada por un usuario a través de su EUDI Wallet.

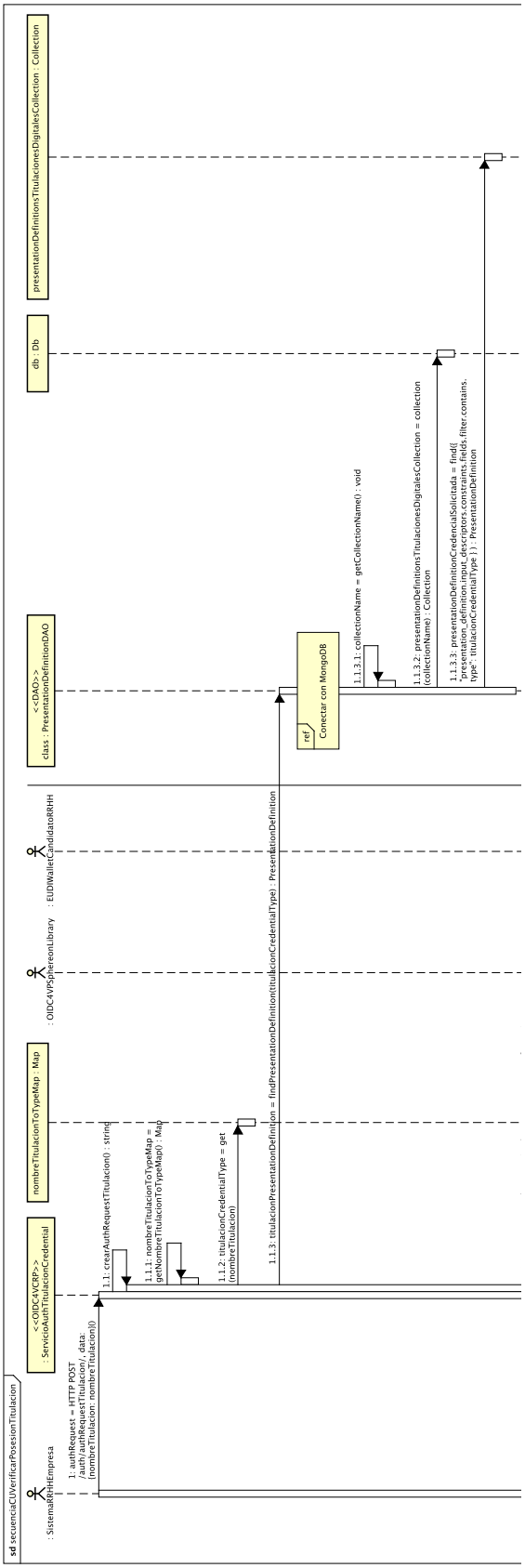


Figura 6.31: Diagrama de secuencia que verifica la posesion de una titulacion digital. Parte 1.

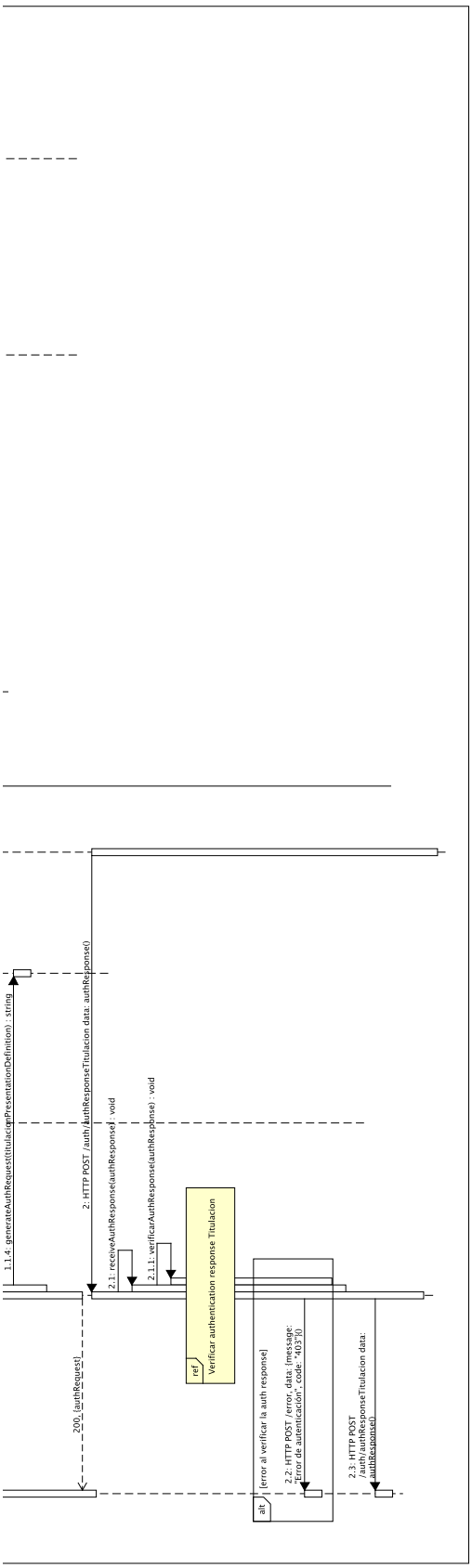


Figura 6.32: Diagrama de secuencia que verifica la posesion de una titulacion digital. Parte 2.

```
{
  "presentation_definition": {
    "id": "Titulacion Grado Filosofia Presentation Definition",
    "input_descriptors": [
      {
        "id": "Titulacion",
        "name": "Grado Filosofia Titulacion Digital VC",
        "purpose": "Request presentation of Grado Filosofia Titulacion Digital",
        "constraints": {
          "fields": [
            {
              "path": [
                "$.type"
              ],
              "filter": {
                "type": "array",
                "contains": {
                  "type": "string",
                  "pattern": "^TitulacionDigitalGradoFilosofia"
                }
              }
            }
          ]
        }
      }
    ]
  }
}
```


Figura 6.33: Presentation Definition Titulación Digital.

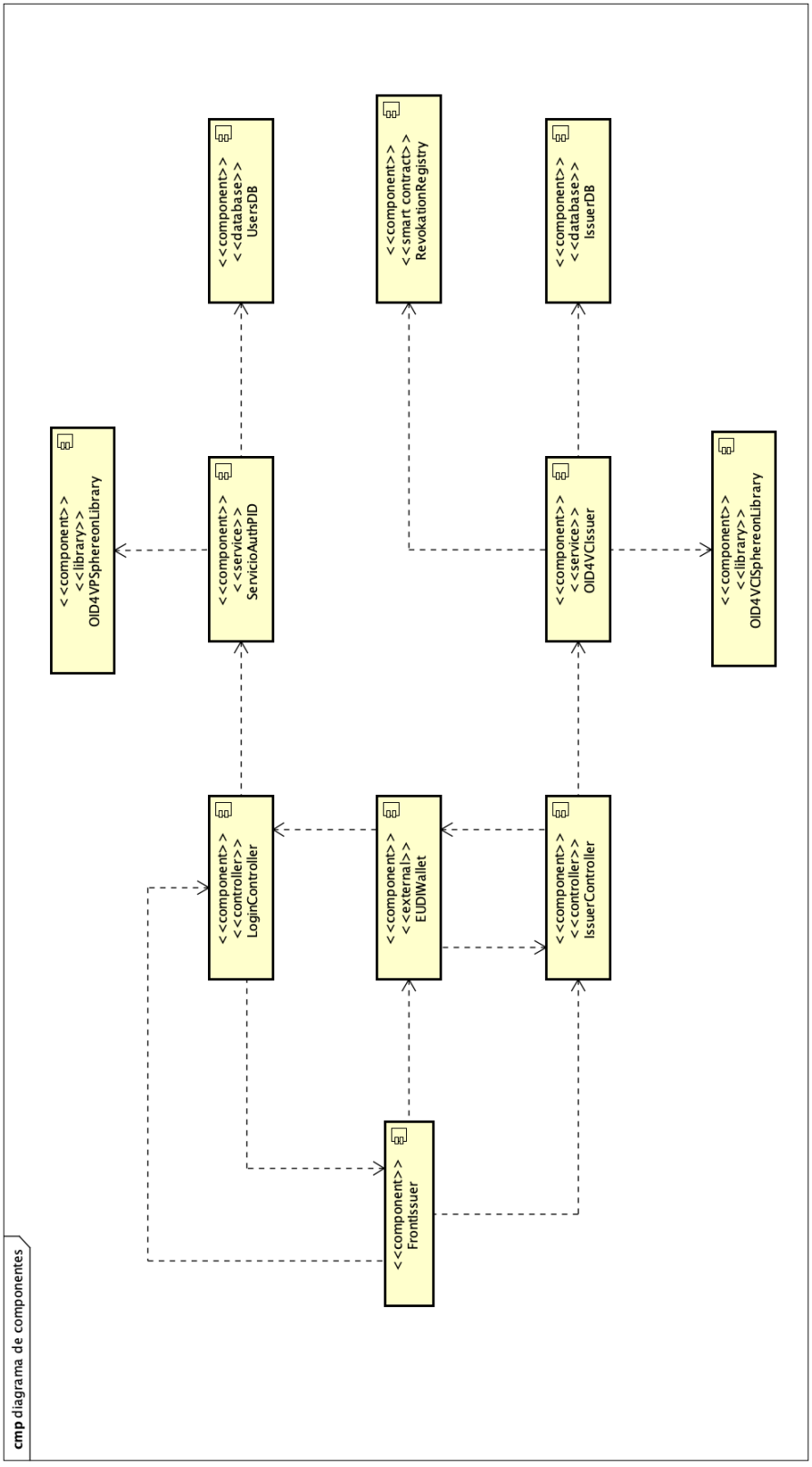


Figura 6.34: Diagrama de componentes del emisor de titulaciones digitales de la UVa.

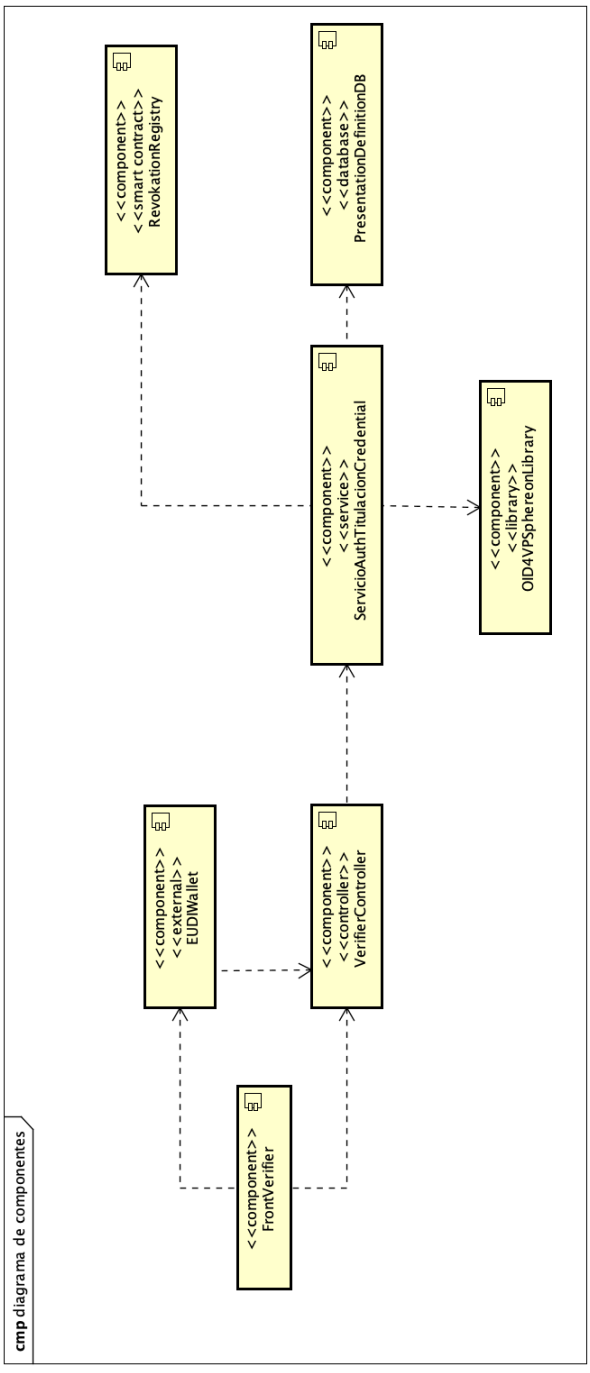


Figura 6.35: Diagrama de componentes del verificador de titulaciones digitales.

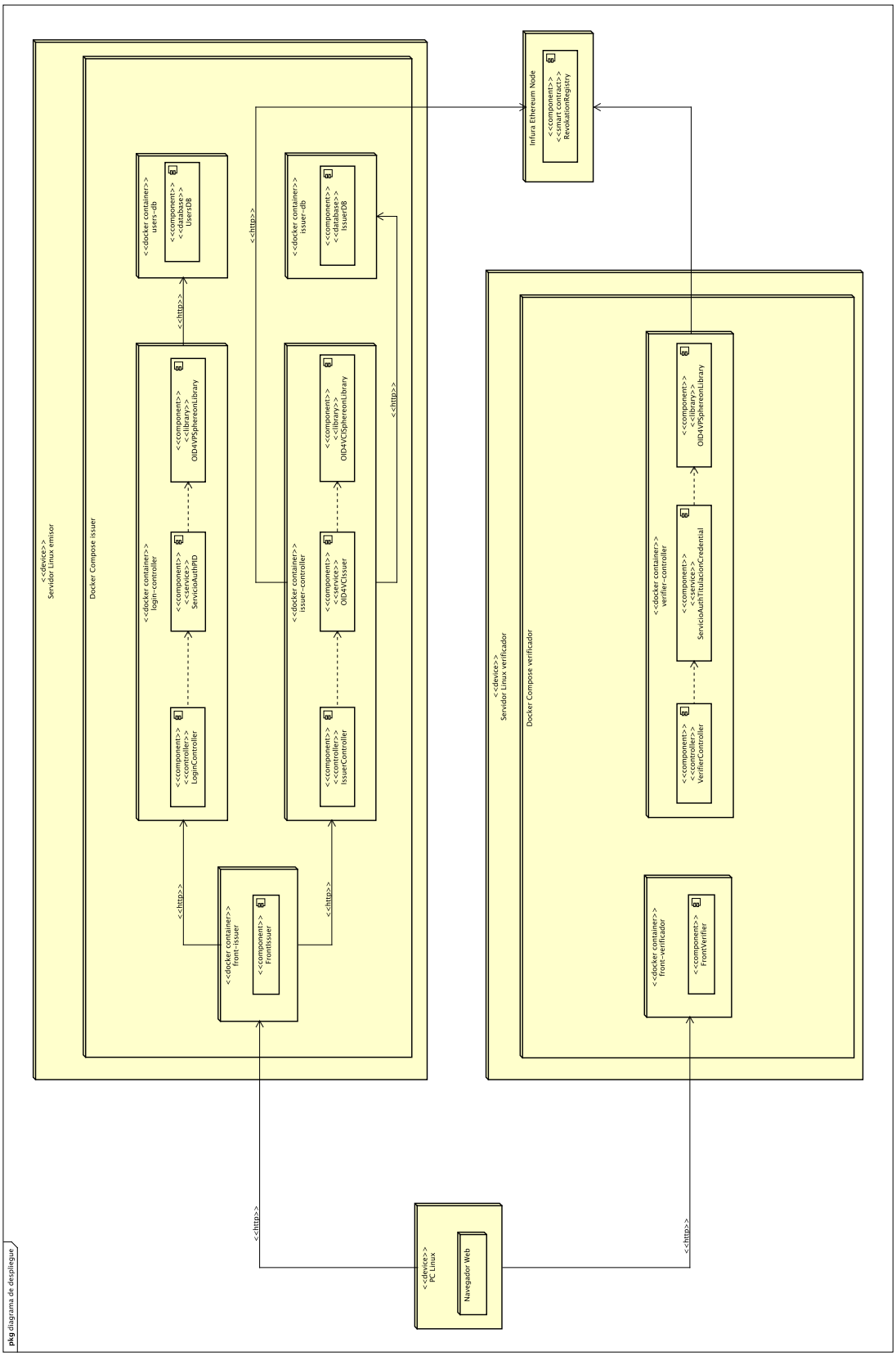


Figura 6.36: Diagrama de despliegue del emisor y del verificador de titulaciones digitales.

Capítulo 7

Implementación

En este capítulo se detalla la implementación del sistema de emisión y verificación de titulaciones digitales de la UVa.

7.1. Introducción

En cuanto a la implementación, se ha elaborado una Prueba de concepto o *Proof of concept* que demuestre la emisión de una titulación digital de la UVa con validez equiparable a su homólogo en papel y la verificación de la misma por un tercero.

En relación a lo expuesto, la credencial de tipo PID no se procesa a la hora de autenticar y autorizar al usuario. Se ha considerado que la autenticación y autorización del usuario no debe entrar en el alcance de la PoC. Por otro lado, por limitación de las bibliotecas existentes que implementan el stack de protocolos OIDC necesarios se ha considerado no utilizar firmas cualificadas, que requerírían el soporte de otro tipo de algoritmos a la hora de firmar y verificar las credenciales.

Por ser una PoC se obvia el control de errores centrándose el desarrollo de un caso válido, lo que se denomina un *Happy Path*. Tampoco se realiza *testing* unitario ni de integración, ni se ha realizado un despliegue en un entorno de producción. Por tanto tampoco hay una documentación que describa con detalle las pruebas del sistema.

Se han realizado algunas adaptaciones en la librería utilizada para los protocolos de emisión e intercambio de Credenciales Verificables ya que no eran del todo correctas y generales como se indica en los propios protocolos.

A continuación se detallan los lenguajes y frameworks utilizados para la implementación de la PoC, las herramientas utilizadas, los logs de la PoC, así como las capturas de pantalla relacionadas y las intrucciones para desplegar todos los componentes involucrados.

7.2. Tecnologías utilizadas

Para la implementación de la PoC se han utilizado los siguientes lenguajes y frameworks:

- **Nuxt JS**: Framework para el desarrollo de aplicaciones web con Vue.js, utilizado para la parte visual del emisor y del verificador.
- **Vue JS**: Framework JavaScript progresivo para construir interfaces de usuario, utilizado en el desarrollo de la interfaz del emisor y del verificador.
- **Nuxt UI**: Biblioteca de componentes para Nuxt.js que facilita el desarrollo de interfaces de usuario.
- **JavaScript**: Para la implementación de la parte visual del emisor y del verificador de titulaciones digitales.
- **Typescript**: Para la implementación de la lógica del emisor y del verificador de titulaciones digitales.
- **Solidity**: Para la implementación del smart contract de revocación de titulaciones digitales.
- **MongoDB**: Base de datos NoSQL utilizada para almacenar las credenciales emitidas y las definiciones de presentaciones esperadas.
- **PostgreSQL**: Base de datos relacional utilizada para almacenar los usuarios del sistema.

7.3. Herramientas utilizadas

En esta sección se detallan las herramientas utilizadas para la elaboración de este TFG.

7.3.1. Herramientas de planificación

Para la planificación se ha utilizado Taiga Project Management como soporte para organizar *sprints*, y Microsoft Teams para hacer el seguimiento del proyecto. También se ha utilizado Excel para disponer de un plan de actividades para tener una visión más general y de alto nivel del proyecto.

7.3.2. Herramientas de soporte para la información textual

Para elaborar esta memoria se utiliza LaTex desde el editor Visual Studio Code en Microsoft Windows 10. Para la elaboración de la presentación se utilizará Microsoft PowerPoint.

7.3.3. Herramientas de programación

Se ha utilizado Visual Studio Core para programar la prueba de concepto, junto con el uso de Docker, y la infrastructura provista como *Software As A Service* de Infura para disponer del acceso a un nodo blockchain en la red de test de Ethereum. Se han utilizado diferentes lenguajes de programación como Typescript, Solidity y JavaScript. Para el desarrollo de los contratos inteligentes se ha utilizado Solidity. También se ha hecho uso de frameworks para la parte visual como Nuxt JS, Vue JS y Nuxt UI.

7.4. Demostración de la emisión de credenciales con logs de uso

En primer lugar, tras loguearse como estudiante (Figura 7.1 y 7.2), al usuario se le muestran todas las titulaciones disponibles para ser emitidas en el sistema de la UVa (Figura 7.3). Sus logs se pueden ver en la Figura 7.4.

Una vez seleccionada la misma, desde el issuer, que en este caso es el componente *oidc4vci-issuer*, se genera la *Credential Offer* correspondiente a la titulación elegida (Figura 7.5, con logs en la Figura 7.6). Al llegar esta a la wallet por medio de la lectura de un QR o por un enlace y tras la aceptación del usuario para proceder con la emisión desde la wallet, esta da comienzo al proceso de emisión de la titulacion (Figura 7.7). Como no se dispone de la wallet como tal para esta PoC el componente *oidc4vci-client* es el que se encarga de emular la aceptación

y comportamiento de la wallet. Como parte de este proceso, desde la wallet se leen los metadatos del issuer necesarios para el proceso de emisión como tal, como es el *credential endpoint*, y formato de firma y de credencial que se utilizará, entre otros. Esto puede verse en la Figura 7.8. Por último se ataca al issuer desde la wallet para solicitarle el access token que permita emitir la credencial haciendo uso del PIN que se incluye en la Credential Offer aceptada junto con el pre authorization code de la misma. Este token se utiliza para dar comienzo a la *Credential Request*, cuya response culmina el proceso de emisión de la credencial (Figura 7.9). Los logs de este proceso se puede ver en la Figura 7.10 y 7.11 (7). La emisión de la titulacion como tal se puede ver en la Figura 7.12 y 7.13. A mayores de quedar almacenada en la wallet ficticia se ha añadido la descarga de un fichero .pkpass que contiene la titulacion emitida en formato de *passbook* para que pueda ser leída por una de las aplicaciones de wallet que soportan este formato, como *Pass2U Wallet* en Android. Simplemente abriendo el fichero descargado desde un dispositivo Android, y elegiendo la opción de abrirlo con la aplicación de *Pass2U Wallet* se importa la credencial emitida a la wallet, como se puede ver en la Figura 7.14 y 7.15.

Figura 7.1: Portal del emisor de titulaciones digitales de la UVa.

Figura 7.2: Seleccion de usuario en el portal del emisor de titulaciones digitales de la UVa.

Figura 7.3: Lista de titulaciones disponibles para ser emitidas en el portal del emisor de titulaciones digitales de la UVa.

Figura 7.4: Logs de la PoC del emisor de titulaciones digitales al loguearse como estudiante.

Figura 7.5: Credential Offer emitida en el portal del emisor de titulaciones digitales de la UVa.

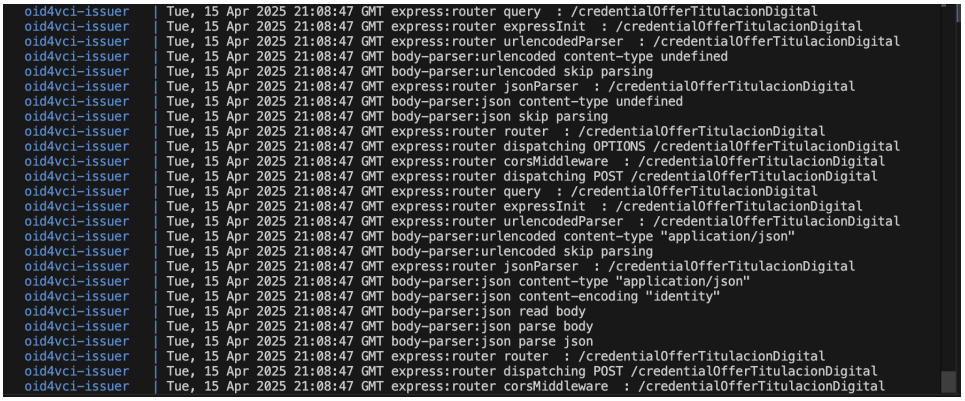


Figura 7.6: Logs de la PoC del emisor de titulaciones digitales al solicitar la emisión de una titulación digital.

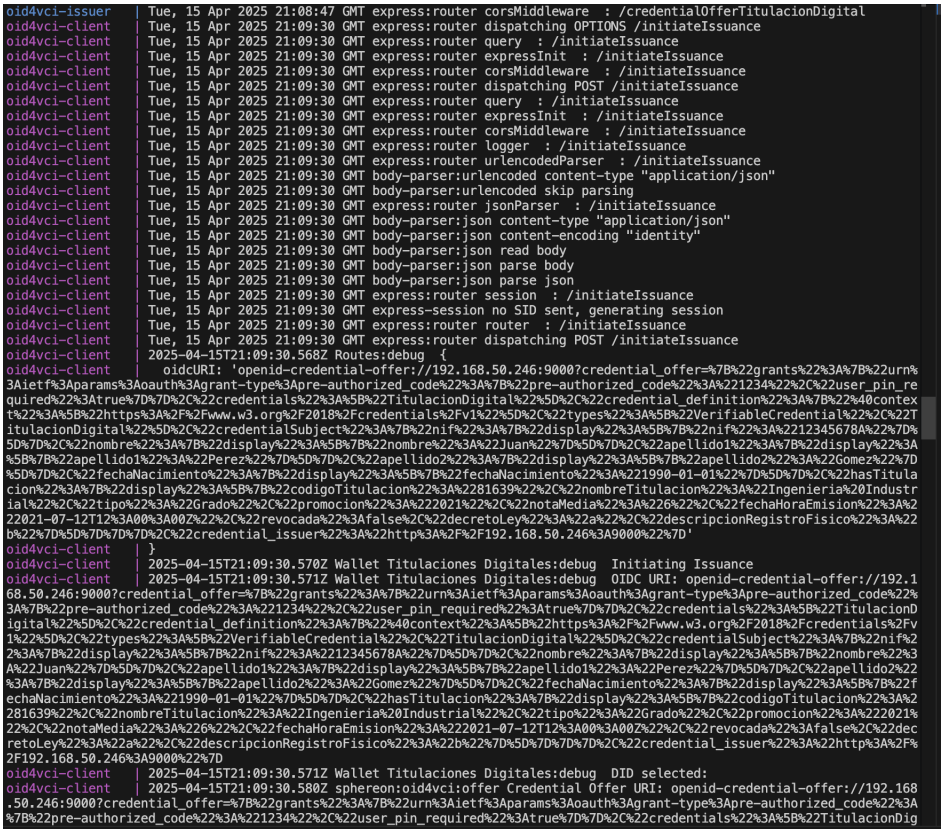


Figura 7.7: Logs de la PoC del emisor de titulaciones digitales al iniciar el proceso de emisión de una titulación digital.

7.5. Demostración de la revocación de credenciales con logs de uso

En primer lugar, tras loguearse como encargado, al usuario se le muestran todas las titulaciones digitales emitidas para poder seleccionar la que se desea revocar. En la Figura 7.16 se puede ver el panel de revocaciones de titulaciones digitales de la UVa.

Para revocar una titulación, el encargado selecciona la titulación que desea revocar a través de su slider concreto y selecciona ”revocar” para esa titulación. Tras pedírsele confirmación se procede a revocar la titulación. En la Figura 7.17 se puede ver la titulacion ya revocada. En las Figuras 7.18 y 7.19 se pueden ver los logs al revocar una titulación digital. En ellos se ve como se realiza la conexión con el nodo de Ethereum para revocar la titulación digital, además de llamadas que la marcan como revocada en la Base de Datos del issuer.

7.6. Demostración del verificador de credenciales con logs de uso

En primer lugar, al visitar la web de RRHH de la empresa que utiliza el verificador de titulaciones se solicita al sistema la generación de la correspondiente *SIOP Request* para solicitar la presentación de una credencial de tipo titulacion digital. Esta se muestra mediante un QR o un enlace con un deeplink que se puede leer desde la wallet del usuario. En la Figura 7.20 se puede ver el frontal al que accede el usuario y en la 7.21 los logs de este proceso. En los logs también se ve como se ha establecido un canal de eventos mediante *Server Side Events* (SSE) para que el frontal de RRHH pueda reaccionar ante la presentación de la titulación una vez se complete.

Una vez el usuario escanea el QR o accede al enlace, la wallet le muestra las credenciales que puede presentar. En este caso solo hay una titulación digital que se puede presentar, por lo que el usuario la selecciona y acepta presentarla. Tras esto la wallet genera y envía la *SIOP Response* al backend del verifier, el *oid4vp-rp* (Figura 7.22 y 7.23). Como se ve en los logs, entre otras comprobaciones, el verifier se conecta con el nodo de Ethereum para verificar que la credencial no ha sido revocada. Una vez verificada la credencial, se envía al sistema de RRHH de la empresa la *Authentication Response*, que contiene la credencial presentada. Ante esta situación, y con propósito de demostrar el flujo completo, el frontal del sistema de RRHH de la empresa recibe un evento por el canal de eventos Server Side Events (SSE) establecido anteriormente, que le indica que se ha recibido una titulacion digital

como se requería y reacciona en consecuencia mostrando la página que le indica al usuario que su acceso ha sido autorizado (Figura 7.24).

Como último flujo se tiene el acceso a información adicional por parte del propio usuario ante la cual se presenta la Authentication Response obtenida como forma de autenticar esa petición para acceder a un recurso determinado en el backend, como se puede ver en la Figura 7.25. Tras verificar esa Authentication Response y el estado de revocación de la titulación, el backend responde al usuario con la información solicitada (Figura 7.26).

7.7. Despliegue de *issuer* y *verifier*

Se ha creado un mono repositorio en *GitHub* para el despliegue de la PoC. Este se encuentra enlazado en el Apéndice C. Para desplegar tanto el emisor como el verificador se ha utilizado *Docker* y *Docker Compose*. Para ello se han creado tantas imágenes (y repositorios) como componentes a ser desplegados, una para cada uno de los componentes. En la raíz del repositorio se encuentra un fichero *docker-compose.yml* que permite desplegar todos los componentes de forma conjunta. Para ello es necesario tener instalado *Docker* y *Docker Compose* en el sistema. En el caso de no tenerlo instalado, se puede seguir la documentación oficial de *Docker* para su instalación.

Una vez instalado se debe ejecutar el siguiente comando en la terminal para desplegar todos los servicios necesarios:

```
docker-compose up
```

Esto construirá las imágenes necesarias y levantará los contenedores correspondientes para el emisor y el verificador de titulaciones digitales. Una vez completado, los servicios estarán disponibles en los puertos configurados en el archivo *docker-compose.yml*. El docker-compose en cuestión se puede ver en la Figura 7.27, 7.28 y 7.29. Los servicios que se despliegan son los siguientes:

1. **mongodb-service**: Este es el servicio que actúa como base de datos para el emisor de credenciales. Se encarga de almacenar las credenciales emitidas y su estado.

2. **hardhat**: Este es el servicio que actúa como nodo de Ethereum. Se encarga de gestionar la conexión con una red de Ethereum local y la verificación de credenciales.

3. **oidc4vci-issuer**: Este es el servicio que emite las credenciales digitales. Se encarga de gestionar la emisión de credenciales y su almacenamiento.

4. **oidc4vci-client**: Este es el servicio que actúa como cliente de la wallet. Se encarga de gestionar la interacción con la wallet y la emisión y presentación de credenciales.

5. **issuer-front**: Este es el servicio que actúa como frontal del emisor. Se encarga de gestionar la interfaz de usuario y la interacción con el emisor de credenciales.

6. **oid4vp-rp**: Este es el servicio que actúa como verificador de credenciales. Se encarga de gestionar la verificación de credenciales.

7. **rp-front**: Este es el servicio que actúa como frontal del verificador. Se encarga de gestionar la interfaz de usuario y la interacción con el verificador de credenciales.

En algunos servicios se hace referencia a la dirección *192.168.50.246*, que es la dirección IP de la máquina donde se ha realizado el despliegue. En caso de que se realice en otra máquina, es necesario cambiar esta dirección por la IP de la máquina. Esto es necesario porque si se utilizara directamente localhost entonces docker entendería que se referimos al contenedor específico que está ejecutando el código que requiere de esa variable en lugar de entender que se se está refiriendo a la máquina, concretamente a otro contenedor con el que intentamos comunicarnos. Tampoco es posible utilizar host.docker.internal ya que no es válida cuando la variable se resuelve desde el lado del cliente y no desde el componente desplegado en el contenedor como tal.

Figura 7.8: Logs de la PoC del emisor de titulaciones digitales al recuperar los metadatos del issuer.

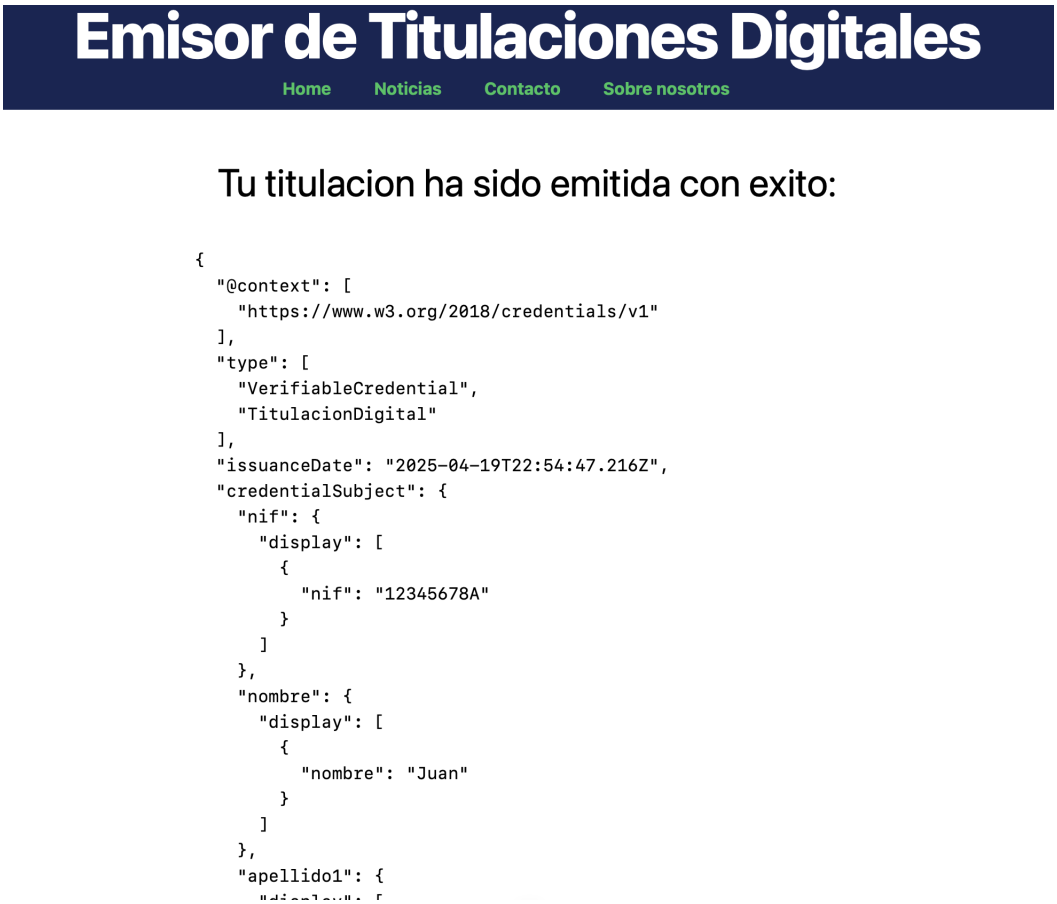


Figura 7.9: Titulacion emitida en el portal del emisor de titulaciones digitales de la UVa.

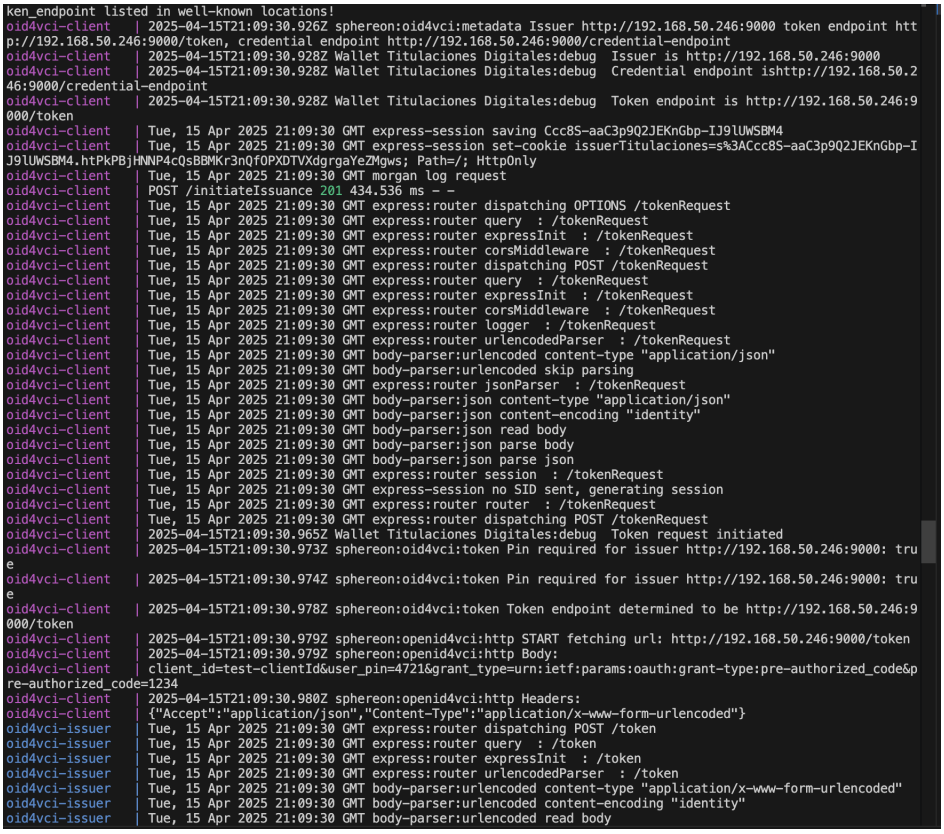


Figura 7.10: Logs de la PoC del emisor de titulaciones digitales al solicitar el access token para emitir la credencial.

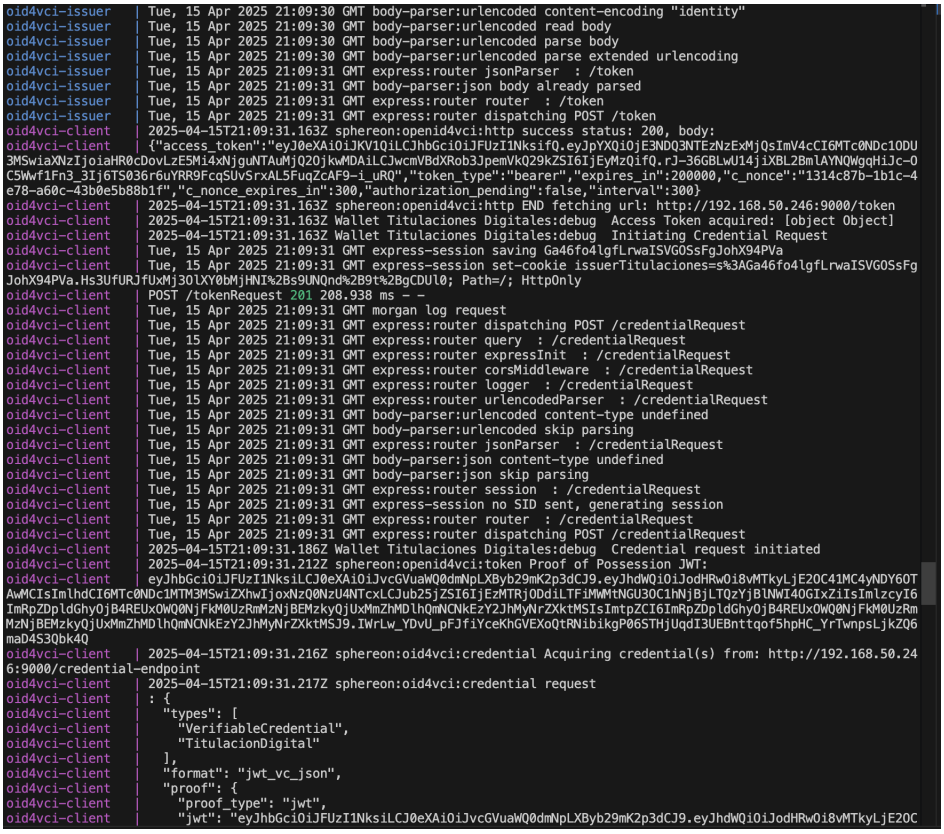


Figura 7.11: Logs de la PoC del emisor de titulaciones digitales al solicitar la emisión de la credencial.

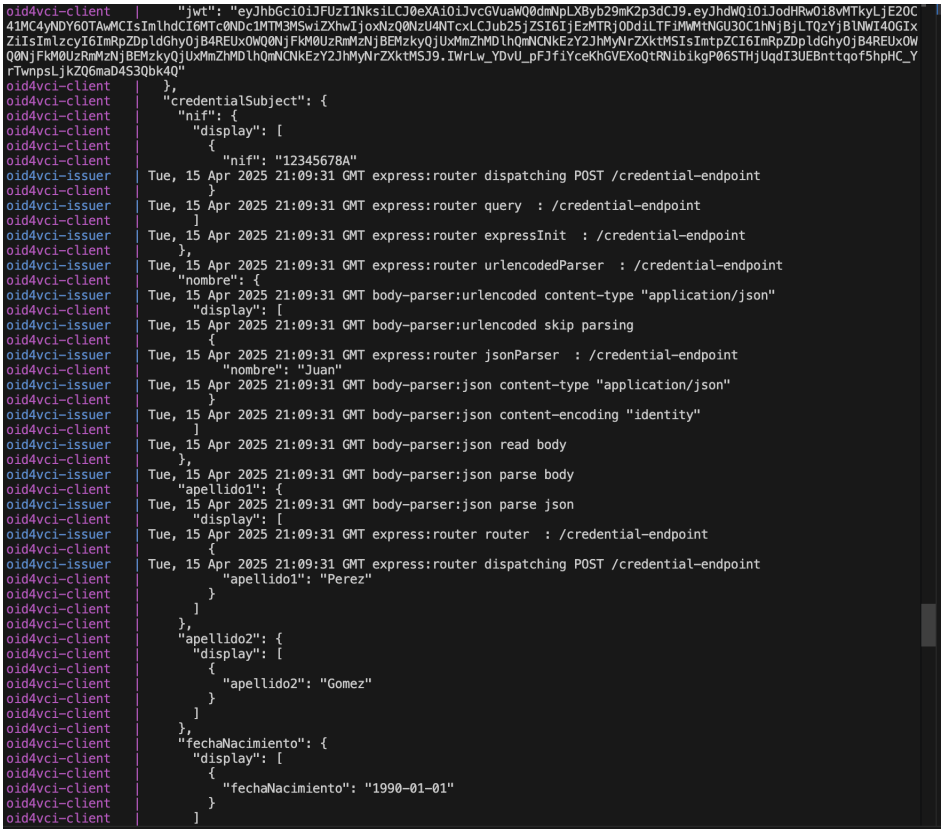


Figura 7.12: Logs de la PoC del emisor de titulaciones digitales al obtener la credencial emitida.

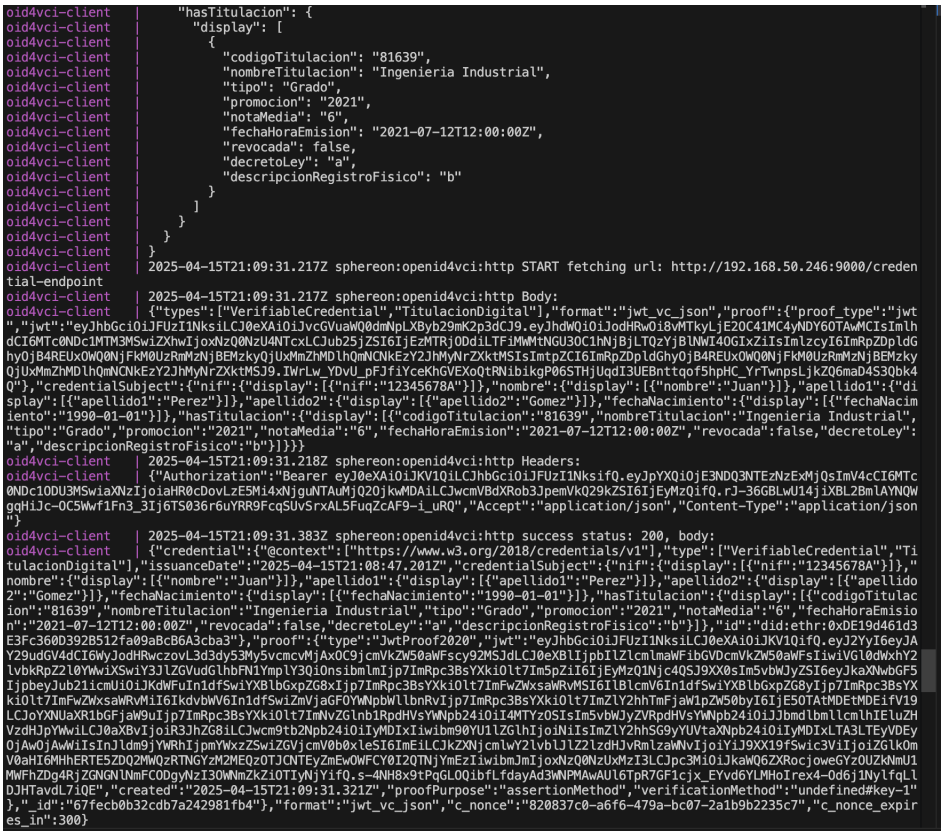


Figura 7.13: Logs de la PoC del emisor de titulaciones digitales al obtener la credencial emitida.

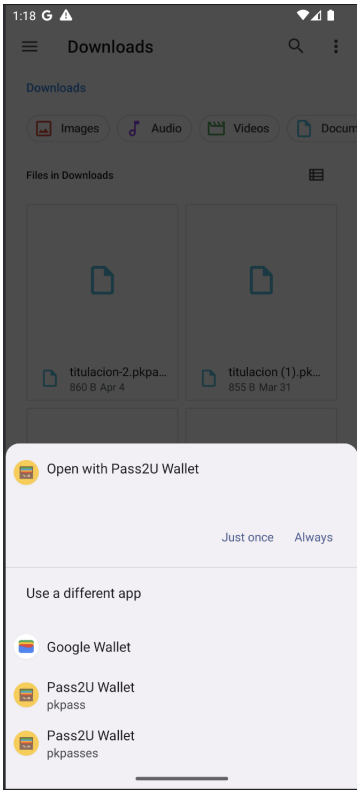


Figura 7.14: Abrir el fichero .pkpass con la aplicación Pass2U Wallet.

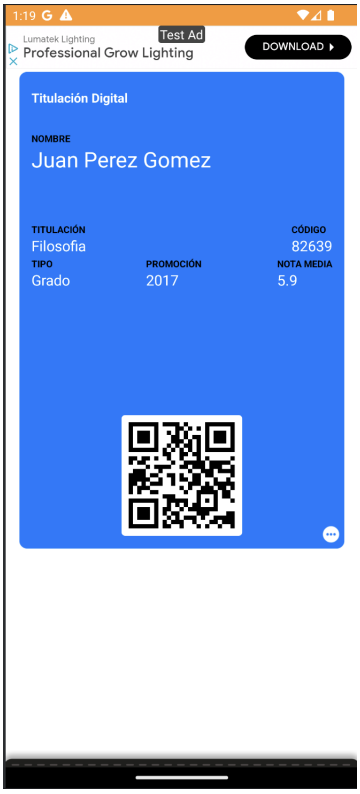


Figura 7.15: Credencial emitida en la aplicación Pass2U Wallet.

Figura 7.16: Panel de revocación de titulaciones digitales de la UVa.

Figura 7.17: Titulacion digital revocada en el panel de revocación de titulaciones digitales de la UVa.

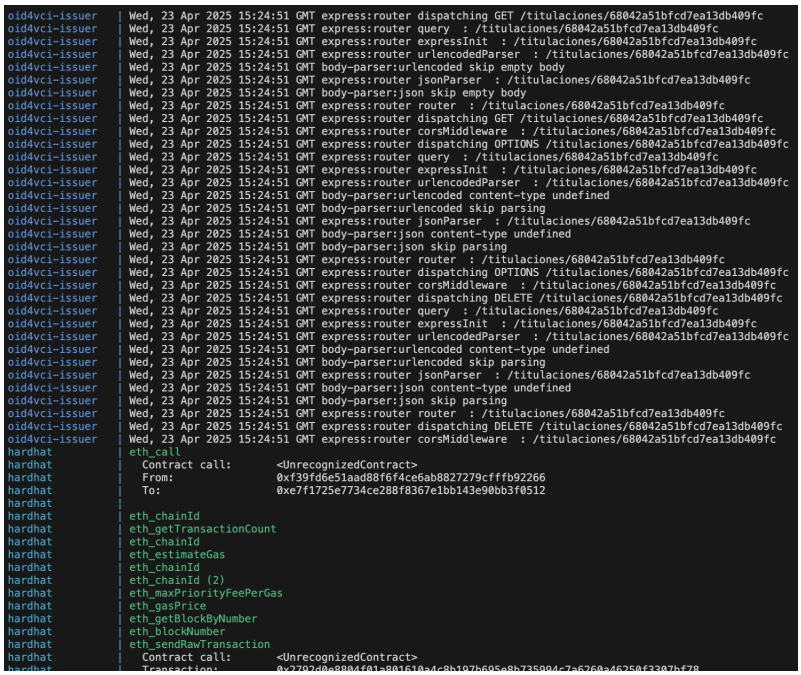


Figura 7.18: Logs de la PoC del emisor de titulaciones digitales al revocar una titulación digital.

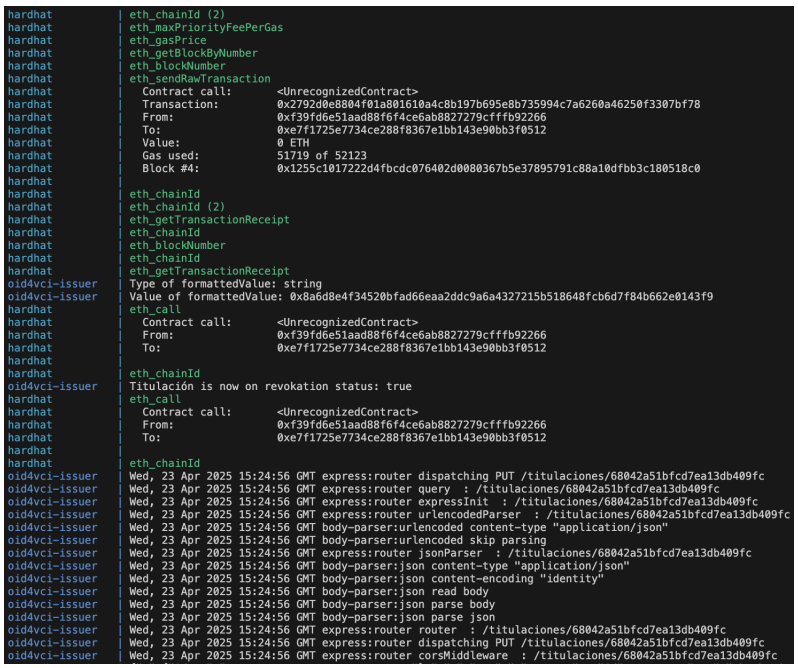


Figura 7.19: Logs de la PoC del emisor de titulaciones digitales al revocar una titulación digital.

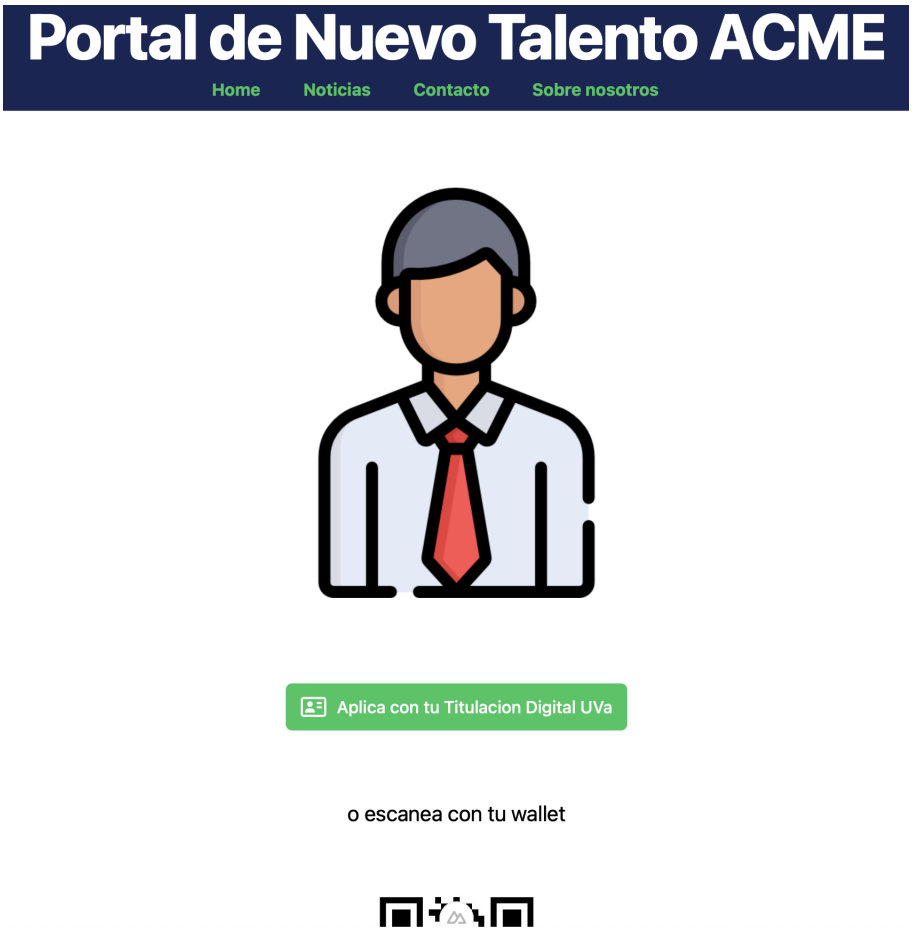


Figura 7.20: Frontal del verificador de titulaciones digitales de la UVa.

Figura 7.21: Logs de la PoC del verificador mostrando la generación de la SIOP Request.

Figura 7.22: Logs de la PoC del verificador mostrando la generación de la SIOP Response.

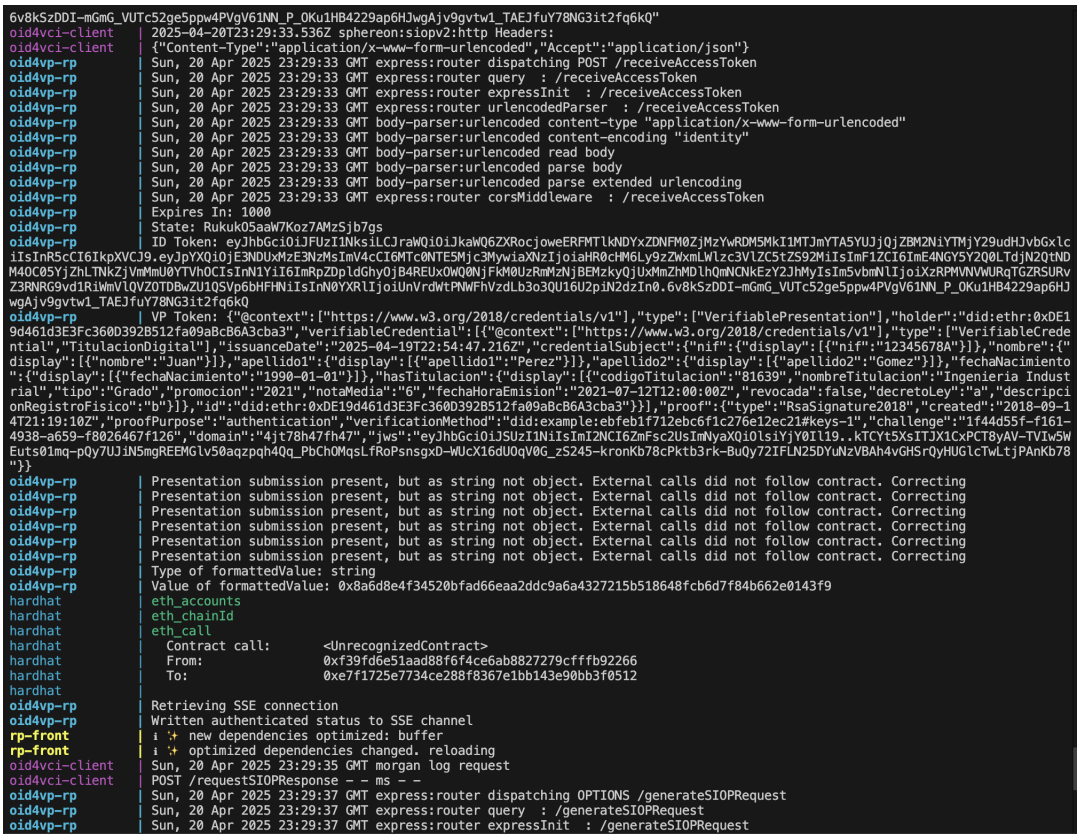


Figura 7.23: Logs de la PoC del verificador mostrando el envío de la SIOP Response al backend del verificador.

Figura 7.24: Frontal del verificador de titulaciones digitales de la UVa tras presentación satisfactoria de la titulacion digital.

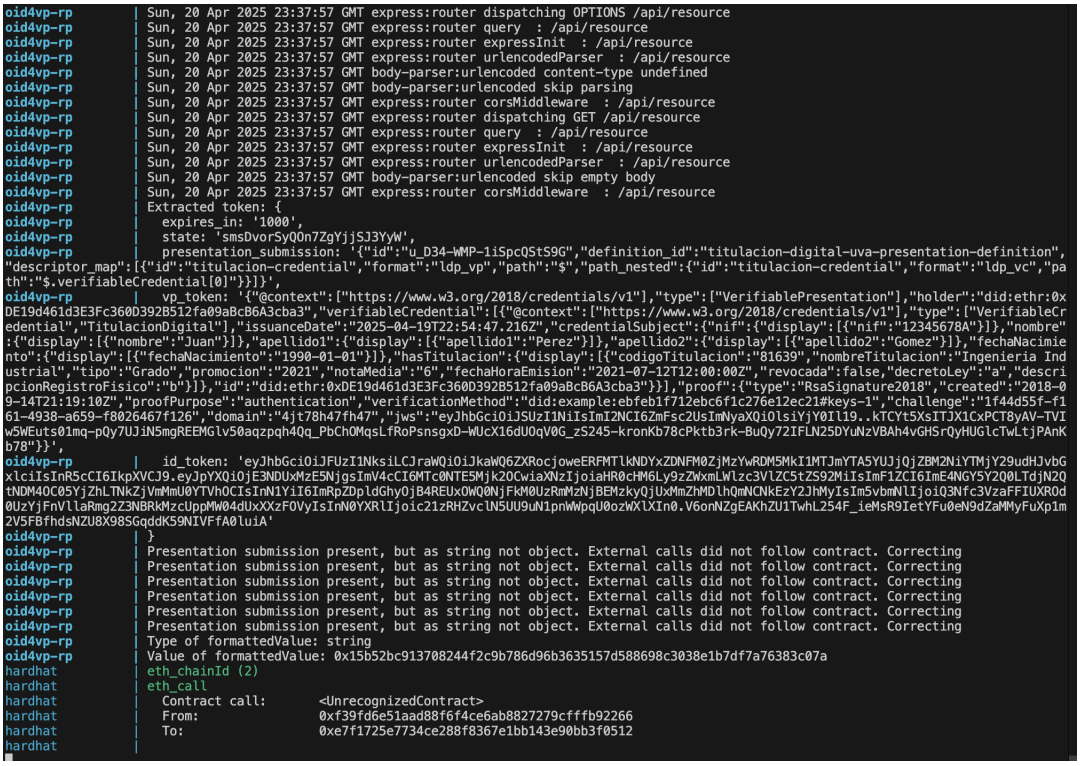
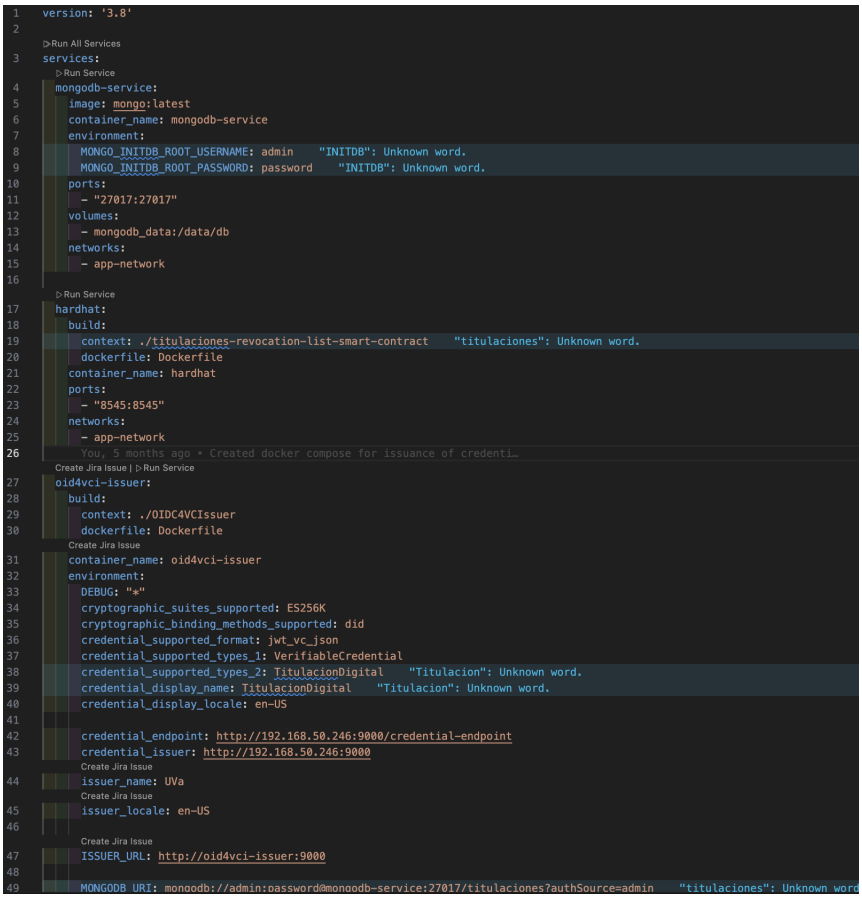


Figura 7.25: Logs de la PoC del verificador mostrando la verificación de la SIOP Response para acceder a información adicional.

Información MUY privada

Figura 7.26: Frontal del verificador de titulaciones digitales de la UVa mostrando información adicional tras la verificación de acceso.



```
version: '3.8'

Run All Services
services:
  Run Service
  mongodb-service:
    image: mongo:latest
    container_name: mongodb-service
    environment:
      MONGO_INITDB_ROOT_USERNAME: admin    "INITDB": Unknown word.
      MONGO_INITDB_ROOT_PASSWORD: password    "INITDB": Unknown word.
    ports:
      - "27017:27017"
    volumes:
      - mongodb_data:/data/db
    networks:
      - app-network

  Run Service
  hardhat:
    build:
      context: ./titulaciones-revocation-list-smart-contract    "titulaciones": Unknown word.
      dockerfile: Dockerfile
    container_name: hardhat
    ports:
      - "8545:8545"
    networks:
      - app-network
      You, 5 months ago • Created docker compose for issuance of credenti…
  Create Jira Issue | Run Service
  oid4vci-issuer:
    build:
      context: ./OIDC4VCIssuer
      dockerfile: Dockerfile
    Create Jira Issue
    container_name: oid4vci-issuer
    environment:
      DEBUG: "*"
      cryptographic_suites_supported: ES256K
      cryptographic_binding_methods_supported: did
      credential_supported_format: jwt_vc_json
      credential_supported_types_1: VerifiableCredential
      credential_supported_types_2: TitulacionDigital    "Titulacion": Unknown word.
      credential_display_name: TitulacionDigital    "Titulacion": Unknown word.
      credential_display_locale: en-US

      credential_endpoint: http://192.168.50.246:9000/credential-endpoint
      credential_issuer: http://192.168.50.246:9000
      Create Jira Issue
      issuer_name: UVa
      Create Jira Issue
      issuer_locale: en-US

      Create Jira Issue
      ISSUER_URL: http://oid4vci-issuer:9000

      MONGODB_URI: mongodb://admin:password@mongodb-service:27017/titulaciones?authSource=admin    "titulaciones": Unknown word
```

Figura 7.27: docker-compose del despliegue de la PoC - Parte 1.

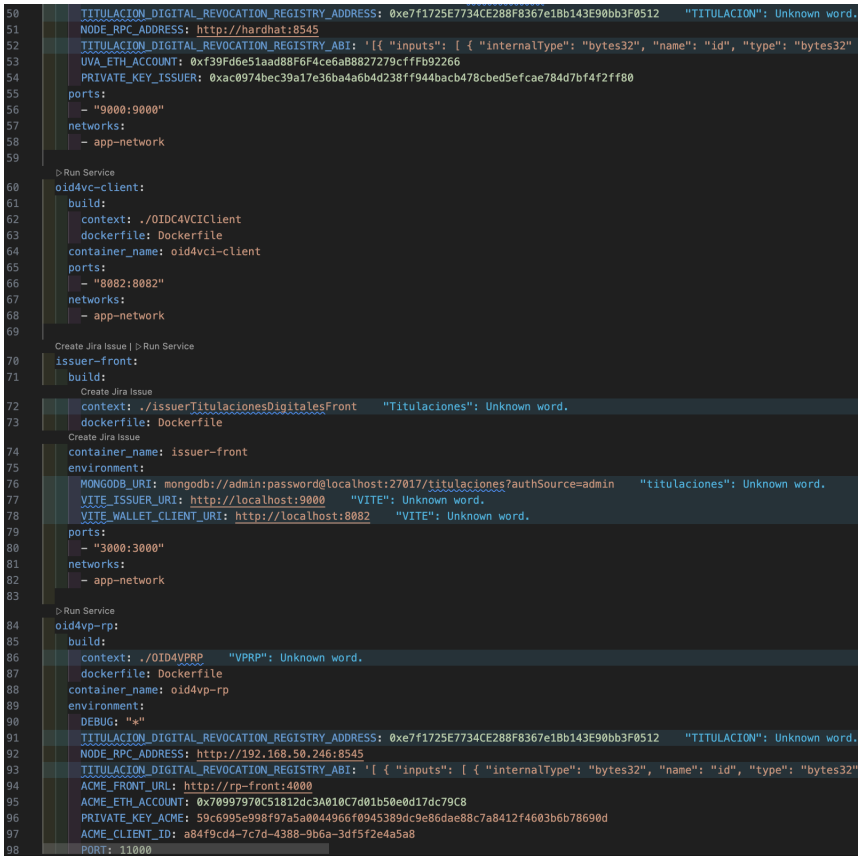


Figura 7.28: docker-compose del despliegue de la PoC - Parte 2.

Figura 7.29: docker-compose del despliegue de la PoC - Parte 3.

Capítulo 8

Conclusiones

En este capítulo se recogen las conclusiones obtenidas y las líneas de trabajo que han quedado pendientes como mejoras futuras.

8.1. Conclusiones

En esta sección se recogen las conclusiones obtenidas tras la realización de este TFG y se relacionan con cada uno de los objetivos planteados en la sección 1.4 para evaluar el grado de consecución de los mismos.

1. En cuanto al objetivo 1, la necesidad de un nuevo modelo de Identidad Digital Autosoberana quedó patente en el propio capítulo de introducción, donde se expusieron las limitaciones de los modelos actuales de Identidad Digital.
2. En cuanto al objetivo 2, también en el capítulo de introducción se justificó la importancia de la utilización de la tecnología blockchain en la Identidad Digital Autosoberana. Como establecía el objetivo fijado también se utilizó la misma en el desarrollo para que el verificador pueda comprobar que la credencial no ha sido revocada por el emisor sin tener que contactar con este.
3. En cuanto al objetivo 3, en el capítulo 3 se detalló el modelo teórico necesario para poder desarrollar un sistema basado en Identidad Digital Autosoberana.

4. En cuanto al objetivo 4, en los capítulos 5 y 6 se ha realizado el análisis y diseño de un sistema de emisión y verificación de titulaciones digitales de la UVa basado en Identidad Digital Autosoberana.

5. En cuanto al objetivo 5, se ha desarrollado una prueba de concepto que demuestra la emisión y verificación de titulaciones digitales de la UVa.

Por lo tanto, se puede concluir que se han cumplido los objetivos planteados al inicio de este TFG, y que se ha conseguido desarrollar un sistema de emisión y verificación de titulaciones digitales de la UVa basado en Identidad Digital Autosoberana.

8.2. Trabajo futuro

Algunas cuestiones se han dejado planteadas como mejoras futuras de este modelo de Identidad Digital Autosoberana por su poco grado de madurez y complejidad. Las siguientes secciones detallan cada una de ellas.

8.2.1. Revocación de Presentaciones

Se ha comentado que es posible que se refleje la revocación de Credenciales en una red blockchain, pero no se ha hablado de la posibilidad de revocar las presentaciones de credenciales que se han ido haciendo a determinados proveedores de servicios. Que el usuario pudiera disponer en su wallet de un histórico de los proveedores a los que ha ido presentando sus credenciales y que pueda revocar la entrega de la información que estas representan facilitaría enormemente el ejercicio del derecho al olvido. Este derecho muchas veces no se ejerce por ser demasiado tedioso y con procedimientos distintos según cada entidad con la que se contacte. Permitir que el usuario deje reflejado en la red *blockchain* que ha decidido revocar cierta presentación de credenciales podría servir como evidencia de que lo ha solicitado y que si se descubre que la entidad ha seguido tratando los datos de este se pueda realizar la sanción correspondiente. Esto es posible gracias a que la *blockchain* garantiza la integridad de esos registros, ya que nada de lo que se guarda en ella se puede borrar.

El modelo de Identidad Digital Autosoberana que propone el consorcio de Blockchain *Alastria*, *AlastriaID* [21], ya implementa esta funcionalidad.

8.2.2. Selective disclosure

El usuario puede presentar credenciales pero lo ideal sería presentar justo los atributos concretos almacenados en estas que sean requeridos para cada situación para cumplir con el principio de *minimización de datos*, entregando solo los datos verdaderamente imprescindibles para recibir los servicios requeridos. Una de las barreras para su implantación sería que se podrían crear con mayor facilidad combinaciones fraudulentas de atributos para asertar cosas que pueden ser falsas, ya que una combinación de varios datos no necesariamente significa lo mismo que significa cada dato por separado, pueden dar lugar a nuevos significados que sean falsos.

8.2.3. *Zero Knowledge Proofs*

Las *Zero Knowledge Proofs* o *pruebas de conocimiento cero* serían un paso más allá de la selective disclosure ya que en teoría permiten presentar pruebas de que se tiene determinada credencial o atributos sin desvelar esas credenciales o atributos haciendo uso de técnicas de criptografía avanzadas.

Bibliografía

[1] "Certificación Scrum Foundations," *EuropeanScrum*, accessed April 10, 2022. [Online]. Available: https://www.europeanscrum.org/certificacion-scrum-foundations.html

[2] D. Butler and N. Devranoglu, "Turkey's currency crisis deepens after Erdogan's latest rate cut," *Reuters*, 2021, accessed April 10, 2022. [Online]. Available: https://www.reuters.com/markets/rates-bonds/turkish-lira-trades-near-record-low-after-rate-cut-fuels-slide-2021-12-17/

[3] D. Poblete, "Web 3.0: Definición, ejemplos y evolución," accessed January 22, 2023. [Online]. Available: https://www.inversionsimple.com/web-3-0-definicion-ejemplos-y-evolucion/

[4] R. Chan, "The Cambridge Analytica whistleblower explains how the firm used Facebook data to sway elections," *Bussiness Insider*, accessed April 15, 2022. [Online]. Available: https://www.businessinsider.com/cambridge-analytica-whistleblower-christopher-wylie-facebook-data-2019-10

[5] "20 biggest GDPR fines so far 2019, 2020, 2021, 2022," *Data Privacy Meaanager*, accessed April 14, 2022. [Online]. Available: https://dataprivacymanager.net/5-biggest-gdpr-fines-so-far-2020/

[6] "Programador junior salarios," *Jooble*, accessed April 16, 2022. [Online]. Available: https://es.jooble.org/salary/programador-junior

[7] "¿Qué es la Identidad digital?" *Gobierno de Canarias*, accessed April 21, 2022. [Online]. Available: https://www3.gobiernodecanarias.org/medusa/ecoescuela/seguridad/identidad-digital-profesorado/que-es-la-identidad-digital/

[8] M. Sporny, D. Longley, and D. Chadwick, "Verifiable Credentials Data Model 1.1," *World Wide Web Consortium (W3C)*, accessed May 3, 2022. [Online]. Available: https://www.w3.org/TR/vc-data-model/

[9] M. Sporny, D. Longley, M. Sabadello, D. Reed, O. Steele, and C. Allen, “Decentralized Identifiers (DIDs) v1.0 Core architecture, data model, and representations,” *World Wide Web Consortium (W3C)*, accessed May 5, 2022. [Online]. Available: https://www.w3.org/TR/did-core/

[10] N. Sakimura, NRI, J. Bradley, P. Identity, M. Jones, Microsoft, B. de Medeiros, Google, Mortimore, and Salesforce, “OpenID Connect Core 1.0,” *OpenID Foundation*, accessed January 29, 2023. [Online]. Available: https://openid.net/specs/openid-connect-core-1_0.html

[11] E. D. Hardt and Microsoft, “The OAuth 2.0 Authorization Framework,” *IETF RFC*, accessed January 29, 2023. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6749

[12] K. Yasuda, M. Jones, and T. Lodderstedt, “Self-Issued OpenID Provider v2,” *OpenID Foundation*, accessed October 26, 2022. [Online]. Available: https://openid.net/specs/openid-connect-self-issued-v2-1_0.html

[13] T. Lodderstedt, K. Yasuda, and T. Looker, “OpenID for Verifiable Credentials Issuance,” *OpenID Foundation*, accessed September 28, 2022. [Online]. Available: https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html

[14] O. Terbu, T. Lodderstedt, K. Yasuda, A. Lemmon, and T. Looker, “OpenID for Verifiable Presentations,” *OpenID Foundation*, accessed September 28, 2022. [Online]. Available: https://openid.net/specs/openid-4-verifiable-presentations-1_0.html

[15] ——, “ISO/IEC 18013-5:2021 Personal identification — ISO-compliant driving licence — Part 5: Mobile driving licence (mDL) application,” *International Organization for Standardization*, accessed February 20, 2023. [Online]. Available: https://www.iso.org/standard/69084.html

[16] D. Buchner, B. Zundel, M. Riedel, K. H. Duffy, D. McGrogan, G. Cohen, O. Steele, W. Chang, D. Chadwick, J. Hensley, N. Klomp, and A. Kesselman, “Presentation Exchange 2.0.0,” *Digital Identity Foundation*, accessed February 5, 2023. [Online]. Available: https://identity.foundation/presentation-exchange/spec/v2.0.0/

[17] “Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC,” *European Parliament*, accessed March 22, 2023. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv%3AOJ.L_.2014.257.01.0073.01.ENG

[18] “Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL amending Regulation (EU) No 910/2014,” *European Commision*, accessed March 21, 2023. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021PC0281

[19] “European Digital Identity Architecture and Reference Framework Outline,” *European commission recommendation*, accessed September 21, 2023. [Online]. Available: https://ec.europa.eu/newsroom/dae/redirection/document/83643

[20] “Chapter 5 Issuers Trust Model,” *European Blockchain Services Infrastructure*, accessed September 28, 2022.

[21] “Alastria presenta la versión 2.0 de su modelo de identidad digital.” *Alastria*, accessed April 22, 2022. [Online]. Available: https://alastria.io/alastria-presenta-la-version-2-0-de-su-modelo-de-identidad-digital/

Anexo A

Ejemplo de Credencial y Presentación Verificable W3C

A.1. Credencial Verificable W3C

```
{
// set the context, which establishes the special terms we will be using
// such as 'issuer' and 'alumniOf'.
"@context": [
  "https://www.w3.org/2018/credentials/v1",
  "https://www.w3.org/2018/credentials/examples/v1"
],
// specify the identifier for the credential
"id": "http://example.edu/credentials/1872",
// the credential types, which declare what data to expect in the credential
"type": ["VerifiableCredential", "AlumniCredential"],
// the entity that issued the credential
```

```
"issuer": "https://example.edu/issuers/565049",
// when the credential was issued
"issuanceDate": "2010-01-01T19:23:24Z",
// claims about the subjects of the credential
"credentialSubject": {
  // identifier for the only subject of the credential
  "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
  // assertion about the only subject of the credential
  "alumniOf": {
    "id": "did:example:c276e12ec21ebfeb1f712ebc6f1",
    "name": [{
      "value": "Example University",
      "lang": "en"
    }]
  }
},
// digital proof that makes the credential tamper-evident
"proof": {
  // the cryptographic signature suite that was used to generate the signature
  "type": "RsaSignature2018",
  // the date the signature was created
  "created": "2017-06-18T21:19:10Z",
  // purpose of this proof
  "proofPurpose": "assertionMethod",
  // the identifier of the public key that can verify the signature
  "verificationMethod": "https://example.edu/issuers/565049#key-1",
  // the digital signature value
  "jws": "eyJhbGciOiJSUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..TCYt5X
```

```
        sITJX1CxPCT8yAV-TVkIEq_PbChOMqsLfRoPsnsgw5WEuts01mq-pQy7UJiN5mgRxD-WUc
        X16dUEMGIv50aqzpqh4Qktb3rk-BuQy72IFLOqV0G_zS245-kronKb78cPN25DGlcTwLtj
        PAYuNzVBAh4vGHSrQyHUdBBPM"
    }
}
```


A.2. Presentación Verificable W3C


```
{
"@context": [
  "https://www.w3.org/2018/credentials/v1",
  "https://www.w3.org/2018/credentials/examples/v1"
],
"type": "VerifiablePresentation",
// the verifiable credential issued in the previous example
"verifiableCredential": [{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.w3.org/2018/credentials/examples/v1"
  ],
  "id": "http://example.edu/credentials/1872",
  "type": ["VerifiableCredential", "AlumniCredential"],
  "issuer": "https://example.edu/issuers/565049",
  "issuanceDate": "2010-01-01T19:23:24Z",
  "credentialSubject": {
    "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
    "alumniOf": {
      "id": "did:example:c276e12ec21ebfeb1f712ebc6f1",
```

```
      "name": [{
        "value": "Example University",
        "lang": "en"
      }]
    }
  },
  "proof": {
    "type": "RsaSignature2018",
    "created": "2017-06-18T21:19:10Z",
    "proofPurpose": "assertionMethod",
    "verificationMethod": "https://example.edu/issuers/565049#key-1",
    "jws": "eyJhbGciOiJSUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..TCYt5X
      sITJX1CxPCT8yAV-TVkIEq_PbChOMqsLfRoPsnsgw5WEuts01mq-pQy7UJiN5mgRxD-WUc
      X16dUEMGlv50aqzpqh4Qktb3rk-BuQy72IFLOqV0G_zS245-kronKb78cPN25DGlcTwLtj
      PAYuNzVBAh4vGHSrQyHUdBBPM"
  }
}],
// digital signature on the presentation
// protects against replay attacks
"proof": {
  "type": "RsaSignature2018",
  "created": "2018-09-14T21:19:10Z",
  "proofPurpose": "authentication",
  "verificationMethod": "did:example:ebfeb1f712ebc6f1c276e12ec21#keys-1",
  // 'challenge' and 'domain' protect against replay attacks
  "challenge": "1f44d55f-f161-4938-a659-f8026467f126",
  "domain": "4jt78h47fh47",
  "jws": "eyJhbGciOiJSUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..kTCYt5
```

```
        XsITJX1CxPCT8yAV-TVIw5WEuts01mq-pQy7UJiN5mgREEMGlv50aqzpqh4Qq_PbChOMqs
        LfRoPsnsgxD-WUcX16dUOqV0G_zS245-kronKb78cPktb3rk-BuQy72IFLN25DYuNzVBAh
        4vGHSrQyHUGlcTwLtjPAnKb78"
    }
}
```

Anexo B

Ejemplo de *VerifiableID for Natural Person* firmado con JAdES


```
{
"alg": "RS256",
"cty": "json",
"kid": "MGcwYKReMFwxCzAJBgNVBAYTAlNJMRQwEgYDVQQKEwtIYWxjb20gZC5kLjEXMBUGA1UEYRM
OVkFUU0ktNDMzNTMxMjYxHjAcBgNVBAMTFUhhbGNvbSBDQSBQTyBlLXNlYWwgMQIDEPSP",
"x5t#S256": "-zpb6qm4B4NrqhEhjoloohMtoj9jRm7BJXG3jkWB4EQ",
"x5c": [
  "MIIGYTCCBUmgAwIBAgIDEPSPMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNVBAYTAlNJMRQwEgYDVQQ
  KEwtIYWxjb20gZC5kLjEXMBUGA1UEYRMOVkFUU0ktNDMzNTMxMjYxHjAcBgNVBAMTFUhhbGNvbSB
  DQSBQTyBlLXNlYWwgMTAeFw0xOTEyMjMxMDI4MzNaFw0yMjEyMjMxMDI4MzNaMIH+MQswCQYDVQQ
  GEwJTSTEmMCQGA1UEChMdQU5USE9OWSBGSVNIRVlgQ0FNSUxMRVJJIFMuUC4xFzAVBgkrBgEEAa4
  zAgMTCDYxMDM4NzUwMRcwFQYDVQRhEw5WQVRTSS02MTAzODc1MDEtMCsGA1UEAxMkQW50aG9ueSB
  GaXNoZXIgQ2FtaWxsZXJpIFMucC4gRSBTZWFsMQ8wDQYDVQQEEwZFIFNlYWwxJjAkBgNVBCoTHUF
  udGhvbnkgRmlzaGVyIENhbWlsbGVyaSBTLnAuMS0wKwYJKoZIhvcNAQkBFh5hbnRob255QGtub3d
  sZWRnZWlubm92YXRpb24uZXUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCq6v/Ddqu
```

```
      FWMOCDqs3PXRyooEPJ7YXZHvhauSL8DncZzCpziPmEDQ9h2NR6sjOM43Om5B9nFWXHdXcUl8a5ww
      AuHH9TkKGJlhgSMDDG8cM6+l2Ns6BrnveVJ2L7Wcbi1sTDfoaqZHxe472X3YwhnP7YEZXzt9KGvF
      O3PyhFEb8y/a3vhL4X30OTicQJmO7GLlLHWVBy28o1z9he3rHFfINOvH9wHnCzAqbTLcNiOYnc0J
      p0RtlFtZj8FwpdY6RGCl8qAVLaIuG/ASpd6tTd8zs8fyazBOMHMKQ2IM92G+TdnesyER6eMLB7Oj
      7VKW9I+JEiZaPaCWsBKRAqgnvvF/DAgMBAAGjggKHMIICgzATBgNVHSMEDDAKgAhJSHZQdwqxDDC
      BggYIKwYBBQUHAQMEdjB0MBUGCCsGAQUFBwsCMAkGBwQAi+xJAQIwCAYGBACORgEBMAgGBgQAjkY
      BBDAyBgYEAI5GAQUwKDAmFiBodHRwczovL3d3dy5oYWxjb20uc2kvcmVwb3NpdG9yeRMCRU4wEwY
      GBACORgEGMAkGBwQAjkYBBglwgYAGCCsGAQUFBwEBBHQwcjBNBggrBgEFBQcwAoZBaHR0cDovL3d
      3dy5oYWxjb20uc2kvdXBsb2Fkcy9yZXBvc2l0b3J5L0hhbGNvbV9DQV9QT19ILXNlYWxfMS5jZXI
      wIQYIKwYBBQUHMAGGFWh0dHA6Ly9vY3NwLmhhbGNvbS5zaTBmBgNVHSAEXzBdMFAGCisGAQQBrjM
      FAwEwQjBABggrBgEFBQcCARY0aHR0cDovL3d3dy5oYWxjb20uc2kvdXBsb2Fkcy9maWxlcy9DUFN
      faGFsY29tX2NhLnBkZjAJBgcEAIvsQAEDMIGzBgNVHR8EgaswgagwgaWggaKggZ+GZWxkYXA6Ly9
      sZGFwLmhhbGNvbS5zaS9jbj1IYWxjb20lMjBDQSUyMFBPJTIwZS1zZWFsJTIwMSxvPUhhbGNvbSx
      jPVNJP2NlcnRpZmljYXRlcmV2b2NhdGlvbmxpc3Q7YmluYXJ5hjZodHRwOi8vZG9taW5hLmhhbGN
      vbS5zaS9jcmxzL2hhbGNvbV9jYV9wb19lLXNlYWxfMS5jcmwwEQYDVR0OBAoECEsy60sNP7RzMA4
      GA1UdDwEB/wQEAwIFoDAYBgYqhXAiAgEEDhMMODg4ODAzMDAwNjc2MAkGA1UdEwQCMAAwDQYJKoZ
      IhvcNAQELBQADggEBAFE+e6vcubeQ4I6Eptx1lE2dBxB+DEaw4m6quPbSZk7yanByp0QRG/rSXFA
      JC2PDQRVc9k/J096VftrE9tIPyOpEYXXugdLJ5t9ufpkTbGNOp1O/ioxqWcMqvY/vyuXrvsu5wAd
      0sAmKaruOqNKLSIxoy1xRxZjhfFUYIjATK8T6SCVRfojZw81Cbx0TNZHRG79dlEEg5zViy8ZPt41
      9O4iCRuzVCUfIlZ8lVtAWiEDALWR4VUXXAJN5GFLgj6Br26kLxiiABTLZcYgr8fEPUCU5mNvHWU+
      gD9yHYv68ploPbEPONK6OlcTfhvEjPitVOOB+/QBiSIr95U3+vkGRSf0="
    ],
    "typ": "jose",
    "sigT": "2022-04-13T07:18:32Z",
    "crit": [
      "sigT"
    ]
  }
```

.


```
{
    "iss": "did:ebsi:z219z1CJKSbtFc69M2jHcFmq",
    "sub": "did:ebsi:zsSgDXeYPhZ3AuKhTFneDf1",
    "jti": "urn:ebsi:status:identity:verifiableID#1dee355d-0432-4910-ac9c-
    70d89e8d674e",
    "iat": 1638316800,
    "nbf": 1638316800,
    "exp": 1953849600,
    "vc": {
        "@context": [
            "https://www.w3.org/2018/credentials/v1"
        ],
        "id": "urn:ebsi:status:identity:verifiableID#1dee355d-0432-4910-ac9c-
        70d89e8d674e",
        "type": [
            "VerifiableCredential",
            "VerifiableAttestation",
            "VerifiableId"
        ],
        "issuer": "did:ebsi:z219z1CJKSbtFc69M2jHcFmq",
        "issued": "2021-12-01T12:00:00.0Z",
        "issuanceDate": "2021-12-01T12:00:00.0Z",
        "validFrom": "2021-12-01T12:00:00.0Z",
        "validUntil": "2031-12-01T12:00:00.0Z",
        "expirationDate": "2031-12-01T12:00:00.0Z",
        "credentialSubject": {
            "id": "did:ebsi:zsSgDXeYPhZ3AuKhTFneDf1",
```

```
        "familyName": "Doe",
        "firstName": "John",
        "dateOfBirth": "1999-03-22",
        "personalIdentifier": "ES/AT/123456789"
    },
    "credentialSchema": {
        "id": "https://api.preprod.ebsi.eu/trusted-schemas-registry/v1/schemas/
        0x14b05b9213dbe7d343ec1fe1d3c8c739a3f3dc5a59bae55eb38fa0c295124f49#",
        "type": "FullJsonSchemaValidator2021"
    },
    "credentialStatus": {
        "id": "urn:ebsi:status:identity:verifiableID#1dee355d-0432-4910-ac9c-
        70d89e8d674e",
        "type": "CredentialStatusList2020"
    },
    "evidence": [{
        "type": [
            "DocumentVerification"
        ],
        "verifier": "did:ebsi:z219z1CJKSbtFc69M2jHcFmq",
        "evidenceDocument": [
            "Passport"
        ],
        "subjectPresence": "Physical",
        "documentPresence": [
            "Physical"
        ]
    }]
```

```
    }
}.<signature>
```

Anexo C

Monorepositorio Titulaciones Digitales UVa

El repositorio del monorepositorio de la PoC de titulaciones digitales de la UVa se puede encontrar en el siguiente enlace: https://github.com/alejandro-nieto-git/titulacionesDigitalesUVa