Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables

Carrera de Ingeniería en Sistemas

Implementación de la tecnología Blockchain para la validación de autenticidad de los certificados académicos digitales

Blockchain technology implementation for the authenticity validation of digital academic certificates

Trabajo de Titulación previa a la obtención del título de Ingeniero en Sistemas

AUTOR:

Edgar Patricio Sánchez Malla

DIRECTOR:

Ing. Cristian Ramiro Narváez Guillen, Mg.Sc.

Loja – Ecuador

2022

Certificación del Trabajo de Titulación

Ing. Cristian Ramiro Narváez Guillen, Mg. Sc.
DIRECTOR DEL TRABAJO DE TITULACIÓN

Certifico:

Que el Sr. Edgar Patricio Sánchez Malla, realizó el trabajo de titulación denominado **"Implementación de la tecnología Blockchain para la validación de autenticidad de los certificados académicos digitales / Blockchain technology implementation for the authenticity validation of digital academic certificates"**, ha cumplido el 100% del trabajo el cual ha sido dirigido, orientado y discutido bajo mi asesoramiento y reúne a satisfacción los requisitos exigidos en una investigación de este nivel por lo cual autorizo su presentación y sustentación.

Loja, 16 de diciembre del 2021

CRISTIAN RAMIRO NARVAEZ GUILLEN
Mi firma en este documento representa mi aceptación de los términos definidos

……

Ing. Cristian Ramiro Narváez Guillen, Mg.Sc.
DIRECTOR DEL TRABAJO DE TITULACIÓN

Autoría

Yo, **Edgar Patricio Sánchez Malla,** declaró ser el autor del presente Trabajo de Titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi Trabajo de Titulación en el Repositorio Institucional - Biblioteca Virtual.

Firma: 

Cédula de Identidad: 1104717572

Fecha: 04 de mayo de 2022

Correo Electrónico: edgar.sanchez@unl.edu.ec

Celular: 0999595341

Carta de autorización del Trabajo de Titulación por parte del autor, para la consulta de producción parcial o total, y publicación electrónica del texto completo

Yo **Edgar Patricio Sánchez Malla**, declaró ser el autor del Trabajo de Titulación titulado **"Implementación de la tecnología Blockchain para la validación de autenticidad de los certificados académicos digitales / Blockchain technology implementation for the authenticity validation of digital academic certificates"**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que, con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior, con los cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del trabajo de titulación que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los cuatro días del mes de mayo de dos mil veintidós.

Firma: Firmado electrónicamente por: EDGAR PATRICIO SANCHEZ MALLA

Autor: Edgar Patricio Sánchez Malla

Cédula: 1104717572

Dirección: Loja (c. Diamantina y Brasilia)

Correo Electrónico: edgar.sanchez@unl.edu.ec

Celular: 0999595341

DATOS COMPLEMENTARIOS:

Director del Trabajo de Titulación: Ing. Cristian Ramiro Narváez Guillen, Mg.Sc.

Tribunal de Grado: Ing. Pablo Fernando Ordoñez Ordoñez, Mg.Sc.

Ing. Francisco Javier Álvarez Pineda, Mg.Sc.

Ing. Wilman Patricio Chamba Zaragocín, Mg.Sc.

Dedicatoria

Dedico este trabajo de manera especial a mis padres Edgar y Nancy quienes son un pilar fundamental en mi día a día, que gracias a sus enseñanzas me han permitido cumplir mi meta. A mis hermanos Dayanna y José quienes siempre me han brindado su apoyo incondicional. A mi abuela Carmen que, aunque no está presente con nosotros es la que me guio y cuido hasta el final, siendo siempre esa inspiración a ser mejor ser humano. A cada uno de mis familiares quienes han aportado para poder culminar esta etapa académica. Finalmente, a mis compañeros Jhon y Luis, quienes han sido un aporte esencial a lo largo de la carrera, que a más de un compañero se ha forjado una amistad dentro y fueras de las aulas.

Edgar Patricio Sánchez Malla

Agradecimiento

Agradecer a Dios por darme la fortaleza necesaria para no rendirme en cada una de las etapas y propósitos planteados a lo largo de mi vida, sobre todo permitir que nuestros familiares y amigos estar en cada paso apoyándome y contribuyendo a mi desarrollo personal.

A mis padres y hermanos que han sido siempre el motor que impulsan mis sueños y esperanzas, quienes siempre han estado a mi lado en los días y noches difíciles durante mis horas de estudio. Gracias por ser quienes son y creer en mí.

Así mismo, agradecer a la Universidad Nacional de Loja y de manera especial a la carrera de Ingeniería en Sistemas por la predisposición ofrecida en todos estos años de formación académica. Con cada uno de los docentes que me impartieron y quienes con su vocación de enseñanza supieron guiar el aprendizaje adquirido.

Al Ing. Cristian Ramiro Narváez Guillen, mi director del Trabajo de Titulación cuya capacidad profesional y calidad humana siempre fueron una fuente de inspiración y admiración que motivaron para la culminación del Trabajo de Titulación, con sus conocimientos y apoyo se pudo realizar de la manera más adecuada.

Y finalmente a mis compañeros por toda esa ayuda colaborativa brindada en todos estos años que se compartió aula de clases. Por estar con su apoyo y constancia en las horas más difíciles de estudio.

Edgar Patricio Sánchez Malla

Índice de Contenidos

Índice de Tablas

Índice de Figuras

Índice de Anexos

1. Titulo

Implementación de la tecnología Blockchain para la validación de autenticidad de los certificados académicos digitales

Blockchain technology implementation for the authenticity validation of digital academic certificates

2. Resumen

El presente Trabajo de Titulación (TT) tiene por objetivo implementar un módulo de software para la validación de autenticidad de certificados académicos digitales usando tecnología blockchain. En el cual se tiene como base la metodología agile block chain DApp engineering (ABCDE) que en sus diferentes fases permitió culminar el TT.

En la primera etapa las fases de la metodología ABCDE que estuvieron inmersas son: definir el objetivo; los actores que intervienen; historias de usuario y requerimientos; el diagrama de la arquitectura y la división en dos subsistemas (contratos inteligentes y de aplicaciones) del módulo de software. En la segunda etapa para cada subsistema se ejecutaron las fases de diseño, codificación y pruebas unitarias. En la tercera etapa las fases que se llevaron a efecto son las pruebas de integración, funcionales y de aceptación; en conjunto con el despliegue del módulo de software en la testnet de Rinkeby de Ethereum.

Las tecnologías empleadas para el desarrollo son Truffle Suite con el box de React, Solidity, MetaMask, Ethereum; que eventualmente fue implantado en el servicio de computación en la nube de Microsoft Azure. Aunque esto último no está contemplado dentro del trabajo de Titulación, se deja en evidencia la culminación del mismo.

Por último, el módulo de software desarrollado permite realizar la validación de autenticidad de los certificados académicos digitales de forma eficiente y correcta. Al mismo tiempo permite que los mismos no puedan ser falsificados, frente a la forma tradicional en el que se validan con los múltiples problemas de confiabilidad y seguridad de la información de un sistema centralizado.

Palabras claves: [ABCDE, blockchain, contratos inteligentes, DApp, desarrollo de software, ethereum, react, truffle]

2.1. Abstract

The objective of this Degree Project (DP) aims to implement a software module for the authenticity validation of digital academic certificates applying blockchain technology in Universidad Nacional de Loja's Events system. In which it is based on the agile blockchain DApp engineering (ABCDE) methodology throughout its different phases, allowed to complete the DP.

In the first stage, the phases of the ABCDE methodology which were involved are: defining the objective; the actors involved; user history and requirements; architecture diagram and the division into two subsystems (smart contracts and applications) of software module. In the second stage, the design, codification and unit testing phases were executed for each subsystem. In the third stage the phases carried out were integration, functional and acceptance tests; jointly with the software module deployment in Ethereum Rinkeby test net.

The technologies used for the development are Truffle Suite with the React box, Solidity, MetaMask, Ethereum; which were eventually implemented in the Microsoft Azure cloud computing service. Even though the latter is not contemplated within the Degree project, its culmination is left in evidence.

Finally, the developed software module allows to carry out the authenticity validation of digital academic certificates efficiently and correctly. Simultaneously, it allows them not to be counterfeited, in contrast to the traditional way in which they are validated with multiple reliability and security problems of a centralized system information.

Keywords: [ABCDE, blockchain, smart contracts, DApp, software development, ethereum, react, truffle].

3. Introducción

La pandemia por coronavirus (COVID-19) ha provocado una crisis sin precedentes y uno de ellos es en el campo de la educación. La emergencia ha dado lugar a un cierre masivo de las actividades presenciales de las instituciones educativas con el fin de evitar la propagación de virus y mitigar su impacto. En donde el uso de las Tecnologías de la Información y de la Comunicación (TICs) también implica escoger y adaptar el material, orientando a los estudiantes en la selección de contenidos para un aprendizaje socialmente pertinente [1].

Eventualmente el modelo actual de enseñanza es más descentralizado, heterogéneo, y difícil de verificar y validar. Así mismo, frente a la pandemia del COVID-19 está transformación se ve más acelerada, dando paso a la teleformación también denominado formación en red, aprendizaje virtual, formación virtual o aprendizaje online [2]. En donde cada vez es más habitual que los estudiantes no solo reciban formación de universidades, sino que también se formen a través de la participación en cursos masivos en internet (MOOC), en talleres presenciales o a distancia, en videotutoriales, en charlas o video entrevistas, etc [3].

En el cual, la forma de reflejar el resultado del aprendizaje es por medio de un certificado académico digital que se define como una copia electrónica auténtica del título profesional, diploma o grado académico que el graduado puede tener a su disposición y que le otorgará las mismas garantías académicas y legales que después podrá compartir con cualquier empleador, universidad o entidad gubernamental [4].

La situación actual en la que se validan los certificados académicos digitales da lugar a que se genere el fraude que se presenta tanto mediante falsificaciones, como por medio de la complicidad de autoridades y personal de los centros de estudio. Debido a que actualmente varias instituciones verifican la autenticidad de sus certificados académicos mediante una consulta online. Otras delegan la tarea a terceros (notarías). Por último, existen ocasiones en las que no existe otra alternativa más que el contacto directo a la casa de estudios y su secretaría académica, para que se confirme o no la validez de un certificado académico [5].

Pero resulta que el sitio web puede ser suplantado fácilmente debido a que son sistemas centralizados. La validación por medio de terceros como una notaría u otro ente genera un costo económico adicional y la demanda de tiempo es muy alta. Además, que los certificados académicos digitales por medio de la tecnología actual pueden ser manipulados a un punto que es inapreciable a la vista. Por lo tanto, la confianza en que los certificados académicos digitales son verídicos es muy decante, debido a la deficiencia de un mecanismo que permita validar con transparencia, confiabilidad e inmutabilidad de que es un certificado académico digital es verídico.

Por consiguiente, en el Trabajo de Titulación se implementa la tecnología Blockchain que permite validar certificados académico digitales, dado que genera un registro único, consensuado y descentralizado en varios nodos de una red, donde cada uno almacena una copia exacta de la cadena que garantiza la disponibilidad de la información en todo momento [6]. Consecuentemente, se eliminan los intermediarios, descentralizando la gestión de validar estos certificados, que resulta en una operatividad más eficiente. Lo que resulta muy útil para la gestión de bienes y documentos digitales.

En la Unión Europea en materia de blockchain se tiene las regulaciones para la provisión de servicios de identificación, también permite brindar servicios de certificados digitales y de sellado de tiempo bajo la figura de los prestadores de servicios de confianza. En los Estados Unidos de América se adoptó la postura de reconocer la validez legal expresa de los actos celebrados usando tecnología blockchain en ciertas jurisdicciones mediante el dictado de una normativa especial. En la actualidad Ecuador carece de normativa especial para la tecnología blockchain, siendo de aplicación las secciones pertinentes de la Ley Ecuatoriana de E-Commerce [7]. En consecuencia, es imprescindible un mecanismo que permita regular la validación de autenticidad de un certificado académico digital con confiabilidad, trazabilidad e inmutabilidad, de forma fácil y accesible para usuarios, instituciones y empresas.

El Trabajo de Titulación tiene por objetivo general el de “Implementar un módulo de software para la validación de certificados académicos digitales por tecnología blockchain”, y para cumplirlo se establecieron tres objetivos específicos: 1. Definir el módulo de software para la validación de certificados académicos digitales usando la Ingeniería de Requisitos, 2. Desarrollar el módulo de software para la validación de certificados académicos digitales por medio de la plataforma Blockchain Ethereum Virtual Machine y 3. Probar la solución en una testnet de blockchain para el módulo de software de validación de certificados académicos digitales.

4. Marco teórico

4.1. Blockchain

La cadena de bloques, más conocida por el término en inglés blockchain, es una tecnología que permite la transferencia de datos digitales con una codificación muy sofisticada y de una manera completamente segura, es decir, se habla de un texto de acontecimientos digitales, esta transferencia o procedimiento no requiere de un intermediario centralizado (como los bancos en el caso del dinero y notarias para el caso de los contratos o acuerdos) que identifique y certifique la información allí contenida, sino que esta información está distribuida en múltiples nodos independientes entre sí, que la registran y la validan. Una vez la información este en la cadena de bloques, “la información no puede ser borrada, solo se podrán añadir nuevos registros, y no será legitimada a menos que la mayoría de ellos se pongan de acuerdo para hacerlo” [8].

También se conoce a esta tecnología como una base red de procesamiento distribuida donde los datos y las tareas de procesamiento de las transacciones son repartidos a los participantes la red, sin embargo esta tecnología no consiste en una única técnica, sino que se vale de la criptografía, las matemáticas, distintos modelos económicos, combinados con redes “peer-to-peer” (punto a punto) y algoritmos de consenso distribuido para resolver el problema tradicional de sincronización de bases de datos distribuidas. Es una construcción de infraestructura que integra muchos campos [9].

En blockchain cada contrato, proceso, tarea y pago podría tener un registro digital y una firma que podría ser identificable, validada, almacenada y compartida. Cualquier aplicación que presente altos costos de intermediación, problemas de agilidad y/o vulnerabilidad por la necesidad de involucrar un intermediario que genere confianza entre proveedores y clientes se puede beneficiar del uso de la blockchain [10].

En cada bloque se almacena [6]:

- Una cantidad de registros o transacciones válidas.
- Información referente a ese bloque.
- Su vinculación con el bloque anterior y el bloque siguiente a través del hash de cada bloque un código único que sería como la huella digital del bloque.

Por lo tanto, cada bloque tiene un lugar específico e inamovible dentro de la cadena, ya que cada bloque contiene información del hash del bloque anterior. La cadena completa se guarda en cada nodo de la red que conforma la blockchain, por lo que se almacena una copia exacta de la cadena en todos los participantes de la red, así como se observa en la Figura 1 tomada de [6].

A medida que se crean nuevos registros, estos son primeramente verificados y validados por los nodos de la red y luego añadidos a un nuevo bloque que se enlaza a la cadena.

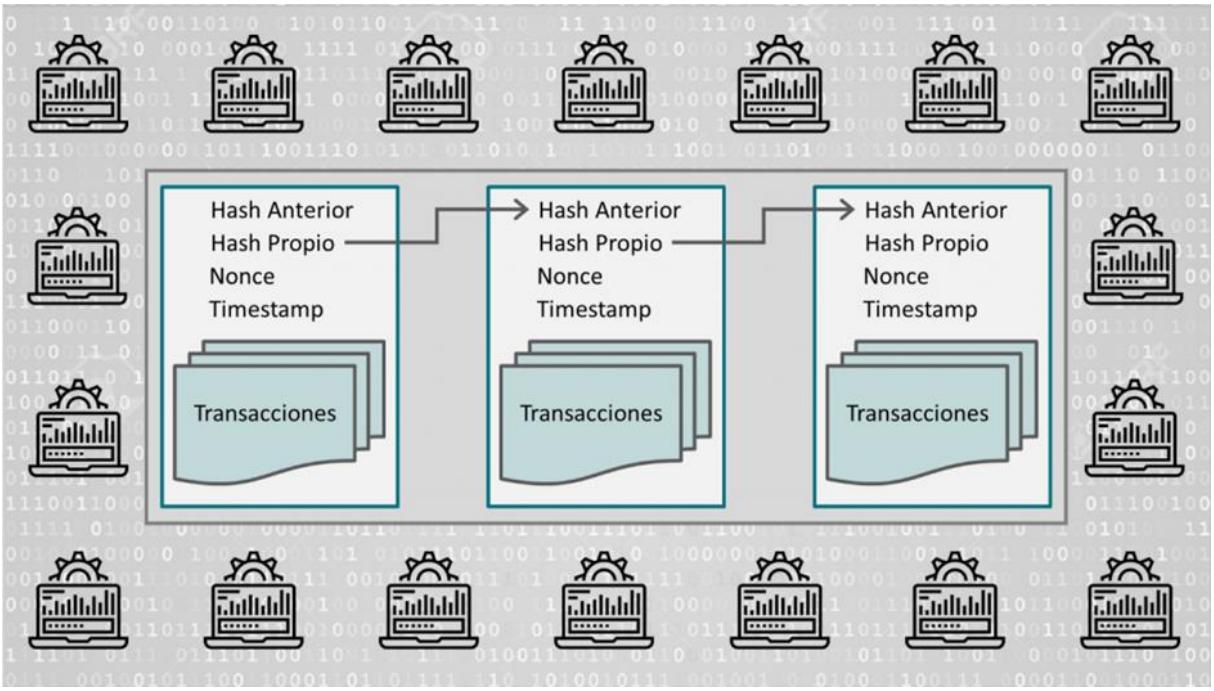


Figura 1. Esquema de la cadena de bloques [6].

4.1.1. Características

Las principales características de la tecnología Blockchain que tenemos según [9]:

- **Descentralización.** Aquí no se confía en un único nodo centralizado (sistema centralizado). Por el contrario, cada transacción es controlada, compartida y autorizada por todos los nodos que participan de la red blockchain (sistema descentralizado).
- **Transparencia.** Los datos almacenados por los sistemas que usan la tecnología Blockchain son transparentes para cada nodo. En una analogía con los libros contables, en la cual, cada transacción queda registrada y cualquier nodo puede consultarla.
- **Código abierto.** Por lo general los sistemas Blockchain son abiertos para cualquier individuo que desee participar en la red. Esto significa que cualquiera puede acceder a la información de la red. Además, los registros pueden ser verificados públicamente y usar esta tecnología para crear cualquier aplicación que se requiera.
- **Autonomía.** Debido a la base del consenso, cada nodo del sistema de Blockchain, tienen la capacidad de transferir o actualizar datos de manera segura y confiable. El objetivo es poder confiar tanto en una sola persona, como en todo el sistema.
- **Inmutabilidad.** En este caso cualquier registro se preservará para siempre, debido a que las operaciones que se realizan no se pueden alterar y son únicas. Las

transacciones efectuadas son realizadas con base a un sistema criptográfico, lo que resulta que las operaciones sean casi imposibles de hackear.

- **Anonimidad.** La tecnología Blockchain solucionó el problema de confianza entre nodos, es decir, la transferencia de datos. En donde, cada transacción puede ser anónima, solo es necesario conocer la dirección de Blockchain de la otra persona.
- **Trazabilidad.** Blockchain garantiza la capacidad de registrar la traza de los productos a lo largo de la cadena de suministro interna o externa, empaquetarlos en un formato legible y prepararlos para poder ser gestionados por el propio software o como respuesta a una solicitud de servicio.

4.1.2. Usos

En esta tecnología se puede usar básicamente cualquier tipo de información que requiera ser preservada de forma intacta y que deba permanecer disponible. Donde es almacenada de manera segura, descentralizada y más económica que a través de intermediarios. A continuación, se describe algunos usos según [6]:

- **Uso de blockchain en la salud.** Por ejemplo, en los registros de salud podrían ser unificados y almacenados en blockchain. De tal manera, la historia médica de cada paciente estaría segura y a la vez disponible para cada médico autorizado, independientemente del centro de salud donde se haya atendido el paciente. Además, en la industria farmacéutica se puede utilizar para verificar medicamentos y evitar falsificaciones.
- **Uso de blockchain para documentos.** En este caso, resultaría muy útil para la gestión de bienes y documentos digitales. Blockchain permite registrar compras, escrituras, documentos o cualquier tipo de bien digital y que no pueda ser falsificado.
- **Otros usos del blockchain.** También se puede revolucionar el mercado de Internet de las Cosas (IoT), donde existe millones de dispositivos conectados a Internet que deben ser gestionados por las empresas proveedoras. En un futuro próximo, el modelo centralizado no va a soportar tantos dispositivos, sin contar que muchos de ellos no son lo suficientemente seguros. Con la tecnología blockchain los dispositivos pueden comunicarse a través de la red de manera directa, segura y confiable, sin intermediarios.

4.1.3. Definiciones complementarias

Según [10] para entender la forma en la que opera blockchain es necesario conocer algunas definiciones complementarias:

- **Hoja única contable abierta y distribuida (Open Ledger):** La tecnología Blockchain está constituida por una base de datos distribuida, que consta de un registro

compartido que contiene la información de los dueños de los activos y el histórico del intercambio de la propiedad de los mismos (transacciones). Esto quiere decir que se puede establecer la cadena de propiedad del activo desde su origen o emisión.

- **Nodos:** Cada participante de la red Blockchain constituye un nodo. Cada vez que se generan transacciones estas se transmiten a los nodos, un bloque se confirma y se añade a la cadena, esta operación actualiza la hoja contable que cada participante almacena.
- **Mecanismos o algoritmos de consenso:** La propiedad de los activos y la transferencia de los mismos son validadas por los participantes de la red. La mayoría de nodos debe de estar de acuerdo sobre el estado resultante y el orden de cada transacción, esto se logra verificando en la hoja contable distribuida i) si el activo existe (origen y emisión), ii) si este pertenece a quien transfiere (cadena de propiedad), y iii) si previamente no ha sido transferido a otro participante (evitar el fraude por doble desembolso). El mecanismo de consenso también se encarga de verificar que los nodos sean “honestos”, es decir, sean reales y no varios participantes falsos dominados por un atacante para ganar más participación en la decisión. Una vez verificada, una transacción no podrá ser eliminada o modificada.
- **Bloque:** Para brindar mayor seguridad las transacciones se almacenan en bloques y este se encadena a bloques anteriores. Cada bloque contiene una estampilla de tiempo que indica el momento en el cual se hicieron las operaciones contenidas en él y un número cifrado (hash) que se crea a partir de la combinación de las transacciones contenidas y la referencia al bloque anterior. Las transacciones quedan en firme cuando su bloque contenedor se enlaza a la cadena de bloques. Cualquier modificación en las transacciones modifica el hash del bloque, esto rompe la cadena y las transacciones se anulan.
- **Mineros:** Son los nodos que se encargan de las operaciones de validación de transacciones y de la creación y enlace de nuevos bloques dentro de la cadena. Los mineros prestan sus recursos para mantener la seguridad y confianza de las transacciones. Los mineros que validan bloques son recompensados por la blockchain con tokens, estos llegan a tener valor de mercado.

4.2. Operación básica de Blockchain

A continuación, se describe la operación básica de la Blockchain según [10]:

1. Acuerdo entre participantes antes de la transacción: Se determinan los componentes para el acuerdo, como las variables de transacción, el tipo de activo, tamaño o cantidad, dirección del cliente, entre otras.
2. Publicación de la operación: La transacción se transmite a los nodos de la blockchain.

3. Creación del bloque: Los nodos combinan la operación con otras transacciones en un bloque, el bloque genera un número Hash, el cual se crea a partir de:
    a. Un número hash resultante de la combinación de todas las transacciones, para ello usa una estrategia conocida como árbol de Merkle. Un cambio en una de las transacciones produce un Hash totalmente diferente.
    b. La estampilla de tiempo de la creación del bloque, esto permite validar que el bloque nuevo es creado posteriormente al bloque anterior.
    c. La combinación con el Hash del bloque anterior, esto permite determinar si el nuevo bloque hace referencia a un bloque anterior existente y válido. Si el bloque anterior no existe se anulan las operaciones.
    d. Un número aleatorio conocido como Nounce.
4. Validar el bloque: Esta es la operación de minería, para ello se usan diferentes aproximaciones como la prueba de trabajo (Proof of work) o la prueba de participación (Proof of Stake). El resultado de esto es un número que combinado con el Hash del nuevo bloque permite enlazarlo a la Blockchain.
5. Distribución de la solución y verificación de transacciones: La solución se transmite a la red, los nodos verifican cada transacción del bloque verificando la existencia, pertenencia y orden de las transacciones.
6. Encadenamiento del bloque: Una vez las transacciones del bloque son verificadas por la mayoría de nodos, el bloque nuevo se combina con los bloques anteriores, creando una cadena de bloques.
7. Confirmación de la transacción: De acuerdo al tipo de Blockchain para confirmar una transacción se debe esperar que se enlacen en promedio otros 5 bloques.

4.3. Algoritmos de Consenso de Blockchain

El algoritmo de consenso es fundamental en la blockchain y se encarga del encadenamiento de los bloques. El objetivo que tiene es proveer de un conjunto de normas bien definidas que deben cumplir todos los usuarios para mantener la consistencia de la cadena. Este debe actuar frente a las bifurcaciones de la cadena (forks), los participantes deben ponerse de acuerdo sobre que cadena es la válida. Además, debe decidir que participante puede o no añadir un nuevo bloque a la cadena y se encarga del encadenamiento de los bloques.

Los algoritmos de consenso más populares son PoW (proof of work) y PoS (proof of stake)

4.3.1. Proof of Work (PoW)

En Proof of Work (prueba de trabajo) los nodos tienen que resolver un problema matemático para poder añadir un bloque a la cadena. La cadena valida es aquella que más bloques contenga. Si alguien intentase modificarla necesitaría un poder computacional más elevado

que el resto de nodos de la red juntos, algo poco probable y requeriría de mucha energía eléctrica [11].

El principal inconveniente es su enorme gasto de recursos. Además, requiere que se actualice el hardware frecuentemente [11].

4.3.2. Proof of Stake (PoS)

En Proof of Stake (prueba de participación), los participantes con más participación en la red, es decir, aquellos que más tokens poseen, tienen más probabilidad de añadir un bloque a la cadena. Se asume que un usuario con mucha participación velará por la seguridad de la cadena (no se perjudicaría a el mismo) de este modo los bloques que genere no serían maliciosos [11].

4.4. Tipos de Blockchain

La tecnología de Blockchain puede ser dividida en dos tipos:

4.4.1. Blockchain pública

La Blockchain pública es aquella que no tiene restricciones en cuanto a la lectura de sus datos y a la visualización de las transacciones para su inclusión en la cadena de bloques, todos los participantes tienen derecho de enviar transacciones y en caso que sean válidas, estas serán incluidas en la Blockchain. Para el proceso de consenso todos los nodos tienen la posibilidad de participar [10].

La Blockchain pública es considerada una arquitectura “totalmente descentralizada”. Por ejemplo, Bitcoin y Ethereum son Blockchain públicas [9].

4.4.2. Blockchain privada

La Blockchain privada es aquella que tiene acceso directo a los datos de la cadena de bloque, pero la vista de las transacciones es limitada a una lista predefinida de entidades, el permiso de escritura es mantenido por solo una organización y los permisos de lectura pueden ser públicos o en cierta manera restringidos arbitrariamente. La Blockchain privada, por su naturaleza, en la mayoría de los casos no brinda ventajas significativas en comparación con el modelo tradicional centralizado de confianza [10].

4.5. Contratos inteligentes

Los contratos inteligentes o smart contracts, son programas de computación que se ejecutan en el bloque mayor, se han convertido en una característica de blockchain. Este tipo de programa puede ser usado para facilitar, verificar, o reforzar las reglas entre las partes, permitiendo el procesamiento directo e interacción con otros contratos inteligentes [12].

Los contratos inteligentes están escritos en código de programación, es decir, son programas informáticos que ejecutan autónoma y automáticamente los términos de un contrato, el programa puede definir las reglas y las consecuencias estrictas del mismo, de la misma manera que lo haría un contrato tradicional, pero a diferencia de un documento legal tradicional también puede obtener información como entrada y procesarla según las reglas establecidas en el contrato para, a continuación, adoptar las medidas que se requieran como consecuencia de ello. Todo ello sin la intervención humana en el proceso [8].

Por otro lado, un contrato inteligente puede ser creado y llamado por personas naturales y/o jurídicas, pero también por máquinas u otros programas que funcionan de manera autónoma. Un contrato inteligente tiene validez, sin depender de autoridades, debido a su naturaleza: es un código visible por todos y que no se puede cambiar al existir sobre la tecnología blockchain, la cual le da ese carácter descentralizado, inmutable y transparente [13].

Es indispensable lograr desarrollar una relación entre la certificación digital y los contratos inteligentes ya que, en la certificación digital, un contrato inteligente podría actuar como una interfaz en la blockchain que establezca las reglas para la gestión y el almacenamiento de atributos de identidad. Sin embargo, es importante apreciar detalles, como la forma que se almacenan los datos y los tipos de datos de datos utilizado [8].

4.5.1. Características

Estas son las principales características que permiten confiar en los contratos inteligentes según [14]:

- **Inmutabilidad:** Ser inmutable significa que una vez que se haya creado, nunca podrá ser modificado. Nadie puede mal intencionadamente alterar el código del contrato inteligente.
- **Distribuido:** Los contratos, como pasa con las transacciones de Bitcoin, también son validados por la red.
- **Deterministas:** Dado que el código alojado en un contrato inteligente se ejecuta simultáneamente en múltiples nodos distribuidos, debe ser determinista, es decir, dada una entrada, todos los nodos deben producir el mismo resultado. Eso implica que el código no debe contener ninguna aleatoriedad.
- **Verificables:** Una vez implementado un contrato inteligente, obtendrá una dirección única. Antes de usarse, las partes interesadas en usarlo pueden ver y comprobar el código para mayor seguridad.

4.5.2. Procedimiento de un contrato inteligente

El procedimiento se puede observar en la Figura 2 tomada de [15]. Donde una cuenta de usuario puede mandar ether a:

1. Una cuenta de usuario.
2. Una cuenta de contrato.
3. El contrato puede desencadenar acciones como consecuencia, es decir, mandar ether a otro usuario o contratos.

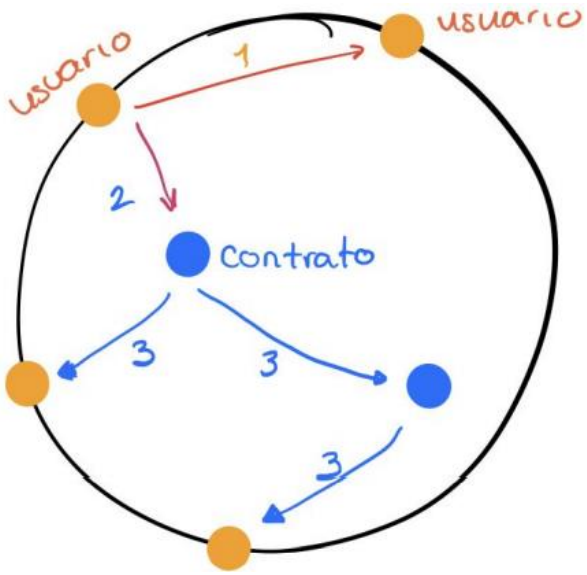


Figura 2. Procedimiento de un contrato inteligente [15].

Actualmente existe una gran variedad de plataformas blockchains que admiten contratos inteligentes, pero el más grande y adaptable al Proyecto de Trabajo de Titulación es Ethereum, que se fundamenta en base al Anteproyecto. Bitcoin también tiene soporte para contratos inteligentes, pero son muy limitados en comparación con Ethereum.

4.5.3. Composición de un contrato inteligente

El contrato tiene una cuenta asociada, la cual está compuesta por 3 campos según [16]:

- **Saldo:** La cantidad de ether que esa cuenta posee, es importante ya que con esa cuenta se harán los despliegues de contratos inteligentes que cuestan ether y todas las modificaciones a otras cuentas también cuestan ether, por lo que será esencial que la cuenta del contrato inteligente tenga saldo para poder realizar estos movimientos (transferencias).
- **Almacenamiento:** simplemente los datos almacenados para este contrato inteligente, cualquier dato que sea esencial para la interacción o funcionamiento del contrato inteligente.

- **Código:** el código esta compilado, por lo que sería imposible leerlo para una persona, pero en cambio para el ordenador con el que interactuamos es su lenguaje natural.

4.5.4. Gas y transacciones

Al igual que se almacena el código en un servidor externo, en Ethereum igualmente hay que pagar por la computación (hacer cálculos, modificar contratos o hacer transacciones). La diferencia radica en que no paga la dueña de la aplicación los gastos de computación, sino los usuarios. Se debe a que Ethereum es una red descentralizada y como tal, se pretende que no sea necesario el control administrativo sobre la aplicación [16].

Una transacción tiene un coste calculado en WEI y un coste máximo de transición, en la siguiente Tabla 1 tomada de [16] se puede visualizar las equivalencias de las unidades de WEI a ETH.

Tabla 1.
Equivalencias de las unidades WEI a ETH

Unidad	Valor de Wei	Wei
wei	1 wei	1
Kwei (babbage)	1e3 wei	1,000
Mwei (lovelace)	1e6 wei	1,000,000
Gwei (shannon)	1e9 wei	1,000,000,000
microether (szabo)	1e12 wei	1,000,000,000,000
miliether (finney)	1e15 wei	1,000,000,000,000,000
ether	1e18 wei	1,000,000,000,000,000,000

4.6. Aplicación Descentralizada (DApp)

DApp (Decentralized Application) es una aplicación que en su mayoría o completamente esta descentralizada. La estructura de una DApp se muestra en las Figura 3 y Figura 4, tomadas de [17]. Considerando los siguientes aspectos:

- Backend.
- Frontend.
- Almacenamiento.
- Comunicación.
- Resolución de nombre.

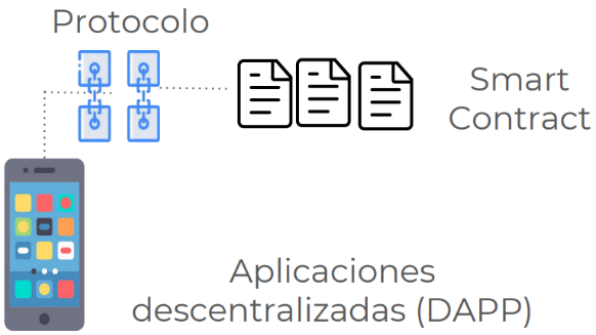


Figura 3. Estructura simplificada de una DApp [17].

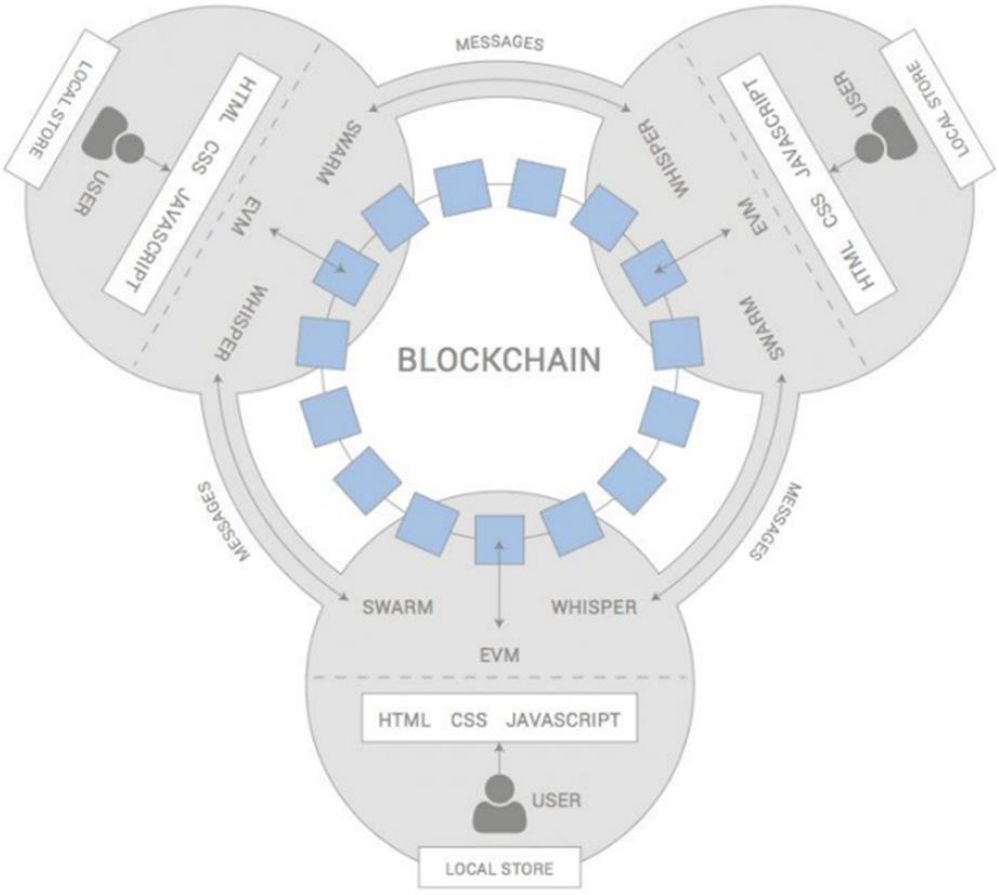


Figura 4. Estructura extendida de una DApp [17].

Las DApps según [17], deben cumplir con las siguientes tres características:

1. **Código abierto.**
    a. Debe funcionar como una aplicación independiente.
    b. Ninguna organización puede reclamar la posesión de la mayor parte de sus tokens.
    c. Un DApp puede adaptar su protocolo en respuesta a las mejoras sugeridas y los comentarios del mercado, pero todos los cambios deben ser adoptados por consenso de todos sus usuarios.

2. **Encriptación.** Los datos de DApp y los informes operativos deben encriptarse y almacenarse en un dominio público, el llamado blockchain descentralizado, para evitar cualquier posible interrupción de la red.
3. **Token.** Un DApp debe requerir un token criptográfico (bitcoin o token de la aplicación original) para acceder a él, cada aportación aportada por los mineros debe ser recompensada en los tokens de DApp.

4.7. Comparativa de una aplicación web tradicional vs una DApp (Blockchain).

Es necesario realizar una comparativa entre una aplicación web tradicional y una DApp como se observa en la Figura 5. La primera utiliza HTML, CSS y JavaScript para representar una página, además de tomar detalles de una base de datos a través de una API. En la segunda es similar, la parte frontal utiliza exactamente la misma tecnología para mostrar al usuario final. La diferencia radica en que, en lugar de una API que se conecta a una base de datos, tiene un contrato inteligente que permite interactuar con la red Blockchain.

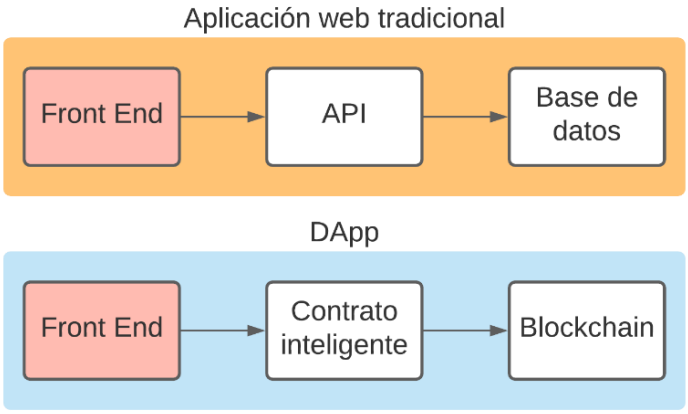


Figura 5. Comparativa de una aplicación web tradicional vs una DApp (Blockchain).

4.8. Ethereum

Ethereum es la segunda plataforma de blockchain más conocida. Fue creada por el conocido Vitalik Buterin, programador y escritor ruso, conocido también por ser el cofundador de Bitcoin Magazine, que a finales de 2014 comenzó con el desarrollo de Ethereum [14].

El propósito de Ethereum es desarrollar una plataforma que permita y facilite a otros programadores la creación de aplicaciones descentralizadas con los contratos inteligentes (smart contracts) como base de estas. Al mismo tiempo sirve como plataforma mundial donde se ejecutan estas aplicaciones [14].

Ethereum tiene su propia criptomoneda llamada Ether (ETH), el combustible que impulsa la plataforma. Esta criptomoneda es la monedan utilizada por los clientes de la blockchain de Ethereum [14]. Además de ser una moneda similar a Bitcoin para poder realizar pagos a otras personas, también es usado en el desarrollo de los contratos inteligentes (smart contracts),

es decir, ether es el incentivo que asegura que los programadores escriban aplicaciones de calidad y por último la recompensa que recibirán los mineros al ir incluyendo los bloques en la cadena.

Ethereum utiliza las características y la tecnología de la blockchain para crear su infraestructura para la evolución de las aplicaciones y servicios centralizados a un mundo descentralizado. Ethereum también utiliza como Bitcoin el mecanismo de Proof of Work para la creación de nuevos bloques en la blockchain. Pero desde el año 2017 se viene hablando de que Ethereum en una de sus próximas actualizaciones cambiará su mecanismo de minado al conocido como Proof of Stake o prueba de participación [14].

A diferencia del Proof of Work, donde el algoritmo recompensa a los mineros que resuelven problemas matemáticos con el objetivo de validar transacciones y crear nuevos bloques, con el Proof- of Stake, el creador de un nuevo bloque es elegido de manera aleatoria.

Proof of Stake se basa en un sorteo. Para participar en él, se debe bloquear tanta cantidad de ether como se desee. La cantidad de ether bloqueada se usará para escoger el minero del bloque. Cada ether tiene un identificador a sus espaldas y, por lo tanto, cuanto más ether bloqueado, más posibilidades hay que, como si papeletas en un sorteo se tratara, de ser elegido. Para escoger el ganador del sorteo, aleatoriamente se seleccionará usando alguno de los valores aleatorios de un bloque [14].

4.8.1. Ethereum Virtual Machine

Ethereum Virtual Machine (EVM) es una máquina virtual que es parte del ecosistema blockchain de Ethereum. La función que cumples es la de permitir la ejecución de programas o contratos inteligentes, con el fin de desplegar sobre dicha blockchain una serie de funcionalidades añadidas para que los usuarios puedan disfrutar de las mismas.

EVM se construyó un lenguaje especializado de alto nivel llamado Solidity. Por medio de este lenguaje de programación se facilita la creación de los contratos inteligentes. El proceso que sigue es, primero se transforma Solidity a los códigos de operación (OP_CODES) y luego a un bytecode. Esté es finalmente ejecutado por la EVM para realizar las operaciones especificadas en un contrato inteligente [18]. Por lo tanto, hace que la EVM puede ejecutar desde las más sencillas hasta las más complejas operaciones. A continuación, se muestran las características de EVM según [18]:

- La EVM está enfocada en proporcionar seguridad y ejecutar códigos no confiables en computadoras de todo el mundo.

- Las aplicaciones descentralizadas y los contratos inteligentes desarrollados en la EVM son completamente descentralizados y distribuidos. Por lo que no requiere de la participación de terceros. Ni pueden ser modificadas ni alteradas.
- La EVM permite el desarrollo de una mayor cantidad de aplicaciones, y que éstas puedan ejecutarse sobre una misma red blockchain, sin afectar otras operaciones.
- Los contratos inteligentes diseñados en la EVM son invariables y pueden ejecutarse y hacerse cumplir por sí mismo, de una manera autónoma y automática. Con lo que se elimina la burocracia, los altos costos y el tiempo de espera típicos en los contratos tradicionales.
- La EVM es sustancialmente menos eficiente que muchas otras máquinas virtuales convencionales. Esto se debe a que principalmente su diseño se basó en la utilidad del momento y no en el alto rendimiento.
- Los cambios y mejoras experimentados por la EVM han sido pocos hasta ahora. Por lo que no está optimizada en cuanto a la velocidad para distintas plataformas de hardware.
- El diseño de la EVM no está dirigido a la portabilidad, lo que limita los espacios en los que dicha máquina virtual puede implementarse.

4.9. Node.js

Node.js se define como un entorno de tiempo de ejecución de JavaScript. En el cual el tiempo de ejecución en tiempo real abarca todo lo se requiere para ejecutar un programa realizado en JavaScript [19]. Además, Node.js según [20] es una plataforma construida encima del entorno de ejecución javascript de Chrome para fácilmente construir rápidas, escalables aplicaciones de red. El mismo usa un modelo de E/S no bloqueante dirigido por eventos que lo hace ligero y eficiente, perfecto para aplicaciones data-intensive en tiempo real.

4.10. Single Page Applications (Aplicaciones de una sola página)

Una SPA es una aplicación web que ejecuta todo su contenido en una sola página. Según [21] una SPA carga un único documento HTML desde el servidor y luego muestra selectivamente diferentes elementos del DOM para crear la ilusión de una navegación de múltiples páginas. En el cual, al cargar por una sola vez, se puede mostrar múltiples vistas utilizando la misma página.

En la Figura 6 tomada de [22], se observa que una SPA después de la primera llamada al servidor, ya no interactúa directamente. Luego de eso las llamadas se hacen a partir de AJAX y se responde a partir de fragmentos de HTML y datos JSON, las mismas que devuelven solo lo necesario y no se necesita recargar la página en el navegador. Es decir, se refresca dinámicamente a diferencia de una aplicación web tradicional.

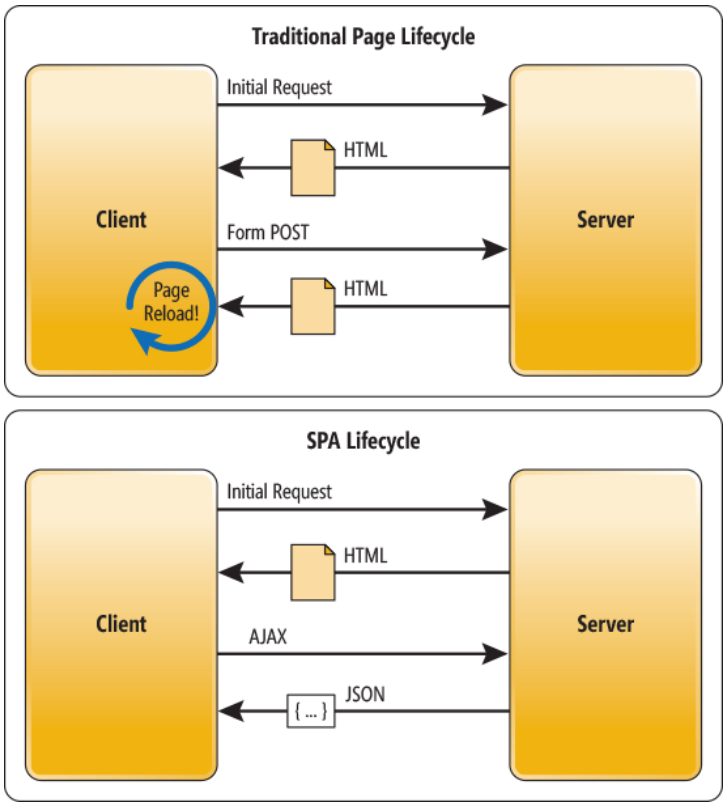


Figura 6. Arquitectura de una aplicación web tradicional vs SPA [22].

En este caso se utiliza React JS para la construcción de la solución del TT, el cual se define a continuación.

4.11. React JS

Es una biblioteca Javascript para crear interfaces de usuario [23]. Otra definición por parte de [24], React JS es una librería JavaScript desarrollada por Facebook, para el desarrollo de interfaces de usuario. Aunque no solo sirve para esto, con ella también se puede desarrollar aplicaciones web y móviles, aplicaciones de una página (SPA) y puede ser usada del lado del servidor y del cliente (front-end).

La librería React JS está conformada por diversos módulos, herramientas y componentes que utilizados correctamente permiten desarrollar completos proyectos en poco tiempo y de forma sencilla, gracias además a que tiene mucho código ya listo para ser usado. React JS también utiliza JSX que no es más que un pseudolenguaje que permite escribir HTML dentro del JavaScript, de forma elegante, fácil y muy bien estructurada. Esta librería puede ser empleada en cualquier framework JavaScript [24].

4.12. Truffle Suite

Truffle Suite proporciona el conjunto de herramientas, que permite a los desarrolladores crear, probar e implementar diversas soluciones de software sobre una Blockchain de Ethereum. Las herramientas son:

- Truffle Teams.
- Truffle.
- Ganache.
- Drizzle

Surge, debido a que el desarrollo de DApps y de contratos inteligentes suele ser algo extremadamente engorroso de desarrollar y gestionar cuando no se dispone de experiencia y las herramientas adecuadas. Truffle Suite es un excelente paquete de herramientas de código abierto especializado en el desarrollo de aplicaciones sobre la Blockchain de Ethereum. A continuación, se detallan cada una de ellas.

4.11.1. Truffle Teams

Truffle Teams es una herramienta que permite gestionar y supervisar el estado de las aplicaciones habilitadas en una Blockchain. Entre sus características más destacadas según [25] son:

- Seguimiento y datos de DApp: Proporciona una visión general de los estados de las pruebas en varios proyectos de una forma cómoda e integral.
- Configuración cero Integración continua: Elimina las inconsistencias ambientales al proporcionar un entorno unificado para pruebas continuas de contratos inteligentes.
- Despliegues automatizados: Facilita el trabajo en equipo, haciéndolo tan fácil como trabajar sobre GitHub. Además, permite ejecutar pruebas automáticamente, liberando horas/labor de los desarrolladores y recursos del sistema.

4.11.2. Truffle

Truffle es un entorno de desarrollo de clase mundial, que provee un marco de pruebas y una cartera de activos para Blockchain utilizando la Máquina Virtual Ethereum (Ethereum Virtual Machine - EVM), con el objetivo de hacer la vida más fácil a los desarrolladores. Entre sus características más destacadas según [25] están:

- Compilación, enlace, despliegue y gestión binaria de contratos inteligentes incorporados.
- Pruebas de contrato automatizadas para un desarrollo rápido.
- Marco de trabajo de implementación y migraciones extensibles y programable.
- Gestión de redes para su despliegue en cualquier número de redes públicas y privadas.
- Gestión de paquetes con EthPM y NPM, utilizando el estándar ERC190.
- Consola interactiva para la comunicación directa con el cliente.
- Construcción configurable de tuberías con soporte para una integración estrecha.

- Corredor de scripts externo que ejecuta scripts dentro de un entorno Truffle.

4.11.3. Ganache

Ganache es una Blockchain personal de Ethereum para elaborar y probar desarrollo tales como DApps y de contratos inteligentes. Viene disponible tanto como una aplicación de escritorio como una herramienta de línea de comandos (anteriormente conocida como TestRPC). Ganache está disponible para Windows, Mac y Linux [25].

4.11.4. Drizzle

Drizzle es una colección de librerías front-end que hacen que el desarrollo de frontends para Dapps sea más fácil y predecible. El núcleo de Drizzle está basado en una tienda de Redux, lo que otorga acceso a espectaculares herramientas de desarrollo de Redux. Drizzle sincroniza los datos de los contratos, de las transacciones, y de muchos otros elementos más. Por ende, mantiene la rapidez de los procesos u operaciones al mantener sincronizado lo configurado [25].

4.13. Visual Studio Code

Visual Studio Code se define como un editor de condigo fuente, considerado ligero pero potente, que se ejecuta a nivel de escritorio. El mismo se encuentra disponible para los sistemas operativos Windows, macOS y Linux. Tiene soporte incorporado para JavaScript, TypeScript y Node.js, que incorpora un gran ecosistema de extensiones para múltiples lenguajes como: C++, C#, Java, Python, PHP y Go. Además, de tiempos de ejecución: como .NET y Unity [26].

4.14. Solidity

Según [16], Solidity es un lenguaje de programación muchas veces comparado con JavaScript por algunas de sus características, aunque tiene diferencias muy importantes, como por ejemplo que al contrario que JavaScript, Solidity es altamente tipado, significando esto que deberemos declarar de que tipo es cada variable para utilizarla, debiendo saber si necesitamos un string, int o boolean antes de asignar la variable.

Al escribir el código en Solidity, al compilarlo y después de la compilación el código tendrá dos resultados, el primero será el Byte code que no es más que el código compilado a muy bajo nivel, para que el ordenador pueda leerlo y será utilizado para el despliegue en la blockchain. La otra parte que tendremos como resultado es el ABI de la aplicación, el ABI de la aplicación es el encargado de hacer la comunicación con el front-end de la web contra el Byte code que se ha desplegado en la blockchain, es decir alguna manera un traductor [16].

Las variables que se usan en el lenguaje de programación según [16] son:

- **Enteros.** Se representan como int/uint siendo signed (puede ser negativo) y unsigned (no puede ser negativo) respectivamente. El tamaño se representa con el número de bits que puede contener el entero, siendo este en saltos de 8. Ejemplo: uint8 será un entero que podrá alojar números del 0 al 2** 8 – 1, uint e int son los sinónimos de uint256 e int256 respectivamente, por lo que cuando se vea en el código uint significara que esa variable puede alojar números hasta 2** 256 - 1. El valor por defecto de esta variable será 0.
- **Booleanos.** Los booleanos se representan como bool. Por defecto son false. La funcionalidad de los mismo es exactamente igual que en otros lenguajes
- **Address.** Tiene dos subtipos que son a grandes rasgos idénticos excepto la siguiente diferencia:
    - **address:** Contiene un valor de 20 bytes (tamaño de una dirección ethereum)
    - **address payable**: Igual que “address”, pero también tiene los métodos “transfer” y “send”.

  La lógica de la distinción es que a una address payable se le pueden mandar fondos y a una address no.
- **String.** Se representa con string.
- **Mapping.** Se representa con **mapping(_KeyType = > _KeyValue)**. Un mapa es muy parecido a los diccionarios en otros lenguajes donde el primer parámetro es la clave y el segundo es el valor de esa clave.
- **Structs.** Es representado como **struct nombre_struct {tipo1 nombre_tipo1; ...}.** Un struct igual que en otros lenguajes es utilizado para crear nuestros propios tipos complejos de variables, esto es entendible como una especie de objeto, el cual está compuesto por diferentes componentes.
- **Arrays.** Se representa **tipo nombre [tamaño]** para los de tamaño fijo y **tipo [ ] nombre** para los arrays de tamaño dinámico. El funcionamiento de los arrays es igual que en otros lenguajes de programación como Java, c#, etc…

4.15. Funciones HASH

La función hash es un tipo de operación que toma una entrada de datos arbitraria y la mapea a una salida de un tamaño fijo, llamado hash o resumen. El tamaño de salida suele especificarse en bits de datos y se incluye a menudo en el nombre de la función hash. Las funciones hash criptográficas se usan para diversas aplicaciones como autenticación, integridad de mensajes, huellas digitales, detección de datos corruptos y firmas digitales. El hash criptográfico difiere claramente de la encriptación porque las funciones hash criptográficas deben ser unidireccionales. es decir, en una función hash

puedes introducir texto sin formato y obtener una salida ininteligible, pero no se puede tomar la salida del hash y recuperar el texto sin formato [27].

Para el presente TT se realiza una comparativa entre MD5 y SHA-256.

4.14.1. MD5

El algoritmo MD5 toma como entrada un mensaje de longitud arbitraria y produce como salida un compendio de mensajes de 128 bits de 128 bits de la entrada. El algoritmo de autenticación calcula un compendio de todos los datos del mensaje secreto, utilizado para la autenticación. MD5 consta de 64 operaciones, agrupadas en cuatro rondas de 16 operaciones. El algoritmo MD5 está diseñado para ser bastante rápido en máquinas de 32 bits. Este algoritmo fue descubierto por el profesor Ronald L. Rivest [28].

4.14.2. SHA

SHA-1 es otra de las funciones hash criptográficas más populares. Fue diseñada por la NSA y publicada en 1995. Trabaja con bloques de 512 bits y genera un resumen hash de 160 bits. Se utiliza en protocolos como TLS/SSL, PGP SSH y en IPsec. Muchas organizaciones recomiendan reemplazar SHA-1 por SHA-2 o SHA-3 y los principales proveedores de navegadores anunciaron sus intenciones de suprimir la compatibilidad con certificados SSL que utilizaban SHA-1 en 2017 debido a la identificación de algunas vulnerabilidades. Actualmente SHA-256 es uno de los hashes más utilizados [27].

4.14.2.1. SHA-256

El algoritmo SHA-256 es una función hash de criptografía y se utiliza en el certificado digital, así como en la integridad de los datos. SHA-256 ha sido desarrollado por el NIST (National Institute of Standards and Technology - Instituto Nacional de Estándares y Tecnología). El algoritmo SHA-256 toma como entrada un mensaje de longitud arbitraria inferior a 264 bits y produce como salida un compendio de mensajes de 256 bits de la entrada [28].

4.14.3. Comparativa entre MD5 y SHA

Las comparativa se ubican en la Tabla 2 tomada de [29]:

Tabla 2.
Comparativa entre SHA y MD5

Parámetros de comparación	SHA	MD5
Definición	SHA es un algoritmo de función hash criptográfica creado por NIST para facilitar la creación de resúmenes de mensajes.	MD5 fue creado por Ron Rivest y se utiliza para convertir mensajes de longitud indiscriminada en resúmenes de mensajes de 128 bits.
Forma completa	La abreviatura SHA significa Secure Hash Algorithm.	La abreviatura MD5 significa Message Digest.

Longitud máxima del mensaje	SHA puede convertir un mensaje de 264 - a 2128 bits para formar un resumen de mensaje de 160-512 bits.	MD5 puede convertir mensajes de cualquier longitud en un resumen de mensajes de 128 bits.
Seguridad	Como algoritmo hash criptográfico, SHA es más seguro que MD5.	MD5 es menos seguro que SHA y su versión mejorada SHA-1.
Velocidad	La versión original del algoritmo es más lenta que MD5. Sin embargo, sus entregas posteriores como SHA-1 ofrecen velocidades mucho más mejoradas.	MD5 es más rápido que la versión SHA original.
Vulnerabilidad	Menos vulnerable a las amenazas cibernéticas y los ataques de piratas informáticos.	Más vulnerable a las amenazas cibernéticas y los ataques de piratas informáticos.
Número de ataques	Menos ataques han podido violar el algoritmo.	Se han informado varios ataques graves.

4.16. ABCDE (agile block chain DApp engineering)

El enfoque ABCDE, tiene en cuenta la diferencia sustancial entre el desarrollo de software tradicional (el sistema de aplicaciones) y el desarrollo de contratos inteligentes, y separa las dos actividades. Para ambos desarrollos, ABCDE aprovecha un enfoque ágil, porque los métodos ágiles son adecuados para desarrollar sistemas cuyos requisitos no se entienden completamente desde el principio, o tienden a cambiar, como es el caso de las DApps. Esto descarta el uso de métodos basados en la planificación, como la cascada, y los métodos iterativos-incrementales que se basan en iteraciones más largas. ABCDE es un método ágil basado en Scrum, debido a la simplicidad de Scrum, a su popularidad y a que es, con diferencia, el método de desarrollo de software más utilizado, así como a la experiencia específica de los autores en el estudio y la aplicación de Scrum [30].

Los pasos del método de desarrollo de ABCDE, se centra en la cadena de bloques Ethereum y el lenguaje Solidity, que se muestran en la Figura 7 tomada de [30]. Obsérvese que la mayoría de los pasos se realizan varias veces, ya que el enfoque es iterativo e incremental. En la Figura 7, los círculos rosas representan las reuniones de planificación de sprint (**SPM**) que se celebran al principio de cada sprint (iteración), y las reuniones de revisión de sprint (**SRM**) que se celebran al final de los sprint. Los scrums diarios (reuniones de pie celebradas cada día) y las reuniones retrospectivas no aparecen.

El proceso de desarrollo que se observa en la Figura 7, es el siguiente tomado de [30]:

1. **Objetivo del sistema.** Escribir entre 10 y 30 palabras que resuman el objetivo y colocar en un lugar visible para todo el equipo. Tiene algunas similitudes con el "Objetivo del Sprint" que el método Scrum prescribe para encontrar y hacer visible al equipo, al principio de cada iteración, pero aquí el objetivo es para todo el sistema.

2. **Encontrar a los actores.** Identificar a los actores que van a interactuar con el sistema de la aplicación. Los actores son roles humanos, y sistemas o dispositivos externos que intercambian información con la DApp a construir.
3. **Historias de usuario.** Los requisitos del sistema se expresan en forma de historias de usuario (HU), para poder seguir el enfoque ágil clásico para la gestión de proyectos. En este paso, el sistema de aplicación que se está desarrollando debe ser considerado en su totalidad. La decisión de desarrollarlo utilizando una cadena de bloques, un conjunto de servidores, posiblemente en la nube, u otra arquitectura, no es importante aquí. En este punto, es útil, aunque no obligatorio, utilizar un Diagrama de Caso de Uso UML para mostrar gráficamente las relaciones entre los actores y los requisitos del sistema.
4. **Dividir el sistema en dos subsistemas.** En los siguientes:
   - Los contratos inteligentes que se ejecutan en la blockchain (Pasos (5)-(6)).
   - El sistema de aplicaciones, es decir, el sistema externo que interactúa con la blockchain, creando y enviando transacciones, y supervisando los eventos que se producen cuando un contrato inteligente ejecuta una función (Pasos (7)-(8)).

   En este punto, elaborar una arquitectura de todo el sistema, destacando qué datos deben colocarse en la chain y qué deben colocarse fuera de la chain. La directriz es que los contratos inteligentes deben gestionar los datos y el procesamiento que deben ser transparentes e inmutables para que los actores confíen en la DApp. Todos los demás datos, procesamientos e interfaces de usuario deben gestionarse fuera de la chain. En el caso de los datos que deben ser de confianza, pero que no pueden ser almacenados en la Blockchain debido a su transparencia en la chain, la privacidad de los datos puede lograrse utilizando el patrón de "almacenamiento de datos fuera de la cadena". La filtración del volumen de las transacciones y de las partes implicadas podría ser posible, y puede evitarse mediante otras técnicas de ofuscación.
5. **Diseño de los contratos inteligentes**. Este paso consiste en el diseño de los contratos inteligentes, utilizando en este caso el lenguaje Solidity. Esta actividad tiene unas características muy peculiares con respecto al diseño de software estándar. La actividad se realiza a través de iteraciones que incluyen la codificación y entrega de incrementos de contratos inteligentes, que son los requisitos elegidos para cada iteración. Se divide en subpasos, que se considera explícitamente y enumeramos siguiendo una secuencia lógica (pero que no necesariamente deben realizarse en una secuencia de "cascada"). Estos subpasos, así como todos los subpasos de los siguientes pasos principales, se derivan de la experiencia en el desarrollo de smart contract y DApp, y de las discusiones mantenidas con muchos desarrolladores de DApp. Son los siguientes:

a. Repite los pasos (2) y (3) (encontrar actores y HUs) centrándose sólo en los actores que interactúan directamente con los contratos inteligentes. Si los contratos inteligentes externos son utilizados por los contratos inteligentes del sistema en desarrollo, deben ser incluidos entre los actores. Para cada historia de usuario definida en este paso, definir también la(s) prueba(s) de aceptación correspondiente(s).
b. Definir ampliamente los contratos inteligentes que componen el subsistema contrato inteligente. Para cada contrato inteligente, indicar las responsabilidades para almacenar información y realizar cálculos, así como las colaboraciones relacionadas con otros contratos inteligentes. Para los sistemas no triviales, normalmente se necesitarán varios contratos inteligentes que interactúen. Considere también el uso de la herencia para abstraer las características comunes de los contratos inteligentes. Describir en detalle las colaboraciones con los contratos inteligentes externos, incluyendo las bibliotecas. Se utilizarán diagramas de clase UML con adiciones adecuadas.
c. Definir el flujo de mensajes y las transferencias de Ether entre los contratos inteligentes, los contratos inteligentes externos y el sistema de aplicaciones. Utilizar diagramas de secuencia UML aumentados para documentar estas interacciones. Si es necesario, definir los cambios de estado de los contratos inteligentes utilizando diagramas de estado UML.
d. Definir detalladamente la estructura de datos de cada contrato inteligente, su interfaz externa (Application Binary Interface, ABI) y los eventos relevantes que puede provocar.
e. Definir las funciones internas y privadas y sus modificadores, funciones especiales que suelen comprobar las condiciones previas necesarias para que una función pueda ejecutarse con seguridad.
f. Definir las pruebas y realizar las prácticas de evaluación de la seguridad. Este es un paso muy importante porque, como ya se ha explicado anteriormente, la mayoría de los contratos inteligentes son muy críticos y tratan con dinero.

6. **Codificación y prueba del sistema de contratos inteligentes**. Siguiendo el enfoque ágil, el sistema de contratos inteligentes se construye y se prueba de forma incremental. Las actividades de codificación y prueba son:
    a. Escribir y probar de forma incremental los contratos inteligentes. Debido a los estrictos requisitos de seguridad, esta actividad no puede realizarse de forma estrictamente incremental. En su lugar, partiendo de la estructura de datos y de las interfaces de los contratos inteligentes, se implementa y se prueba primero la arquitectura general del contrato inteligente del núcleo.

b. Realizar la evaluación de la seguridad y la optimización del gas del código escrito para el incremento.
c. Escribir pruebas unitarias (PUs) y pruebas de aceptación (PAs) automatizadas para los contratos inteligentes y requisitos implementados, respectivamente. Añade las nuevas pruebas al conjunto de pruebas. El entorno de pruebas más utilizado para Solidity es Truffle. Ejecutar todo el conjunto de pruebas para asegurarse de que las adiciones no rompen el sistema.

7. **Diseño del subsistema de interacción externa (Sistema de aplicaciones).** Este paso consiste en diseñar el sistema de aplicaciones, que interactúa con los usuarios y dispositivos, envía mensajes a la blockchain y puede gestionar sus propios repositorios (bases de datos y/o documentos). Esta actividad es muy similar al diseño de una aplicación web estándar. Sólo se añade otro actor, el blockchain, que puede recibir (pero no enviar) mensajes, y puede lanzar eventos. Hay que tener en cuenta que también en este caso hay que tener mucho cuidado con los aspectos de seguridad. De hecho, a menudo los hackers de los sistemas DApps se hacen explotando las debilidades del sistema de aplicaciones, en lugar de las de los contratos inteligentes.
    a. Redefinir los actores y los requisitos para el sistema de aplicaciones, partiendo de los recogidos en los pasos (2) y (3), añadiendo los nuevos actores representados por los contratos inteligentes que interactúan con el sistema de aplicaciones. Definir las pruebas de aceptación del sistema de aplicaciones.
    b. Diseñar la arquitectura de alto nivel del sistema de aplicaciones, incluyendo los niveles de servidor y cliente, y detallar la forma en que se accede a la cadena de bloques, estableciendo y ejecutando uno o más nodos, a través de un proveedor externo, o utilizando una wallet estándar.
    c. Definir la interfaz de usuario del sistema de aplicaciones, normalmente con un enfoque responsivo, para que pueda funcionar tanto en terminales móviles como en ordenadores. Contar con una interfaz de usuario atractiva es de suma importancia para lograr el éxito de la aplicación en el mercado.
    d. Definir cómo se descompone el sistema de la aplicación en módulos, sus interfaces y el flujo de mensajes entre ellos. Definir, si es necesario, los diagramas de estado de los módulos, y las acciones que realizan cuando los eventos son planteados por los contratos inteligentes. Definir la estructura y la memorización de los datos permanentes. Seleccionar qué datos se anclan a la blockchain, mediante la notarización de su hash digest a través del patrón "Off-Chain Data Storage". Definir la estructura de los datos o clases del sistema de aplicaciones, incluyendo el flujo de datos y control entre módulos. Las interacciones con los contratos inteligentes deben ser consistentes con el

análisis del Paso (5)(c). Esta actividad de diseño no se realiza por adelantado, sino a través de iteraciones que incluyen la codificación y la entrega de los incrementos del sistema de aplicaciones, implementando los requisitos elegidos para la iteración. Debido a los estrictos requisitos de seguridad, esta fase de diseño debe ser bastante detallada y realizarse de forma coherente con las actividades correspondientes del diseño de los contratos inteligentes. Los diagramas de clase y secuencia UML pueden ayudar a diseñar y documentar también este sistema.

e. Realizar una evaluación de la seguridad del sistema externo.

8. **Codificación y prueba del sistema de aplicaciones.** Paralelamente al sistema de contratos inteligentes, se construye y prueba el sistema de aplicaciones, utilizando el mismo enfoque de desarrollo de contratos inteligentes. Si los desarrollos de contratos inteligentes y del sistema de aplicaciones se realizan de forma iterativa, cada dos o tres iteraciones se deben integrar los resultados de las dos ramas, como se muestra en la Figura 7. Las actividades que se desarrollan en paralelo son:
    a. Implementar de forma incremental los requisitos del sistema de aplicaciones. Este paso pertenece al "flujo correcto" de ABCDE (véase la Figura 7), y no difiere de la implementación de una aplicación web.
    b. Realice la evaluación de seguridad del código escrito para el incremento.
    c. Escriba PUs y PAs automatizados para los requisitos implementados. Añadir las nuevas pruebas al conjunto de pruebas. Ejecute todo el conjunto de pruebas para asegurarse de que las adiciones no rompen el sistema.
9. **Integrar, probar y desplegar el Sistema DApp.** Para integrar los Contratos Inteligentes y el Sistema de Aplicaciones, los sistemas globales construidos hasta ese momento deben desplegarse en una blockchain local o en una red de pruebas, y deben ejecutarse pruebas de integración para comprobar si todos los componentes interactúan juntos como se espera (por ejemplo, los eventos planteados por los contratos inteligentes son recogidos por el Sistema de Aplicaciones, los mensajes enviados por el Sistema de Aplicaciones activan las transacciones de la blockchain que son validadas y ejecutadas correctamente, etc.)

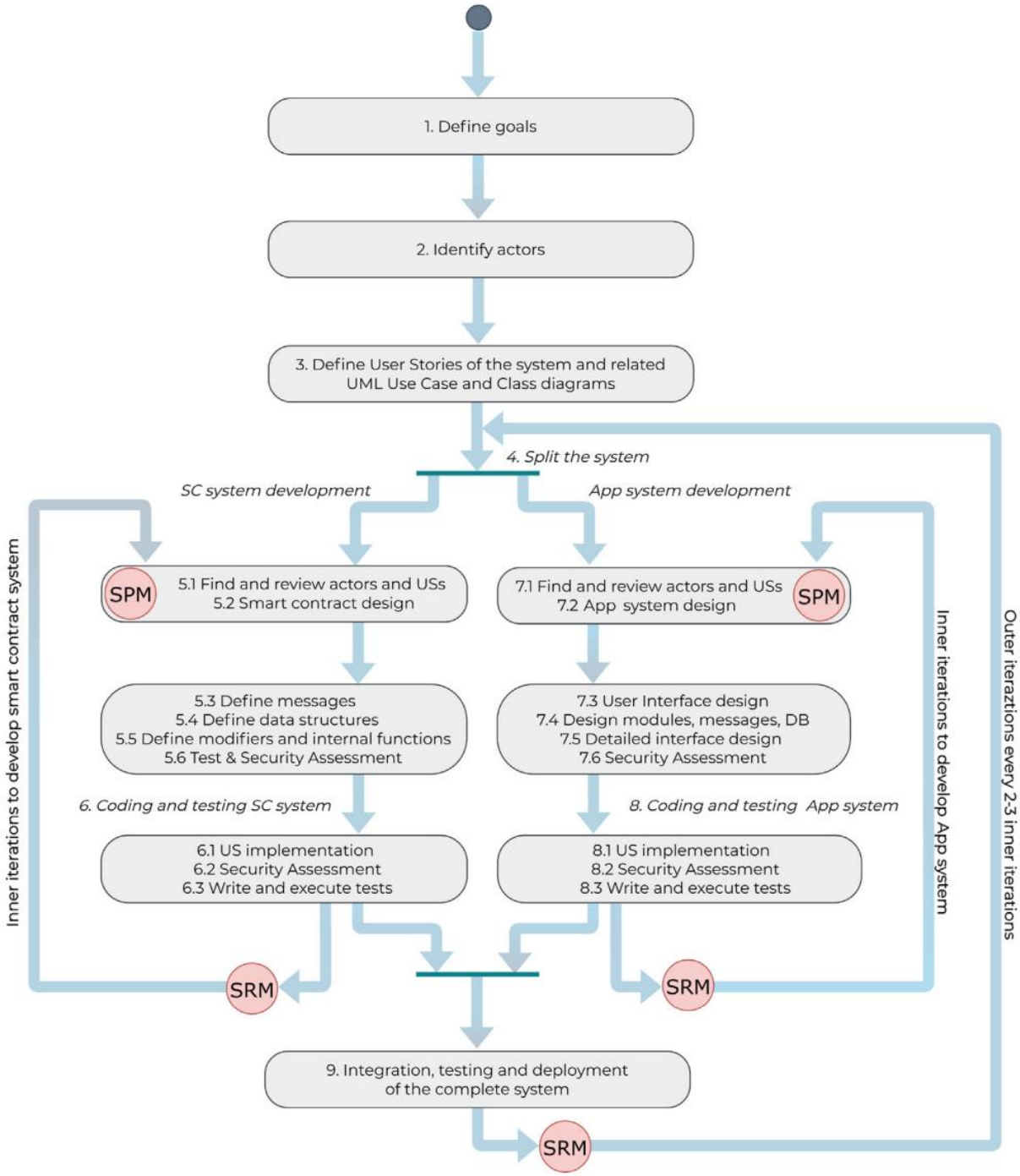


Figura 7. El proceso de la metodología de desarrollo ABCDE [30].

4.17. Trabajos relacionados

En Ecuador actualmente no se encuentra un proyecto o solución informática relacionada a la validación de certificados académicos digitales por tecnología blockchain. Por ende, se ha buscado a nivel de Latinoamérica donde existen proyectos que usen la tecnología Blockchain para validar documentos, dando mayor confiabilidad y seguridad a los usuarios, mismos que se hacen referencia en la Tabla 3 con su respectivo hallazgo.

Tabla 3.
Trabajos relacionados

Titulo	Hallazgo	Referencia
Certificados Académicos Digitales mediante Blockchain	En este documento se ilustra el desarrollo de software de la plataforma Stampo que significa estampilla y mediante los diferentes capítulos se detalla el diseño y la arquitectura de la generación de los certificados académicos digitales basadas en las actividades de los estudiantes.	[31]
Certificación digital de documentos académicos basada en la tecnología de blockchain	En este artículo se describe las principales características de Blockchain y cómo podrían ser utilizadas en diversos procesos de expedición de documentos oficiales, como es la generación de documentos que requieran ser autenticados y/o verificados.	[9]
Aplicación web para la gestión de diplomas digitales en centros de capacitación mediante firma electrónica y blockchain	Es una aplicación web que permite gestionar el registro, emisión y entrega de certificados de capacitación de manera eficiente.	[32]
Diseño de la Arquitectura de un Sistema de Contratos Inteligentes Basada en la Tecnología Blockchain Aplicada al Proceso de Registro de Estudiantes en el Sistema de Educación Colombiano	En este artículo buscan aplicar la tecnología Blockchain en un sistema de información que podría aportar a la solución de la problemática del registro de personas en el sistema de educación colombiano.	[33]
Smart Contracts for user registration on Ethereum technology: Systematic Literature Review	Es una Revisión Sistemática de la Literatura (RSL), que busca la forma de desarrollar contratos inteligentes dentro del módulo de registro de usuarios de un sistema, además de detectar cuáles son las herramientas que proporcionarán una mejor aplicación de los contratos.	[34]
Registro de títulos académicos mediante una aplicación basada en Blockchain y Smart Contracts	Es una propuesta de una aplicación fundamentado en la tecnología Blockchain y en los contratos inteligentes para reproducir el proceso de asignar títulos académicos a estudiantes.	[35]

5. Metodología

El Trabajo de Titulación logró obtener como resultado realizar la validación correcta de la autenticidad de los certificados académicos digitales. El cual surgió en base al problema de investigación y pregunta: **¿Cómo validar la autenticidad de los certificados académicos digitales?,** a partir de la cual se dio inicio al desarrollo del TT hasta la finalización del mismo.

Además, se elaboró, definió y sistematizó el conjunto de metodología, métodos, técnicas, estándares y materiales que establecieron una secuencia de pasos ordenados para el desarrollo del proyecto. Las cuales se describen a continuación:

5.1. Metodología de desarrollo de software

Para la elaboración del presente Trabajo de Titulación, se utilizó la metodología de desarrollo agile block chain DApp engineering (**ABCDE**) la misma que se identificó como la más apta de acuerdo a las características que tiene la misma. Para conocer a más detalle la metodología se puede observar en el punto **0 del Marco teórico.**

5.2. Proceso

El proceso para alcanzar el objetivo general del TT se detalla a continuación, mencionando cada uno de los objetivos con sus actividades:

1) Definir el módulo de software
    a) Definición del objetivo del módulo de software que es equivalente al objetivo del TT, correspondiente a la fase 1 de la metodología de desarrollo de software ABCDE (ver el **punto 6.1.1 del Objetivo 1 la sección Resultados**).
    b) Determinar los actores que intervienen el módulo de software, correspondiente a la fase 2 de la metodología de desarrollo de software ABCDE (ver el **punto 6.1.2 del Objetivo 1 la sección Resultados**).
    c) Determinación de los requerimientos del módulo de software, en el que realizo el documento de especificación de requisitos de la IEEE 830, correspondiente a la fase 3 de la metodología de desarrollo de software ABCDE (ver el **punto 6.1.3 del Objetivo 1 la sección Resultados**).
    d) División del módulo de software, además de determinar las tecnologías necesarias para poder proceder al siguiente objetivo, correspondiente a la fase 4 de la metodología de desarrollo de software ABCDE (ver el **punto 6.1.4 del Objetivo 1 la sección Resultados**).
2) Desarrollar el módulo de software
    a) Configuración inicial tomando en cuenta las tecnologías mencionadas en el diagrama obtenido en la fase 4 de la metodología ABCDE (ver el **punto 6.2.3 del Objetivo 2 la sección Resultados**).

b) Diseño de los subsistemas de contratos inteligentes y de aplicaciones, correspondiente a las fases 5 y 7 de la metodología de desarrollo ABCDE (ver los **puntos 6.2.5 y 6.2.7 del Objetivo 2 la sección Resultados**).
c) Codificación y pruebas unitarias de cada subsistema, tanto los contratos inteligentes y el de aplicaciones, correspondiente a las fases 6 y 8 de la metodología de desarrollo ABCDE (ver los **puntos 6.2.6 y 6.2.8 del Objetivo 2 la sección Resultados**).

3) Probar la solución en una testnet
a) Generación y ejecución de Pruebas de Integración, correspondiente a la fase 9 de la metodología de desarrollo ABCDE (ver el **punto 6.3.1.1 del Objetivo 3 la sección Resultados**).
b) Generación y ejecución de Pruebas Funcionales y Aceptación, correspondiente a la fase 9 de la metodología de desarrollo ABCDE (ver el **punto 6.3.1.2 del Objetivo 3 la sección Resultados**).
c) Despliegue del módulo en la testnet Rinkeby y el servicio en la nube de Microsoft Azure, correspondiente a la fase 9 de la metodología de desarrollo ABCDE (ver el **punto 6.3.1.3 del Objetivo 3 la sección Resultados**).

5.3. Métodos

5.3.1. Analítico

Este método radica en la separación de un todo descomponiéndolo en sus partes o elementos para observar las causas, naturaleza y los efectos [36], se utilizó para descomponer el Trabajo de Titulación en diversas fases las cuales fueron establecidas como los objetivos específicos, acompañadas de sus actividades.

5.3.2. Investigación-acción

La metodología investigación acción se estructura por ciclos y flexible, debido a que es válido e incluso necesario realizar ajustes conforme se avanza en el estudio, hasta que se alcanza el cambio o la solución al problema. En el cual lo primero es detectar un problema; segundo se procede a elaborar un plan para solucionarlo; tercero implementar y evaluar el plan; y cuarto una realimentación. Adaptándose en las diferentes fases del Trabajo de Titulación.

5.3.3. Experimentales Controlados

5.3.3.1. Replicación

Permite desarrollo de múltiples versiones de un producto, en este caso el módulo de software hasta llegar a que cumpla con el documento de especificación de requisitos con estándar IEEE 830.

5.3.3.2. Simulación

Este método permite la ejecución de un producto con datos artificiales [37]. Por lo tanto, permitirá simular el módulo de software en una testnet de blockchain, justamente cumpliendo el objetivo número tres denominado probar la solución en una testnet de blockchain para el módulo de software de validación de certificados académicos digitales.

5.3.4. Métodos Observacionales

5.3.4.1. Análisis estático

El método permite realizar un examen minucioso a la estructura del producto, en nuestro caso el módulo de validación de autenticidad de certificados académicos, con la finalidad de descubrir errores.

5.3.4.2. Caso de estudio

Este método tiene el objetivo de usar herramientas para monitorear un proyecto en profundidad. Así aporta para determinar la aceptación del Trabajo de Titulación y determinar si las herramientas usadas son las adecuadas.

5.4. Técnicas

5.4.1. Entrevista

Es uno de los medios tradicionales utilizada para la elicitación de requerimientos y/o requerimientos, la técnica permite abordar asuntos relacionados con el proceso de validar la autenticidad de los certificados académicos digitales, con el fin de tomar y elicitar los requisitos para el módulo de software.

5.4.2. Reuniones

Esta técnica permite a través de las personas interesadas e inmersas en el Trabajo de Titulación, revisar los diferentes avances en cada una de las fases, siempre generando una retroalimentación.

5.5. Estándares

5.5.1. IEEE 830

Es un conjunto de recomendaciones para la especificación de los requerimiento o requisitos de software. Este sirve para poder cumplir el primer objetivo específico de definir el módulo de software para la validación de certificados académicos digitales usando la Ingeniería de Requisitos del Trabajo de Titulación.

5.5.2. ERC 20

Una estructura que permite que la generación de contratos inteligente, anexados en el segundo objetivo específico del TT, puedan ser diseñado con interoperabilidad y compatibilidad entre los diferentes tokens de la red Ethereum e ir mejorando el ecosistema.

5.6. Materiales

5.6.1. Recurso Humano

Para el desarrollo del Trabajo de Titulación estuvieron inmersas personas en diferentes ámbitos que se observan en la Tabla 4.

Tabla 4.
Recurso Humano

Nombre	Descripción
Edgar Patricio Sánchez Malla	Estudiante a cargo de la ejecución del proyecto
Ing. Cristian Ramiro Narváez Guillen, Mg.Sc.	Docente director del Proyecto de Trabajo de Titulación

5.6.2. Recursos de Software y Hardware

En el desarrollo del Trabajo de Titulación se empleó los recursos de Software y Hardware que se puede observar en la Tabla 5 y la Tabla 6.

Tabla 5.
Recursos de Software

Recurso	Descripción	Link de acceso
Zoom	Software de videochat que permitió el trabajo colaborativo, revisiones del Trabajo de Titulación, entrevistas, entro otros.	https://zoom.us
OneDrive	Es un servicio que permitió el almacenamiento y trabajo colaborativo de documentos.	https://www.microsoft.com/es-ww/microsoft-365/onedrive/online-cloud-storage
Lucidchart	Software que permitió la construcción de diferentes diagramas o figuras.	https://www.lucidchart.com
InVision	Herramienta para el desarrollo de prototipos que permitió realizar el desarrollo de la interfaz gráfica.	https://www.invisionapp.com
Node js	Entorno de ejecución para JavaScript	https://nodejs.org/es
Ethereum	Plataforma de Blockchain para el desarrollo	https://ethereum.org/es
Etherscan	Plataforma para observar la trazabilidad de las transacciones (en este caso con las testnet Rinkeby)	https://rinkeby.etherscan.io
Rinkeby	Testnet de Ethereum para el despliegue	https://www.rinkeby.io/#stats
Infura	Suite para obtener las credenciales de conexión con la testnet Rinkeby	https://infura.io
Truffle Suite	Herramienta con varias tecnologías que permitió el desarrollo del módulo de software.	https://www.trufflesuite.com
Visual Studio Code	Editor de código fuente	https://code.visualstudio.com

GitHub	Plataforma web que permitió el trabajo colaborativo y control de versiones del módulo de software.	https://github.com
GitKraken	Es un software con interfaz más amigable para Git, con integraciones a GitHub.	https://www.gitkraken.com
MetaMask	Extensión para navegadores web, es la puerta de entrada para las aplicaciones de Blockchain.	https://metamask.io
web3.js	API JavaScript de Ethereum, es una librería que permitió interactuar con un nodo local o remoto de Ethereum utilizando HTTP, IPC o WebSocket.	https://web3js.readthedocs.io
js-sha256.js	Una simple función hash SHA-256 / SHA-224 para JavaScript soporta la codificación UTF-8.	https://www.npmjs.com/package/js-sha256
sweetalert2	Librería para los mensajes de alerta	https://sweetalert2.github.io
Jest	Marco de pruebas de JavaScript.	https://jestjs.io
Enzyme	Es una herramienta de pruebas de JavaScript para React.	https://enzymejs.github.io/enzyme
Cypress	Es una herramienta de testing para diferentes tipos de pruebas.	https://www.cypress.io
@truffle/hdwallet-provider	Librería para generar la conexión con la wallet (metamask) para firmar transacciones para direcciones derivadas de un mnemónico de 12 o 24 palabras.	https://www.npmjs.com/package/@truffle/hdwallet-provider
Microsoft Azure	El servicio de computación en la nube para probar y desplegar el módulo de software.	https://azure.microsoft.com/es-es

Tabla 6.
Recursos de Hardware

Recurso	Descripción
Laptop	Equipo fundamental para el desarrollo del Trabajo de Titulación.

5.6.3. Insumos

Los insumos necesarios para la ejecución del Proyecto de Trabajo de Titulación se muestran en la Tabla 7.

Tabla 7.
Insumos

Insumos	Descripción
Materiales de oficina	Usados como complemento para el desarrollo del Trabajo de Titulación.
Internet	Utilizada para la comunicación, investigación y acceso a otras herramientas.

6. Resultados

En la presente sección se detalla los resultados obtenidos del desarrollo del Trabajo de Titulación, el mismo que se llevó a cabo de acuerdo a la metodología de desarrollo ABCDE y los objetivos específicos planteados.

6.1. Objetivo 1

Definir el módulo de software para la validación de certificados académicos digitales usando la Ingeniería de Requisitos.

En esta etapa intervienen las siguientes fases de la Figura 8:

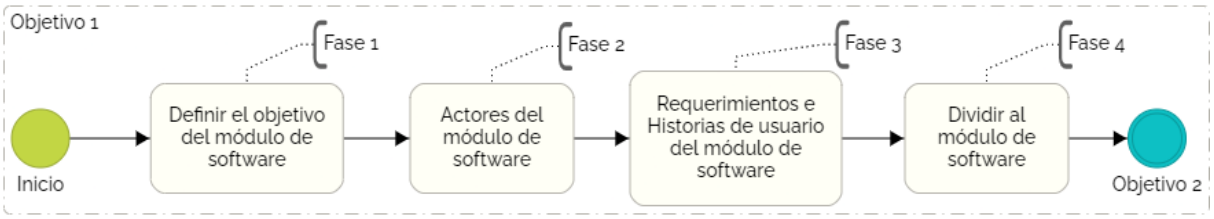


Figura 8. Proceso de desarrollo para el objetivo 1.

Las fases de la Figura 8 que se encuentran son las de: **objetivo** (fase 1), **actores** (fase 2), **requerimientos e historias de usuario** (fase 3) y **dividir el sistema** (fase 4) para mayor detalle se encuentran en los puntos 1, 2, 3 y 4 de la sección **ABCDE (agile block chain DApp engineering)** del Marco teórico.

6.1.1. Fase 1: Objetivo del módulo de software

Se relaciona al objetivo general del Trabajo de Titulación que es “Implementar un módulo de software para la validación de certificados académicos digitales por tecnología blockchain”.

6.1.2. Fase 2: Actores del módulo de software

El módulo de software tiene lo siguientes actores:

- **Usuario final:** El usuario que requiere validar el certificado académico digital.
- **Usuario administrador:** El usuario dueño del despliegue de los contratos que está autorizado para registrar el certificado académico digital.
- **SC-VC (Smart Contract – Validate Certificates):** contrato(s) inteligente(s) en la blockchain de Ethereum (testnet local o Rinkeby) que tiene las funcionalidades de registrar y validar un certificado académico digital.

6.1.3. Fase 3: Requerimientos e Historias de usuario del módulo de software

En la presente fase se describen los siguientes puntos que son: los requisitos funcionales y no funcionales (extraídos del **Anexo 1.** Especificación de requisitos de software) siguiendo estándar IEEE 830. Los mismos que fueron extraídos en base a la observación del funcionamiento de la consulta de certificados académicos digitales como el Título de bachiller

en sistema de servicios del Ministerio de Educación, la experiencia del estudiante y el director del TT en el proceso de validación de certificados académicos. Así mismo, las historias de usuario (**Anexo 2.** Descripción de Historias de usuario), y el diagrama de casos de usos.

6.1.3.1. Requisitos funcionales

En la Tabla 8 se muestra los requisitos funcionales de módulo de software para la validación de autenticidad de certificados académicos digitales. Es importante mencionar que se usa la función SHA-256 con base en la comparativa del punto **4.14.3 del Marco teórico**, debido a que en una DApp es importante el tamaño de la información porque a mayor tamaño mayor costo en la transacción. Además, es el más usado y es más seguro que el MD5.

Tabla 8.
Requisitos funcionales

Código	Requisito	Descripción
RF01	Generar HASH de certificado académico digital	El módulo de software generará un SHA-256 del certificado académico digital (archivo PDF).
RF02	Obtener DNI del usuario	El módulo de software recibirá el DNI del usuario al que pertenece el certificado académico digital.
RF03	Registrar CAD en la Blockchain	El módulo de software por medio de un formulario y del contrato inteligente (smart contract) almacenará el hash del CAD y el DNI del usuario en la Blockchain de Ethereum, solo por la cuenta que despliegue los contratos inteligentes (usuario administrador).
RF04	Obtener DNI y CAD a validar	El módulo de software permitirá recibir a través de un formulario web el DNI del usuario y un archivo en formato PDF del CAD a validar.
RF05	Generar HASH del CAD a validar	El módulo de software generará un SHA-256 del CAD a validar en formato PDF, recibido del formulario web.
RF06	Obtener CADs registrados	El módulo de software por medio del contrato inteligente (smart contract) consultará los CADs registrados en la plataforma blockchain.
RF07	Validar el certificado académico digital	El módulo de software buscará en los CADs obtenidos del RF06, si existe un CAD idéntico al generado en el RF04.
RF08	Mostrar resultado	El módulo de software alertará con un mensaje en el sitio web, el resultado de la validación de la autenticidad del certificado académico digital.

6.1.3.2. Requisitos no funcionales

En la Tabla 9 se presenta los requisitos funcionales de módulo de software para la validación de autenticidad de certificados académicos digitales.

Tabla 9.
Requisitos no funcionales

Código	Requisito	Descripción
RNF01	Rendimiento	El módulo de software debe proveer un tiempo de respuesta aceptable de 2 a 7 segundos aproximadamente. La transacción tarda de 2 a 5 segundos en una testnet de Blockchain de Ethereum y en un ambiente real de 15 segundos a 5 minutos.

RNF02	Usabilidad	El módulo de software debe proveer una interfaz amigable e intuitiva, haciendo que el proceso sea comprensible y fácil de llevar a cabo. Además, de permitir ser utilizado en diversos navegadores web.
RNF03	Fiabilidad	El módulo de software debe proveer la disponibilidad las 24 horas del día y los 7 días de la semana, y en caso de que se presente algún error, se debe recuperar en el menor tiempo posible. El módulo de software debe permitir recuperar los datos que se vean afectados en el caso de alguna falla, respecto al tiempo y esfuerzo que este genere.
RNF04	Seguridad	El módulo de software debe garantizar disminuir las vulnerabilidades de ataques de fuerza bruta. Garantizar la seguridad del módulo de software con respecto a la información y datos que se manejan tales sean documentos o archivos.

6.1.3.3. Historias de usuario

En la Tabla 10 se muestra el resumen de las historias de usuario del módulo de software para la validación de autenticidad de certificados académicos digitales. El desarrollo completo de las Historias de usuario se encuentra en el **Anexo 2.** Descripción de Historias de usuario.

Tabla 10.
Historias de usuario

Identificador de la historia	Rol	Característica / Funcionalidad	Razón / Resultado
HU01	Como usuario final	Necesito validar un CAD	Con la finalidad de validar la autenticidad del CAD
HU02	Como usuario administrador	Necesito registrar un CAD	Con la finalidad de registrar un CAD en la red Blockchain
HU03	Como Contrato inteligente (SM-VC)	Necesito validar un CAD	Con la finalidad de validar si el CAD se encuentra registrado en la red Blockchain
HU04	Como Contrato inteligente (SM-VC)	Necesito registrar un CAD	Con la finalidad de validar si el CAD se encuentra registrado en la red Blockchain
HU05	Como WEB	Necesito generar el SHA-256 del CAD	Con la finalidad de generar el SHA-256 del CAD

6.1.3.4. Caso de usos

En la Figura 9 se muestra los diferentes actores con su respectivo caso de uso.

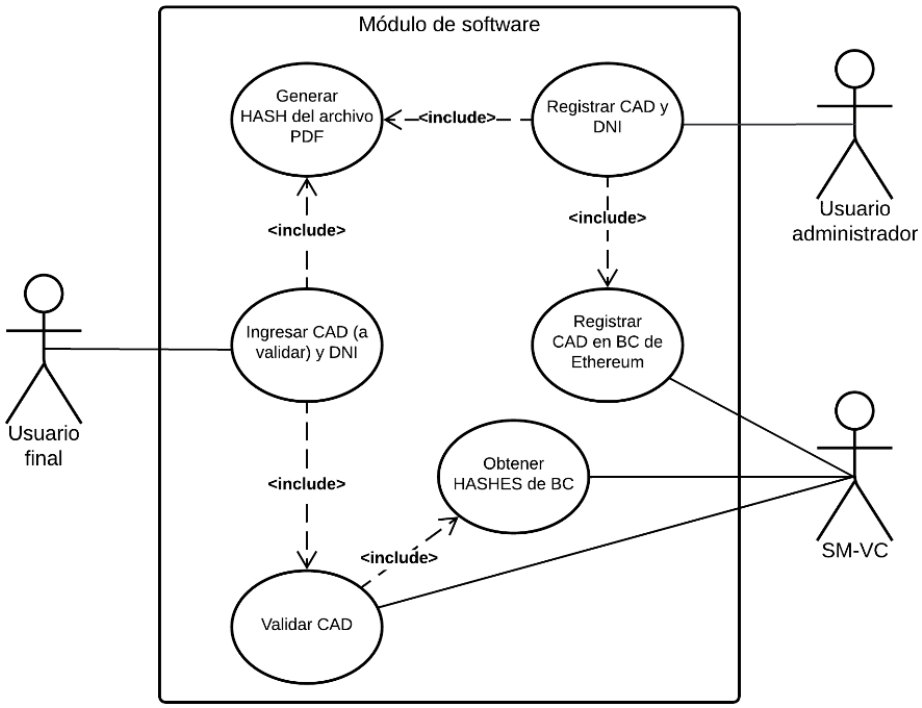


Figura 9. Diagrama de casos de uso del módulo de software.

6.1.4. Fase 4: Dividir al módulo de software

El módulo de software se procede a subdividir de la siguiente manera:

- Subsistema de los contratos inteligentes que se ejecutan en la blockchain (las fases que intervienen en este punto son 5 y 6).
- Subsistema de aplicaciones que interactúa con el usuario final (las fases que intervienen en este punto son 7 y 8).

Con base en los dos puntos anteriores se procede a determinar dos diagramas de la arquitectura que se detalla en el siguiente punto.

6.1.4.1. Diagrama de la arquitectura del módulo de software

En esta fase se procede a plantear el diagrama de la arquitectura del módulo de software para comprender la arquitectura revisar en la sección del Marco teórico el **punto 4.7.** Comparativa de una aplicación web tradicional vs una DApp (Blockchain).

En la Figura 10 y Figura 11 se encuentran dos diagramas de la arquitectura del módulo de software para la validación de certificados académicos digitales, con conexión a la testnet local Ganache y la testnet Rinkeby de Ethereum respectivamente. En la que se observa las diferentes tecnologías en su respectivo subsistema.

En el cual por medio del Frontend (HTML, CSS y JavaScript), web3.js, React y la extensión de MetaMask instalada en el navegador, permite invocar las funciones de un contrato inteligente desarrollado bajo el lenguaje de programación Solidity y la herramienta de Truffle.

Que permite generar la conexión con la Blockchain de Ethereum de la testnet correspondiente. Además, que una vez desplegada en la testnet Rinkeby para el despliegue ya se puede interactuar a través del navegador de la aplicación móvil de MetaMask. Eventualmente se debe tener una cartera (wallet) generada en MetaMask.

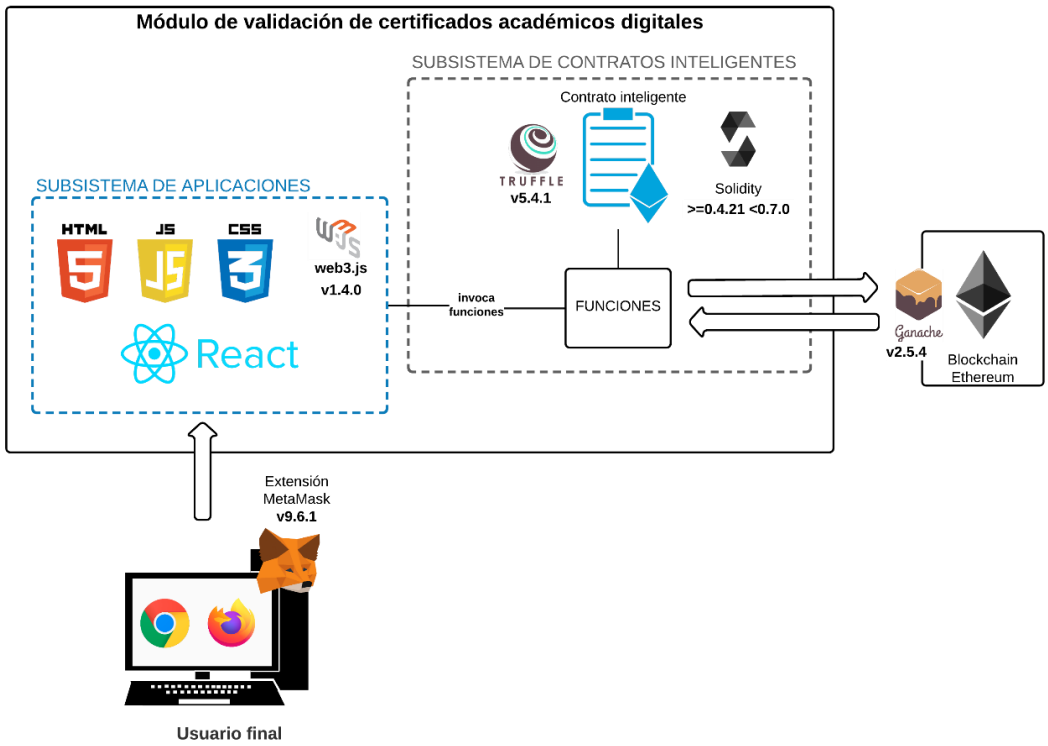


Figura 10. Arquitectura del módulo de software en una testnet blockchain local (Ganache).

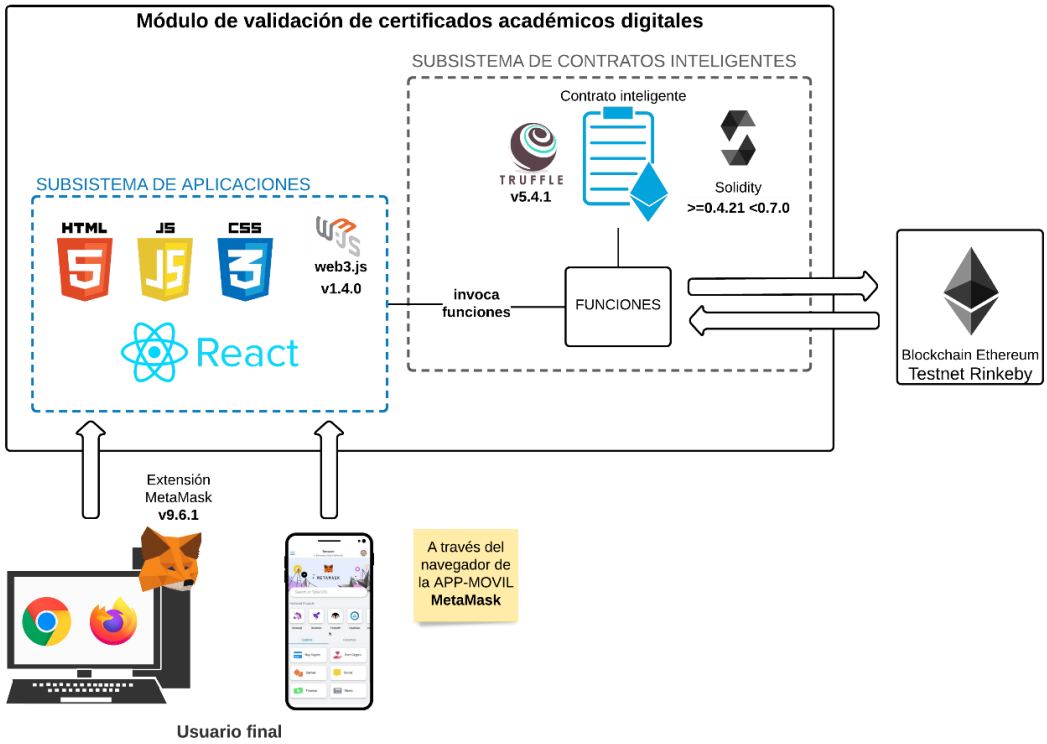


Figura 11. Arquitectura del módulo de software en una testnet blockchain Ethereum Rinkeby.

6.2. Objetivo 2

Desarrollar el módulo de software para la validación de certificados académicos digitales por medio de la plataforma Blockchain Ethereum Virtual Machine.

En el presente objetivo intervienen las siguientes fases de la Figura 12:

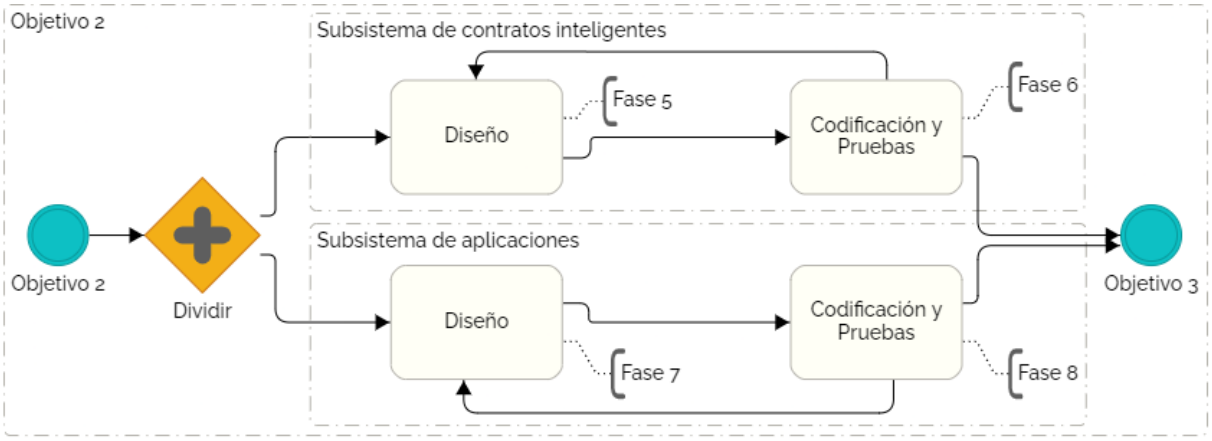


Figura 12. Proceso de desarrollo para el objetivo 2.

Las fases de la Figura 12 se distribuyen de la siguiente forma:

- En el subsistema de los contratos inteligentes que intervienen las fases de: **diseño del subsistema de contratos inteligentes** (fase 5) y **codificación y prueba del subsistema de contratos inteligentes** (fase 6).
- En el subsistema de aplicaciones que intervienen las fases de: **diseño del subsistema de interacción externa (Sistema de aplicaciones)** (fase 7) y **codificación y prueba del sistema de aplicaciones** (fase 8).

Para mayor detalle se encuentran en los puntos 5, 6, 7 y 8 de la sección **ABCDE (agile block chain DApp engineering)** del Marco teórico.

6.2.1. Iteraciones (sprints) del módulo de software

La metodología ABCDE al ser en la mayoría de sus fases iterativa e incremental, y puntualmente en las fases de la 5 a la 8 que corresponden al objetivo 2. Para cada subsistema se plantearon sprints que permitieron cumplir adecuadamente como se observa en la Tabla 11 y Tabla 12.

Tabla 11.
Iteraciones (sprints) del subsistema de contratos inteligentes

No. Sprints	Nombre	Duración en horas
1	Configuración inicial	10
2	Diseño del subsistema	10
3	Codificar el contrato inteligente	40
4	Configuración de la testnet local	10

5	Compilación del contrato	10
6	Migración del contrato	10
7	Pruebas unitarias	20

Tabla 12.
Iteraciones (sprints) del subsistema de aplicaciones

No. Sprints	Nombre	Duración en horas
1	Configuración inicial	10
2	Conexión con web3	10
3	Conexión con contratos inteligentes	10
4	Conexión con las funcionalidades del contrato inteligentes	20
5	Generar SHA-256 del certificado académico digital	10
6	Diseño de interfaz de acuerdo a la línea gráfica de los sistemas informáticos de la Universidad Nacional de Loja.	20
7	Pruebas unitarias	20

6.2.2. Control de versiones del módulo de software

Por medio del software GitKraken en la Figura 13 se observa los diferentes commits realizados, esto se relaciona en los diferentes **sprints** realizados durante el desarrollo de los subsistemas en base a la metodología ABCDE. Todo esto se encuentra en el repositorio de GitHub ubicado en el siguiente enlace: https://github.com/EdgarPatricio/M-dulo-de-software-de-VCAD-por-Blockchain.git.

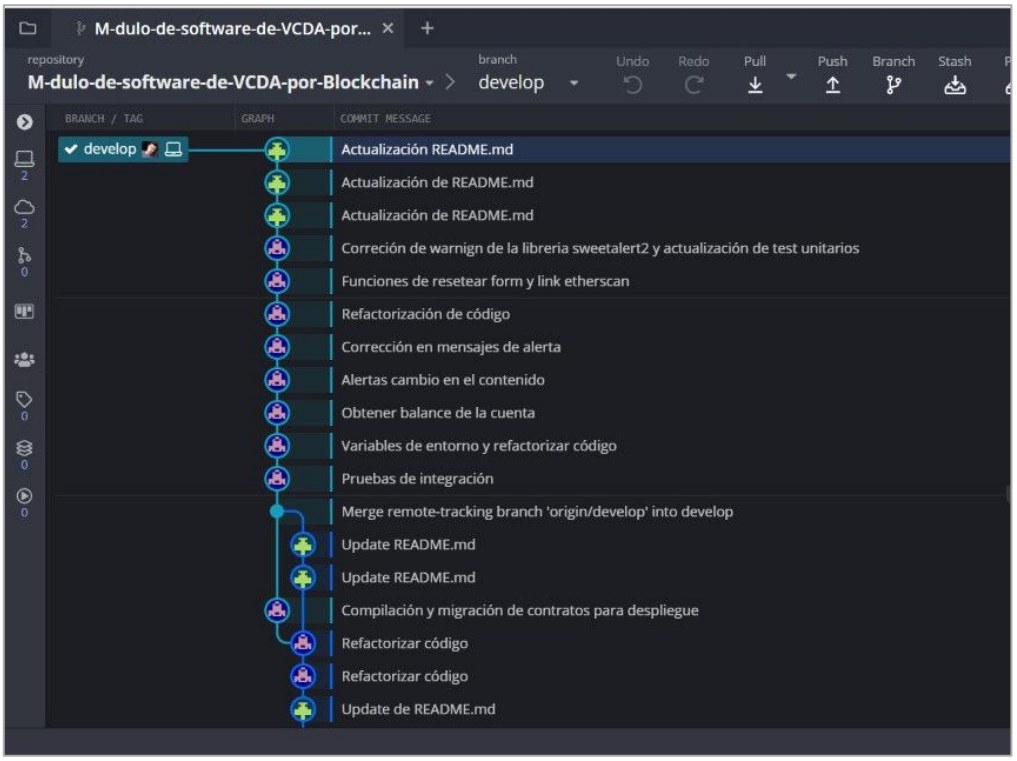


Figura 13. Control de versiones con GitKraken.

6.2.3. Configuración inicial

Para los dos subsistemas se realizó en conjunto la siguiente configuración inicial, que se observa en la Tabla 13:

Tabla 13.
Configuración inicial

Tareas	Estado
Instalar node.js	✓
Instalar la librería web3.js	✓
Instalar Truffle	✓
Instalar GIT	✓
Instalar y configurar GitKraken	✓
Instalar un Entorno de desarrollo integrado de preferencia, en este caso se usó Visual Studio Code, con dos plugin denominados: • "Solidity" • "Solidity Visual Developer"	✓
Instalar y configurar Metamask	✓
Instalar y configurar el box de react-truffle	✓
Instalar la librería js-sha256.js en la carpeta client	✓
Instalar la librería sweetalert2 en la carpeta client	✓
Instalar la librería animate.css en la carpeta client	✓
Instalar la librería enzyme en la carpeta client	✓
Instalar el framework de testing Cypress en la carpeta client	✓

6.2.4. Estructura del módulo de software

El módulo de software de la validación de autenticidad de certificados académicos digitales tiene la estructura que se muestra a continuación. Todo esto generado por medio del box de react-truffle, recomendación de Truffle Suite.

6.2.4.1. Estructura del subsistema de contratos inteligentes

La estructura del subsistema de contratos inteligentes se observa en la Figura 14 y en la Tabla 14 se observa a detalle su composición.

Figura 14. Estructura del subsistema de contratos inteligentes.

Tabla 14.
Estructura de subsistema de contratos inteligentes

Nombre	Descripción
Módulo de software de VCAD por Blockchain	Carpeta que contiene todo el proyecto del módulo de software del TT.
client	**Subsistema de aplicaciones.**
contract	Carpeta donde que contiene los diferentes contratos inteligentes.
migrations	Carpeta que contiene archivos js que permiten la migración de los diferentes contratos.
node_modules	Carpeta que almacena los paquetes que utiliza Node JS y donde se guardan todas las dependencias para subsistema.
test	Carpeta que almacena las pruebas unitarias.
.env	Archivo que contiene variables de entorno para el despliegue de los contratos inteligentes.
.gitattributes	Archivo de texto simple que da attributes a los nombres de ruta.
.gitignore	Archivo que especifica los archivos o carpetas que git no tiene en cuenta y no almacena las modificaciones que se han realizado.
LICENSE	Licencia MIT.
LICENSE.md	Licencia MIT.
package-lock.json	Fichero que contiene información acerca del subsistema (nombre, versión, etc) además de listar los paquetes de los que depende.
README.md	Archivo que contiene información acerca del subsistema.
truffle-config.js	Es un archivo js que contiene las configuraciones de truffle.

6.2.4.2. Estructura del subsistema de aplicaciones

La estructura del subsistema de aplicaciones se observa en la Figura 15 y en la Tabla 15 se observa a detalle.

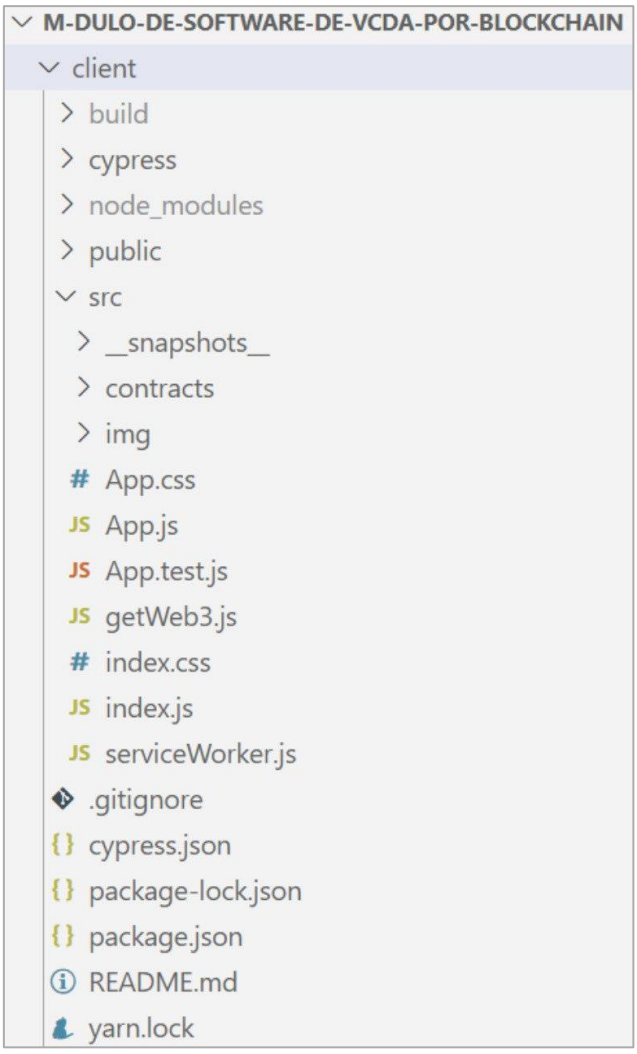


Figura 15. Estructura del subsistema de aplicaciones.

Tabla 15.
Estructura de subsistema de aplicaciones

Nombre	Descripción
build	Carpeta que almacena el software construido para despliegue en un hosting.
cypress	Carpeta que almacena los archivos para las pruebas de integración.
node_modules	Carpeta que almacena los paquetes que utiliza Node JS y donde se guardan todas las dependencias para subsistema.
public	Carpeta que contiene los archivos que contine el archivo HTML base, fuente, estilos, entre otros.
src	Carpeta que contiene los elementos de conexión con web3, contratos y las funciones de front-end.
snapshots	Carpeta que almacena una copia exacta del renderizado del front-end para pruebas unitarias.
contracts	Carpeta que contiene los contratos inteligentes compilados y migrados en formato JSON.
img	Carpeta que contiene imágenes.
App.css	Archivo de estilo.
App.js	Archivo que contiene las funcionalidades principales del subsistema.
App.test.js	Archivo que contiene los métodos de las pruebas unitarias.

getWeb3.js	Archivo para la conexión a web3.
index.css	Archivo de estilo.
index.js	Archivo base para React JS.
serviceWorker.js	Archivo opcional para ejecutarse en segundo plano (Proxy de red programable)
.gitignore	Archivo que especifica los archivos o carpetas que git no tiene en cuenta y no almacena las modificaciones que se han realizado.
cypress.json	Archivo de configuración para cypress.
package-lock.json	Fichero que contiene información acerca del subsistema (nombre, versión, etc) además de listar los paquetes de los que depende.
package.json	Archivo que contiene los paquetes o dependencias del subsistema.
README.md	Archivo que contiene información acerca del subsistema.
yarn.lock	Archivo que guarda las versiones de dependencia exactas.

A continuación, se detalla las diferentes fases de la metodología ABCDE con sus respectivos resultados que intervienen en el objetivo 2.

6.2.5. Fase 5: Diseño del subsistema de contratos inteligentes

En esta fase se encuentra el **diagrama de clases** que tiene la finalidad de diseñar la estructura del módulo de software con sus respectivos atributos, operaciones y las relaciones. Todo esto se observa en Figura 16. Para mejor visualización se encuentra en el **Anexo 4.** Descripción del Diagrama de Clases.

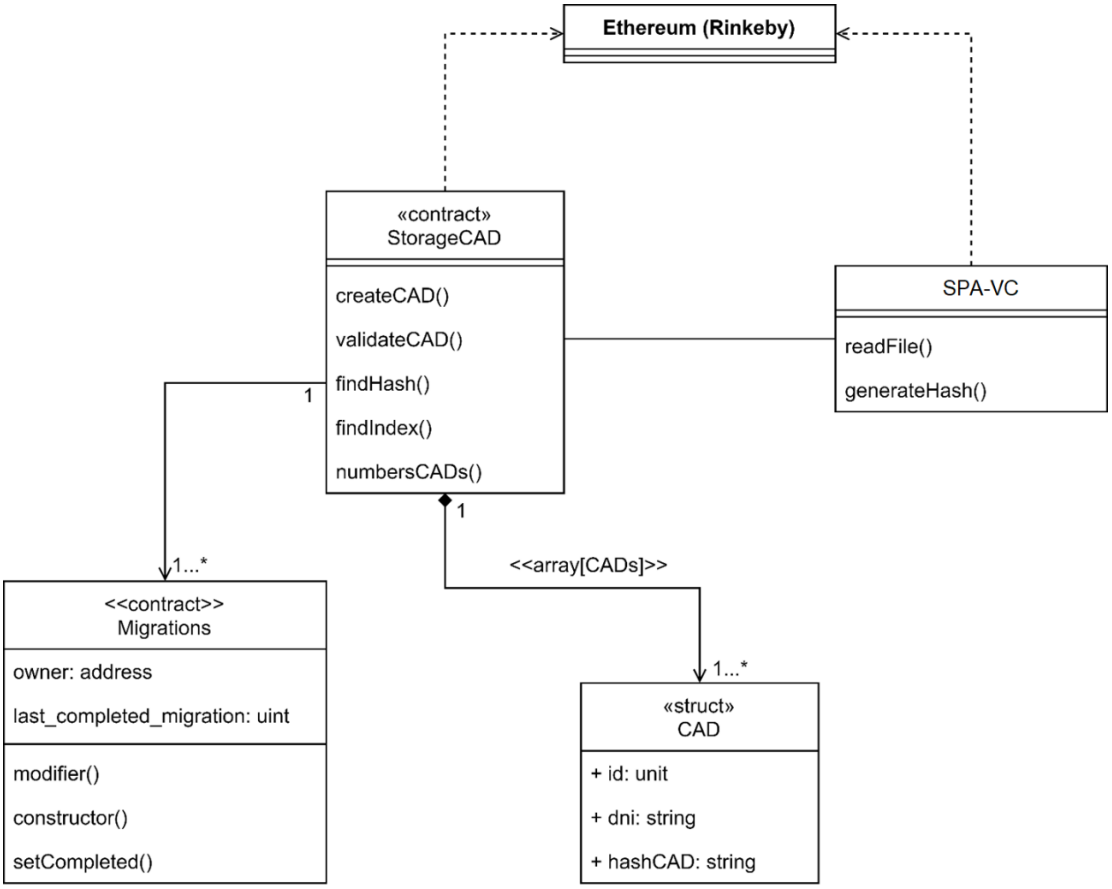


Figura 16. Diagrama de clases del módulo de software.

6.2.6. Fase 6: Codificación y pruebas del subsistema de contratos inteligentes

6.2.6.1. Codificación del subsistema de contratos inteligentes

Los contratos codificados son **Migrations.sol** y **StorageCAD.sol**, el primer contrato viene por defecto y el código fuente de programación de cada uno se detalla en el **Anexo 5.** Código de programación de subsistema de contratos inteligentes. Para el primer contrato inteligente se realizó pseudocódigo para un mejor entendimiento, el cual se observa en la Tabla 16.

Tabla 16.
Pseudocódigo del contrato Migrations.sol

```
INICIO
        owner: CADENA            // Dirección del dueño del despliegue del contrato
        last_completed_migration: ENTERO        // Última migración completada
        msg.sender: CADENA      // Dirección del que requiere realizar la migración
        SI owner = msg.sender ENTONCES
                REALIZAR [migrar el contrato]
                last_completed_migration:= completed
        FIN SI
FIN
```

El contrato inteligente **StorageCAD.sol** se observa el pseudocódigo en la Tabla 17. Tenemos las siguientes funciones:

- La funcionalidad de **registrar** que se denomina **registerCAD()**, se relaciona con el **RF03**, recibe los parámetros del DNI del usuario y el SHA-256 del CAD (_dni y _hashCAD). Determina si el CAD se encuentra registrado usando la función de findHash(), que busca si el SHA-256 ya se encuentra registrado. Además, se verifica para que pueda registrar solo con la cuenta que desplego el contrato inteligente, es decir el propietario. Hay que tener en cuenta que esta funcionalidad de registrar es la única que tiene un costo como transacción.
- La funcionalidad de **validar** denominada **validateCAD()** relacionada con el **RF07**. Recibe los parámetros de dni y SHA-256 del CAD, retornado un valor booleano de true o false, según se encuentre o no registrado el certificado académico digital en la red Blockchain.
- La funcionalidad denominada **numbersCADs()** no recibe ningún parámetro, lo único que retorna es el número de certificados académicos digitales registrados en la Blockchain.

Las siguientes funciones no están contempladas en el TT, pero están realizadas para la escalabilidad del proyecto. Las funcionalidades que se encuentran son:

- **readCAD().** Esta función permite leer la información del CAD.
- **updateCAD().** Esta función permite actualizar la información del CAD

- **deleteCAD().** Esta función no elimina la información del CAD. Lo que realiza es un reseteo de la información, es decir, a las variables les da su valor por defecto. En blockchain no es posible borrar la información una vez ya registrada.

Tabla 17.
Pseudocódigo del contrato StorageCAD.sol

```
INICIO
        owner: CADENA           // Direcció del que requiere realizar la transacción
        msg.sender: CADENA      // Direcció del que requiere realizar la transacción retornado
                                // por Ethereum
        owner:= msg.sender
        CAD: REG        // Estructura del certificado academico digital
                id: ENTERO
                dni: CADENA
                hashCAD: CADENA
        FIN REG

        cads[]: CAD     // Arreglo de CADs

        ** Función registrar un certificado academico digital
        FUNC registerCAD ( _dni: CADENA, _hashCAD: CADENA, _owner: CADENA ) RET: vacío
                registered: BOOLEANO
                registered:= find(_hashCAD)
                SI registered = falso ENTONCES   // Verifica si está registrado
                        SI owner = _owner ENTONCES     // Verifica si es el dueño
                                REALIZAR [registrar certificado]
                        FIN SI
                        DE LO CONTRARIO
                                ESCRIBIR "Intentó registrar con una cuenta diferente"
                        FIN DE LO CONTRARIO
                FIN SI
                DE LO CONTRARIO
                        ESCRIBIR "El certificado academico digital está duplicado"
                FIN DE LO CONTRARIO
        FIN FUNC

        ** Función validar un certificado academico digital
        FUNC validateCAD ( _dni: CADENA, _hashCAD: CADENA ) RET: BOOLEANO
                a: ENTERO
                a:= cads.length
                PARA ( i ;DESDE 1 HASTA a; CON PASO 1 ) HACER
                        SI cads[i].dni = _dni & cads[i].hashCAD = _hashCAD ENTONCES
                                RET verdadero
                        FIN SI
                        DE LO CONTRARIO
                                RET falso
                        FIN DE LO CONTRARIO
                FIN PARA
        FIN FUNC

        ** Función de buscar hash de CAD
        FUNC findHash ( _hashCAD: CADENA ) RET: BOOLEANO
                a: ENTERO
                a:= cads.length
```

```
                PARA ( i ;DESDE 1 HASTA a; CON PASO 1 ) HACER
                        SI cads[i].hashCAD = _hashCAD ENTONCES
                                RET verdadero
                        FIN SI
                        DE LO CONTRARIO
                                RET falso
                        FIN DE LO CONTRARIO
                FIN PARA
        FIN FUNC
FIN
```

6.2.6.2. Pruebas del subsistema de contratos inteligentes

Las **pruebas unitarias** realizadas en este punto permitieron comprobar el correcto funcionamiento del subsistema de contratos inteligentes por unidad de código, asegurando que cada unidad funcione correctamente y eficientemente por separado. Haciendo uso de la herramienta de test que brinda Truffle. Donde el primer paso es **compilar el contrato inteligente**, que en caso de tener un error los mismos no podrían realizar esta acción y sería el primer indicador de errores en el código de los contratos inteligentes. Además, se genera la automatización de 6 pruebas unitarias relacionadas con las funciones del contrato inteligente, que se observan en la Figura 17, cada una con su respectiva aprobación y el tiempo de ejecución unitaria y total.

Para mayor detalle observar el **Anexo 6.** Plan de Pruebas Unitarias del subsistema de contratos inteligentes.

```
$ truffle test
Using network 'test'.


Compiling your contracts...
===========================
> Compiling .\contracts\Migrations.sol
> Compiling .\contracts\StorageCAD.sol
> Compiling .\contracts\StorageCAD.sol
> Compiling .\test\TestStorageCAD.sol
> Artifacts written to C:\Users\ASUS\AppData\Local\Temp\test--14096-krYThq4732t4
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang



  TestStorageCAD
    √ testItRegisterCAD (507ms)
    √ testItValidateCADAuthentic (317ms)
    √ testItValidateCADNotAuthentic (305ms)
    √ testItFindHashCADRegistered (345ms)
    √ testItFindHashCADNotRegistered (261ms)
    √ testItNumbersCADs (234ms)


  6 passing (14s)
```

Figura 17. Resultado de las pruebas unitarias del subsistema de contratos inteligentes.

6.2.7. Fase 7: Diseño del subsistema de aplicaciones

En esta fase se relaciona al patrón de diseño Modelo-Vista-Controlador (MVC) con la DApp, en el cual se detalla cada punto a continuación y se puede observar representado en la Figura 18.

- **Modelo:** estaría compuesto por la estructura de datos que se lo relacionaría con el contrato inteligente en la Blockchain, el mismo se almacena en el bloque génesis de la cadena.
- **Controlador:** es el encargado de actualizar el modelo, por lo que se lo relaciona con los métodos que se tienen en el contrato inteligente, específicamente en la interfaz ABI, generados a partir de la compilación y migración del contrato.
- **Vista:** es todo el front-end desarrollado bajo la forma de desarrollo de una **SPA (Single Page Application – Aplicaciones de una sola página,** para mayor detalle consultar el punto **4.10 de la sección Del Marco teórico),** compuesta por las vistas de registrar y validar.

Adicionalmente, la librería web3.js específicamente con el módulo web3.eth, es el paquete que permite interactuar con la Blockchain de Ethereum y los contratos inteligentes. Es decir, es el que permite llamar a los métodos de la interfaz ABI del contrato inteligente, así que web3.eth se puede establecer como la plomería entre la vista y el controlador/modelo.

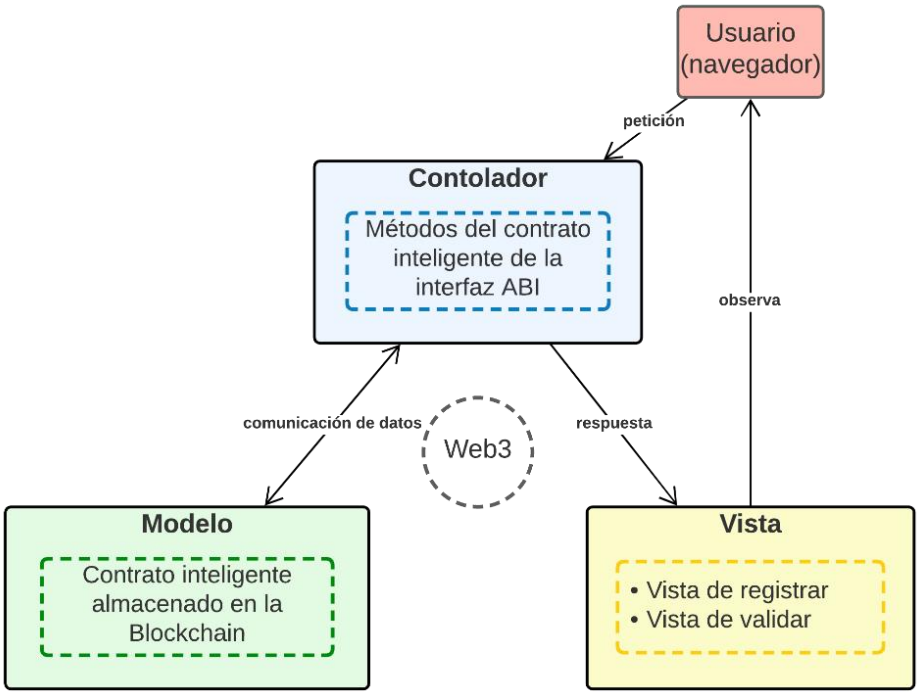


Figura 18. Modelo-Vista-Controlador y relación con una DApp en Blockchain.

Así mismo, aunque no fue factible realizar un diseño de un diagrama de clases, debido a que el subsistema de aplicaciones está compuesto solo por el front-end y el back-end para mejor

comprensión se lo puede relacionar con el subsistema de contratos inteligentes, aunque hay que aclarar que el subsistema de aplicaciones puede tener su propio back-end en las tecnologías deseadas. El subsistema de aplicaciones se ve representado como la clase denominada **SPA-VC (Single Page Application – Validate Certificates)** en el punto **6.2.5** (Fase 5: Diseño del subsistema de contratos inteligentes), la Figura 16.

Eventualmente, se generó la **arquitectura-vista física** que abarca al subsistema de aplicaciones que modela la disposición física de los nodos y componentes del módulo de software, todo esto se observa en la Figura 19.

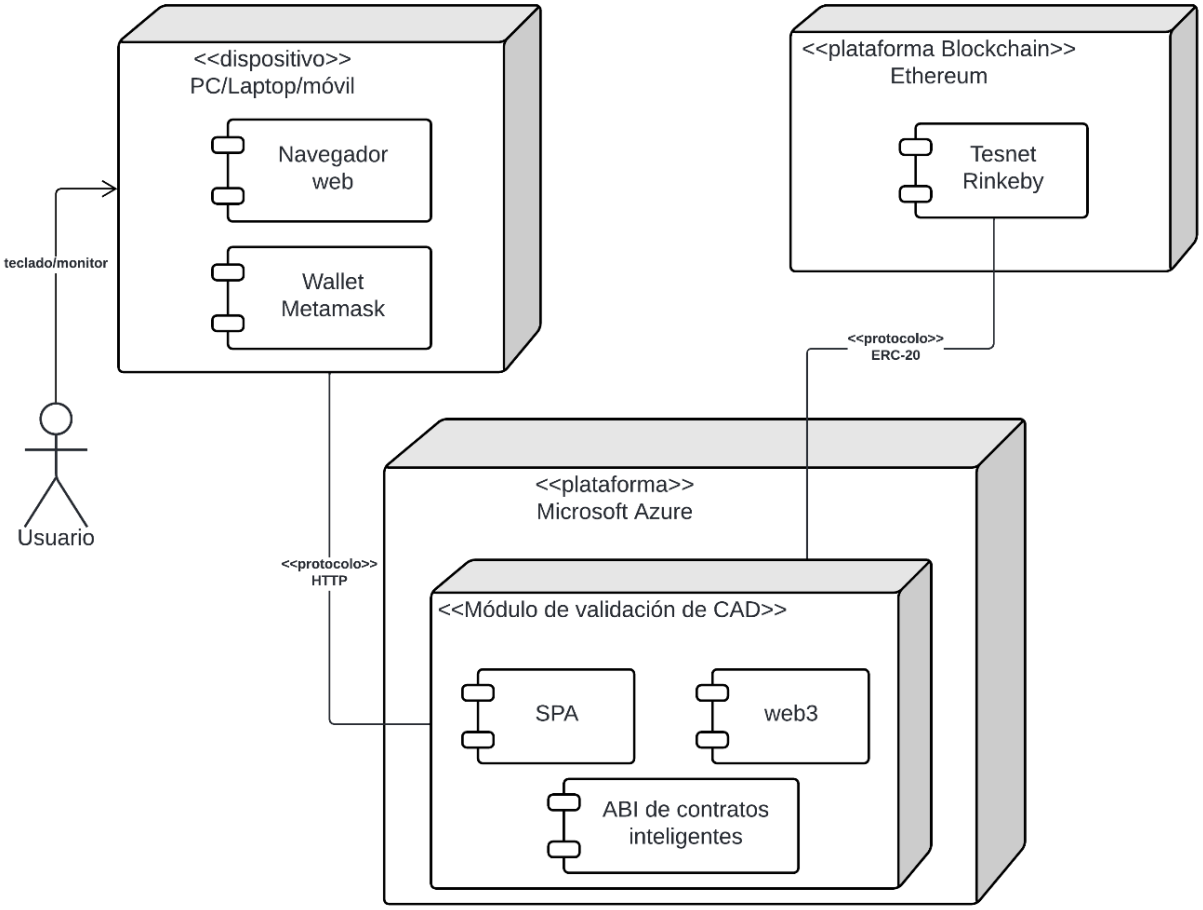


Figura 19. Arquitectura-vista física del módulo de software.

6.2.8. Fase 8: Codificación y prueba del subsistema de aplicaciones

6.2.8.1. Codificación del subsistema de aplicaciones

La codificación del subsistema de aplicaciones esta desarrollada bajo la biblioteca Javascript denominada React. Por lo que se detallara de manera puntual el archivo **App.js** que es el archivo en donde se ubica la interacción con el contrato inteligente y la interfaz que visualiza el usuario final. Para mayor detalle se puede observar el código fuente de programación en el **Anexo 7.** Código de programación de subsistema de aplicaciones.

Las principales importaciones de componentes o librerías son la declaración de los contratos inteligentes migrados y las librerías: web3, SHA-256 y sweetalert2, entre lo a destacar.

Los estados de las variables necesarias y que se manejan para obtener los valores como: si se encuentra conectado a web3, la extensión del Metamask, las cuentas y la dirección del contrato inteligente, se determina si la cuenta conectada es el propietario del despliegue del contrato inteligente, entre lo principal, todo esto se observa en el pseudocódigo de la Tabla 18.

Tabla 18.
Pseudocódigo estados y conexión con web3, metamask, contrato y cuentas

```
INICIO
        web3:BOOLEANO                   // Estado de conexión de web3
        accounts[0]: CADENA             // Cuenta de la wallet
        deployementOwner: CADENA        // Cuenta del dueño del contrato

        SI web3 = verdadero ENTONCES
                isOwner()
        FIN SI
        DE LO CONTARIO.
                ESCRIBIR "No se conectó con web3, su wallet o la red Blockchain"
        FIN DE LO CONTRARIO

        ** Verificar si es el dueño del contrato
        FUN isOwner( _accounts[0]: CADENA, _deployementOwner: CADENA ) RET: BOOLEANO
                isDeployementOwner: BOOLEANO
                SI _accounts[0] = _deployementOwner ENTONCES
                        isDeployementOwner:= verdadero
                        RET isDeployementOwner
                FIN SI
                DE LO CONTARIO.
                        ideployementOwner:= falso
                        RET isDeployementOwner
                FIN DE LO CONTRARIO
                REALIZAR run() // La función a la que se llama es la que se inicia luego de verificar
                        // los estados y la conexión
        FIN FUNC
FIN
```

La función **handleSubmit()** se observa en la Tabla 19 llama a la función del contrato para registrar el CAD, siempre verificando que los campos no deben estar vacíos y que no se encuentre ya registrado el CAD. Así mismo se actualiza en pantalla el número de registros de CADs, relacionado con el **RF02** y **RF03.**

La funcionalidad de validar un CAD se denomina **handleValidate()**, relacionado con el **RF04, RF05, RF06** y **RF07**. Cambiando el estado de la variable validate, mostrando un resultando u otro. Se observa el pseudocódigo en la Tabla 19.

Tabla 19.
Pseudocódigo de la función de registrar y validar un CAD

```
INICIO
        storageDNI: CADENA
        storageHashCAD: CADENA
        registered:= contract.findHash(storageHashCAD)

        ** Función registrar un certificado academico digital
        FUNC handleSubmit( ) RET: vacío
                SI storageDNI = vacío || storageDNI = vacío ENTONCES
                        ESCRIBIR "Los campos no pueden estar vacíos"
                FIN SI
                DE LO CONTARIO
                        SI registered = false ENTONCES
                                REALIZAR [registrar certificado]
                        FIN SI
                        DE LO CONTRARIO
                                ESCRIBIR "El certificado academico digital ya está registrado"
                        FIN DE LO CONTRARIO
                FIN DE LO CONTRARIO
        FIN FUNC

        ** Función validar un certificado academico digital
        FUNC handleValidate( ) RET: vacío
                SI storageDNI = vacío || storageDNI = vacío ENTONCES
                        ESCRIBIR "Los campos no pueden estar vacíos"
                FIN SI
                DE LO CONTARIO
                        response: BOOLEANO
                        response:= contract.validateCAD(storageDNI, storageHashCAD)
                        SI response = verdadero ENTONCES
                                ESCRIBIR "El certificado academico digital es autentico"
                        FIN SI
                        DE LO CONTRARIO
                                ESCRIBIR "El certificado academico digital no es autentico"
                        FIN DE LO CONTRARIO
                FIN DE LO CONTRARIO
        FIN FUNC
FIN
```

La función **readFile()** recibe el documento solo en formato pdf, lo convierte en formato Binary String para evitar conflicto con caracteres especiales para pasar a generar el SHA-256, relacionado al **RF01** y **RF05**. Una vez obtenido modifica el estado de la variable denominado storageHashCAD. Todo esto se muestra en el pseudocódigo la Tabla 20.

Tabla 20.
Pseudocódigo de la función de leer el CAD y generar SHA-256

```
INICIO
        file: FILE
        file_reader: CADENA
        hash256File: CADENA
        storageHashCAD: CADENA
```

```
        ** Función de leer CAD
        FUNC readFile( ) RET: vacío
                LEER file
                file_reader:= fileReader(file)
                hash256File:= readAsBinary(file_reader)
                storageHashCAD:= sha256(hash256File)
        FIN FUNC

        ** Función generar SHA-256
        FUNC sha256( _hash256File: CADENA ) RET: CADENA
                REALIZAR [Llamar a la función sha-256 de la librería importada]
                RET _hash256File
        FIN FUNC
FIN
```

6.2.8.2. Pruebas del subsistema de aplicaciones

Las **Pruebas Unitarias** generadas para el subsistema de aplicaciones, aunque este embarque en su totalidad front-end. Actualmente es procedente realizar estas pruebas debido a que el front-end contiene más carga lógica a comparación de años atrás. Los resultados se observan en la Figura 20. Además, se procedió a generar la automatización de 13 pruebas unitarias en la que se observa el estado de aceptación con el tiempo de ejecución de cada una y el total. Las mismas fueron realizadas con las librerías de **Jest** y **Enzyme**.

Para mayor detalle se puede revisar el **Anexo 8.** Plan de Pruebas Unitarias del subsistema de aplicaciones. Teniendo la misma finalidad que en el subsistema de contratos inteligentes que es la de comprobar que las diferentes partes del código funcione correctamente.

```
PASS  src/App.test.js (6.757s)
  Pruebas unitarias a: <App />
    √ Renderizar <App /> al no estar conectado a web3, cuentas, contrato o metamask  (70ms)
    √ Renderizar <App /> correctamente al estar conectado correctamente (45ms)
    √ Mostrar formulario para validar un CDA (67ms)
    √ Mostrar formulario para registrar un CDA (75ms)
    √ Mostrar el mensaje adecuado cuando el CDA es autentico (53ms)
    √ Mostrar el mensaje adecuado cuando el CDA no es autentico (70ms)
    √ Llamar a componentDidMount() (103ms)
    √ Llamar a la función run() (80ms)
    √ Llamar a la función handleSubmit() (59ms)
    √ Llamar a la función handleValidate() (46ms)
    √ Llamar a la función handleChangeValidate() (36ms)
    √ Llamar a la función handleChangeRegister() (40ms)
    √ Llamar a la función readFile() (64ms)

Test Suites: 1 passed, 1 total
Tests:       13 passed, 13 total
Snapshots:   2 passed, 2 total
Time:        9.073s
Ran all test suites related to changed files.
```

Figura 20. Resultado de las pruebas unitarias del subsistema de aplicaciones.

6.3. Objetivo 3

Probar la solución en una testnet de blockchain para el módulo de software de validación de certificados académicos digitales.

En el presente objetivo interviene la siguiente fase (Figura 21):

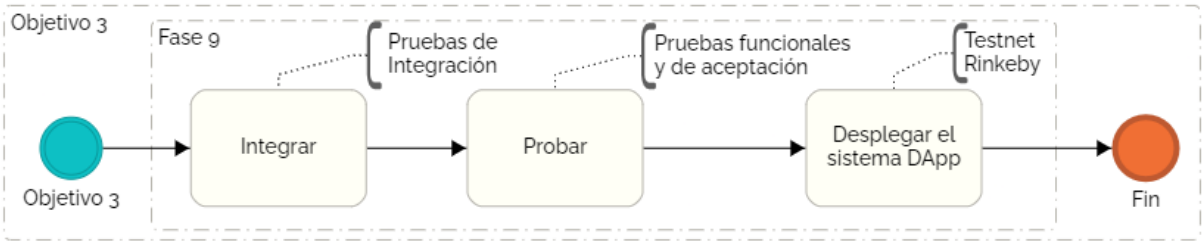


Figura 21. Proceso de desarrollo para el objetivo 3.

En la Figura 21 la fase 9 que se encuentra se denominada **Integrar, probar y desplegar el sistema DApp**, para mayor detalle se encuentran el punto 9 de la sección **ABCDE (agile block chain DApp engineering)** del Marco teórico.

6.3.1. Fase 9: Integrar, probar y desplegar el sistema DApp (módulo de software)

6.3.1.1. Integrar el sistema DApp (módulo de software)

En este punto se realizó las **Pruebas de Integración** que se complementan a las pruebas unitarias, comprueban que todos los elementos del módulo de software funcionen correctamente de manera conjunta, aquí se valida que los subsistemas se integren de manera correcta. La Tabla 21 muestra los casos de pruebas planificados y ejecutados, para ejecutar las pruebas se usó el framework de testing denominado **Cypress**, el resultado se observa en la Figura 22. Permitiendo analizar paso a paso la ejecución completa de las pruebas automatizadas con el tiempo total de ejecución y la aceptación de cada prueba.

Tabla 21.
Resultados de las Pruebas de Integración

Nro. Caso de Prueba	Componentes	Descripción de lo que se probará	¿OK?	Observación
CP01	Subsistema de contratos inteligentes – Subsistema de aplicaciones	Se registre correctamente el DNI y el SHA-256 del CAD en la testnet de Blockchain	✓	N/A
CP02	Subsistema de contratos inteligentes – Subsistema de aplicaciones	Se valide el DNI y EL SHA-256 del CAD, si se encuentra en la testnet de Blockchain	✓	N/A
CP03	Wallet - Subsistema de aplicaciones	Conexión con la wallet de Metamask	✓	N/A
CP04	Subsistema de aplicaciones - Blockchain	Al hacer clic en el botón de VALIDAR, se comunique con la blockchain	✓	N/A
CP05	Subsistema de aplicaciones - Blockchain	Al hacer clic en el botón de REGISTRAR, se comunique con la blockchain	✓	N/A

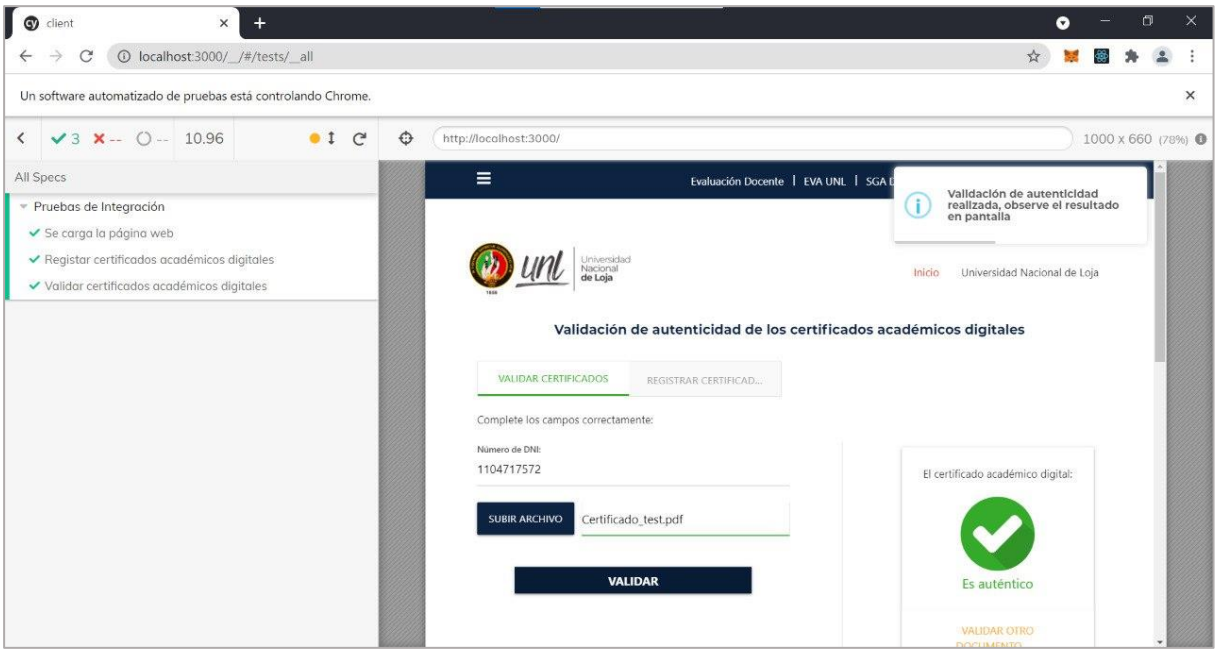


Figura 22. Resultado de las Pruebas de Integración del módulo de software.

Para observar el documento completo del plan de pruebas de integración ejecutadas para el módulo de software, revisar el **Anexo 9.** Plan de Pruebas de Integración.

6.3.1.2. Probar el sistema DApp (módulo de software)

Las pruebas realizadas en este punto son las **Pruebas Funcionales** (**Anexo 10.** Plan de Pruebas Funcionales) y las **Pruebas de Aceptación** (**Anexo 11.** Plan de Pruebas de Aceptación) del módulo de software.

El resumen de los casos de pruebas realizado en el Plan de Pruebas Funcionales se observa en la Tabla 22. Todo esto con el fin de verificar que el módulo de software satisface los requisitos especificados con las funciones establecidas inicialmente.

Tabla 22.
Casos de pruebas del plan de Pruebas Funcionales

Nro. Caso de prueba	Descripción del caso de prueba	RF relacionados	Resultado	Observación
CP01	Se probará la respuesta del módulo de software al registrar un DNI con un certificado académico digital en formato PDF	RF01 RF02 RF03 RF08	SI	
CP02	Se probará la respuesta del módulo de software al registrar, pero con los campos vacíos.	RF01 RF02 RF03 RF08	SI	
CP03	Se probará la respuesta del módulo de software al registrar un certificado	RF01 RF02 RF03	Parcialmente	Se corrigió debido a que usaba la función validateCAD() y la

	académico digital ya registrado con anterioridad	RF08		correcta es findHash(), del contrato inteligente
CP04	Se probará la respuesta del módulo de software al validar el DNI correcto con un certificado académico digital auténtico en formato PDF	RF04 RF05 RF06 RF07 RF08	SI	
CP05	Se probará la respuesta del módulo de software al validar el DNI correcto con una copia del certificado académico digital auténtico en formato PDF.	RF04 RF05 RF06 RF07 RF08	SI	
CP06	Se probará la respuesta del módulo de software al validar el DNI incorrecto con un certificado académico digital auténtico en formato PDF	RF04 RF05 RF06 RF07 RF08	SI	
CP07	Se probará la respuesta del módulo de software al validar el DNI correcto con un certificado académico digital editado en formato PDF	RF04 RF05 RF06 RF07 RF08	SI	
CP08	Se probará la respuesta del módulo de software al validar el DNI correcto con un certificado académico digital diferente al registrado en formato PDF.	RF04 RF05 RF06 RF07 RF08	SI	
CP09	Se probará la respuesta del módulo de software al validar, pero con los campos vacíos.	RF04 RF05 RF06 RF07 RF08	SI	

En el caso de las **Pruebas de Aceptación** tienen el fin de verificar la aceptación del producto, se aplicó una encuesta luego de la presentación y manipulación del de software módulo por parte del estudiante, aplicando los siguientes parámetros que su resumen se observa en la Tabla 23. En los diferentes parámetros se observa un porcentaje de aceptación muy alto superando incluso el 86% en la mayoría. Esto se realizó en una muestra de 62 estudiantes pertenecientes a la carrera de Ingeniería en Sistemas/Computación de la Universidad Nacional de Loja. Adicionalmente a estos parámetros se solicitó en la encuesta aplicada sus nombres y apellidos; cédula y registrar automáticamente el correo electrónico para la veracidad de los resultados.

Tabla 23.
Resumen de Pruebas Funcionales

No.	Parámetros	Porcentaje (%)		
		Sí	Parcialmente	No
1	¿Es simple el vocabulario utilizado?	72,6	27,4	0
2	¿Se proporciona tiempo suficiente para realizar las entradas por teclado?	91,9	6,5	1,6
3	¿Se entienden la interfaz y su contenido?	95,2	4,8	0
4	¿Resulta fácil identificar un objeto o una acción?	79	19,4	1,6
5	¿Resulta fácil entender el resultado de una acción?	83,9	16,1	0
6	¿Está diseñada la interfaz para facilitar la realización eficiente de las tareas de la mejor forma posible?	88,7	11,3	0
7	¿Son apropiados los mensajes presentados por el sistema?	85,5	14,5	0
8	¿Actúa el sistema en la prevención de errores?	88,7	8,1	3,2
9	¿El sistema informa claramente sobre los errores presentados?	88,7	11,3	0
10	¿Permite una cómoda navegación dentro del producto y una fácil salida de éste?	96,8	3,2	0
11	¿Se presenta al usuario la información que sólo necesita?	83,9	16,1	0
	PROMEDIO	86,8	12,60	0,6

En el parámetro número ocho emitió un resultado negativo, debido a que un participante de la encuesta encontró un error al cancelar la transacción en la wallet, situación que no se había tomado en cuenta hasta ese momento, que se procedió a corregir oportunamente.

Además, como se observa en la Tabla 23 el resultado del criterio sí, parcialmente y no con un: 86,8; 12,60 y 0,6 por ciento respectivamente. Encadenando lo mencionado con el criterio del estado de las pruebas de la Tabla 24, se determine que las Pruebas de Aceptación tienen un estado **positivo**.

Tabla 24.
Criterio para el estado de las Pruebas de Aceptación

Estado de las Pruebas de Aceptación	Criterio		
	Sí	Parcialmente	No
Positivo	>= 80%	<= 20%	<1%
Negativo	< 80%	> 20%	>1%

6.3.1.3. Desplegar el sistema DApp (módulo de software)

En este punto se procedió a desplegar en la testnet **Rinkeby**, con base a lo establecido en el objetivo 3. Para proceder a realizar el despliegue de la testnet Rinkeby se usó un servicio de **Infura**, que se puede conocer como nodo como servicio o infraestructura como servicio (IaaS) para crear el proyecto denominado “CAD-UNL” y así se obtuvo las credenciales para la conexión a la testnet, como se observa en la Figura 23.

Realizado lo anterior, se procede a compilar (Figura 24) y migrar (Figura 25) los contratos inteligentes. En la Figura 26 se observa el costo total de la transacción del despliegue de los contratos inteligentes en la testnet Rinkeby, que es 0.001488989016047929 ETH que equivale a 4.68 dólares estadounidenses. Hay que tener en cuenta que el valor en dólares varía de acuerdo al a la tasa de conversión ETH/USD. Eventualmente por medio de la herramienta de **Etherscan** se puede dar trazabilidad a las diferentes transferencias con el contrato inteligente como se observa en la Figura 27.

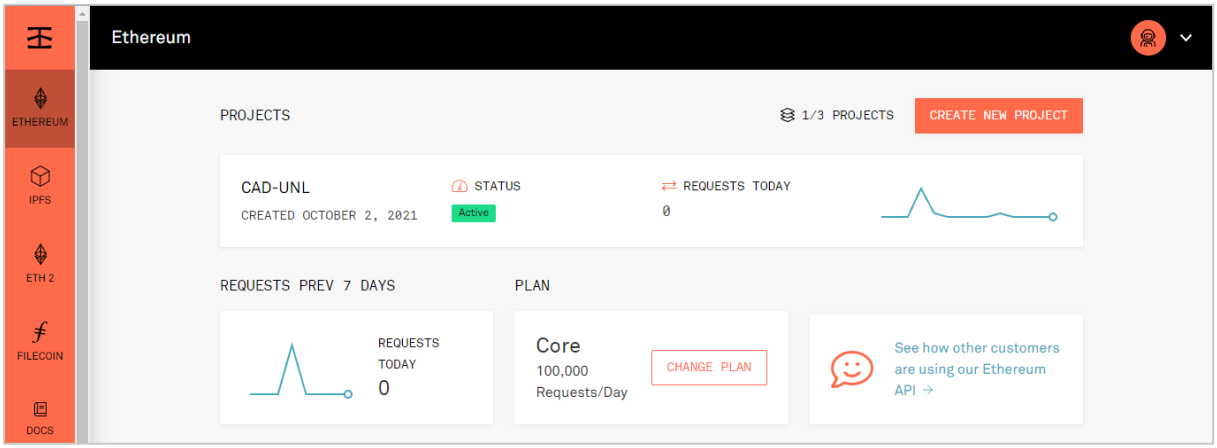


Figura 23. Creación del proyecto en Infura para conexión a testnet de Rinkeby.

```
truffle(develop)> compile

Compiling your contracts...
===========================
> Compiling .\contracts\Migrations.sol
> Compiling .\contracts\StorageCAD.sol
> Artifacts written to C:\Users\ASUS\Documents\PATO\Tesis\M-dulo-de-software-de-VCDA-por-Blockchain\client\src\
contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang
```

Figura 24. Compilación de los contratos inteligentes.

```
truffle(develop)> migrate --network rinkeby

Compiling your contracts...
===========================
> Compiling .\contracts\Migrations.sol
> Compiling .\contracts\StorageCAD.sol
> Artifacts written to C:\Users\ASUS\Documents\PATO\Tesis\M-dulo-de-software-de-VCDA-por-Blockchain\client\src\
contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang



Migrations dry-run (simulation)
===============================
> Network name:    'rinkeby-fork'
> Network id:      4
> Block gas limit: 30000000 (0x1c9c380)
```

Figura 25. Migración de los contratos en la testnet Rinkeby.

```
Summary
=======
> Total deployments:   2
> Final cost:          0.001488989016047929 ETH
```

Figura 26. Costo de transacción de la migración en ETH.

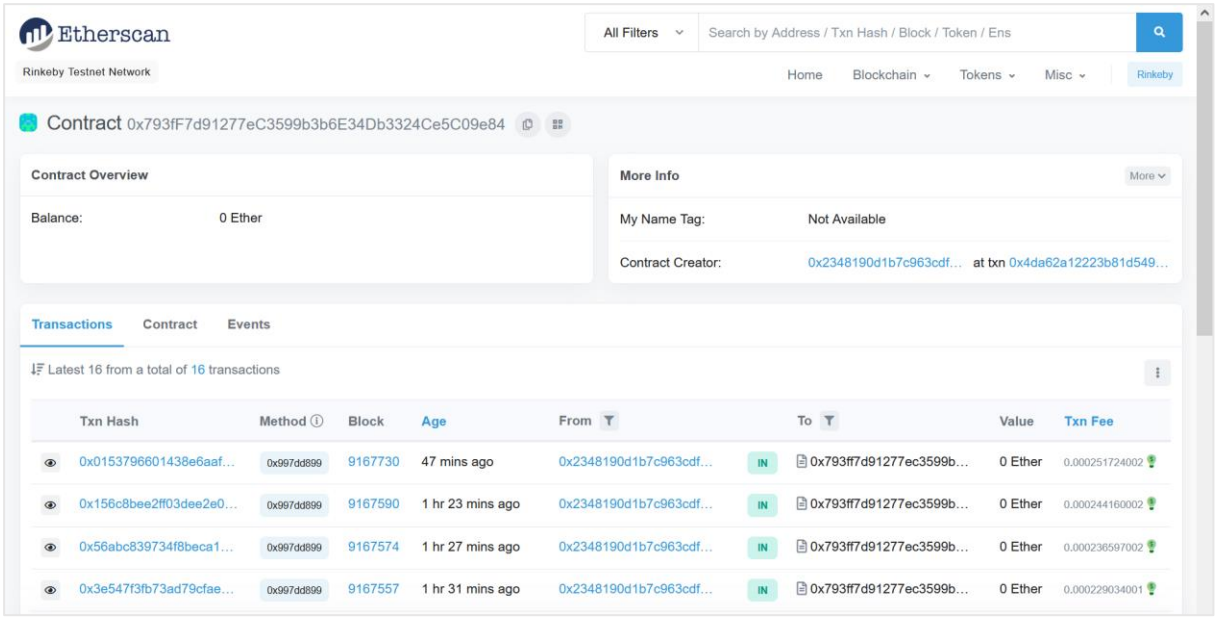


Figura 27. Trazabilidad de las transacciones en Etherscan.

Adicionalmente se desplegó en el servicio de computación en la nube **Microsoft Azure**, como se observa en la Figura 28, en donde se generó un proyecto denominado certificados-UNL y el resultado del mismo se puede observar en la siguiente dirección web: https://certificados-unl.azurewebsites.net/.

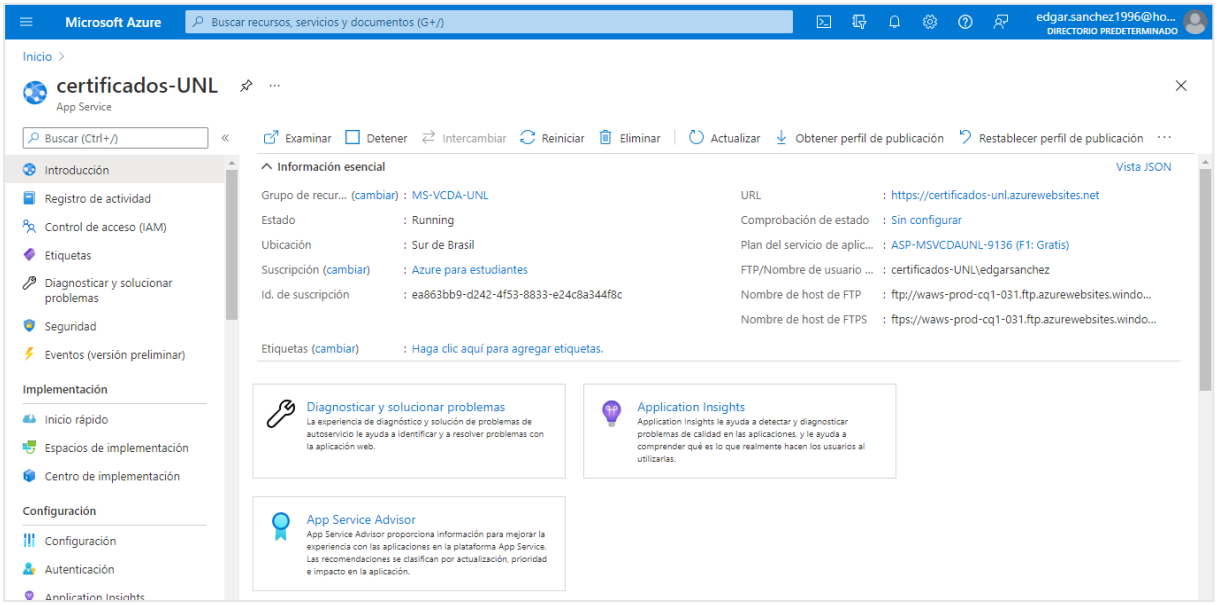


Figura 28. Proyecto generado en Microsoft Azure.

Igualmente, en la Figura 29 se procedió a generar un diagrama en donde se representa modelo cliente servidor, relacionando las tecnologías empleadas y mencionadas anteriormente. En el cual, el cliente demanda los servicios de registrar o validar un certificado académico digital a la aplicación alojada en el servicio de la nube Microsoft Azure, específicamente en el servicio App Services, que se determina Plataforma como servicio (PaaS). Eventualmente, la aplicación por medio de la wallet Metamask e Infura (nodo de

acceso a Ethereum-IaaS), realiza peticiones a la testnet Rinkeby de la Blockchain de Ethereum.

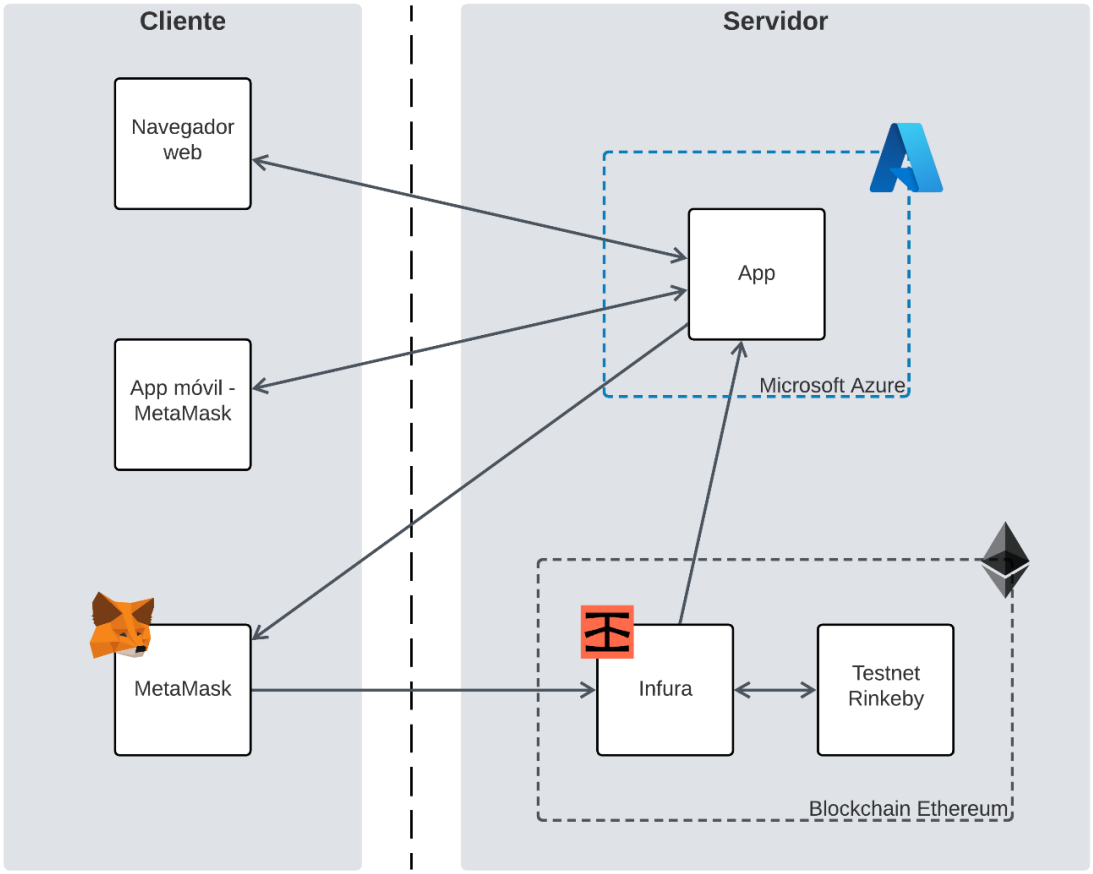


Figura 29. Diagrama cliente-servidor de la solución.

El costo de registro de un certificado académico digital, ya que es la única funcionalidad con un costo. En la Tabla 25 se puede observar los valores de un gas máximo de 0.000268 ETH ($ 0.84) que es el valor máximo que se podría pagar por registrar un certificado, pero el valor final pagado es de 0.000179 ETH ($ 0.56), todo esto se constata en la Figura 30. Hay que tener en cuenta que este costo puede variar dependiendo del gasto computacional que se requiera e incluso el tráfico en red que exista, pero siempre gira en torno a los valores mencionados. Así mismo, el valor en dólares varía de acuerdo al a la tasa de conversión ETH/USD.

Tabla 25.
Costo de registro de un certificado académico digital

Gas máximo a usarse (ETH)	Costo (ETHER)	Costo (dólares estadounidenses)
0.000268	0.000179	0.56
		Varía según la tasa de conversión ETH/USD

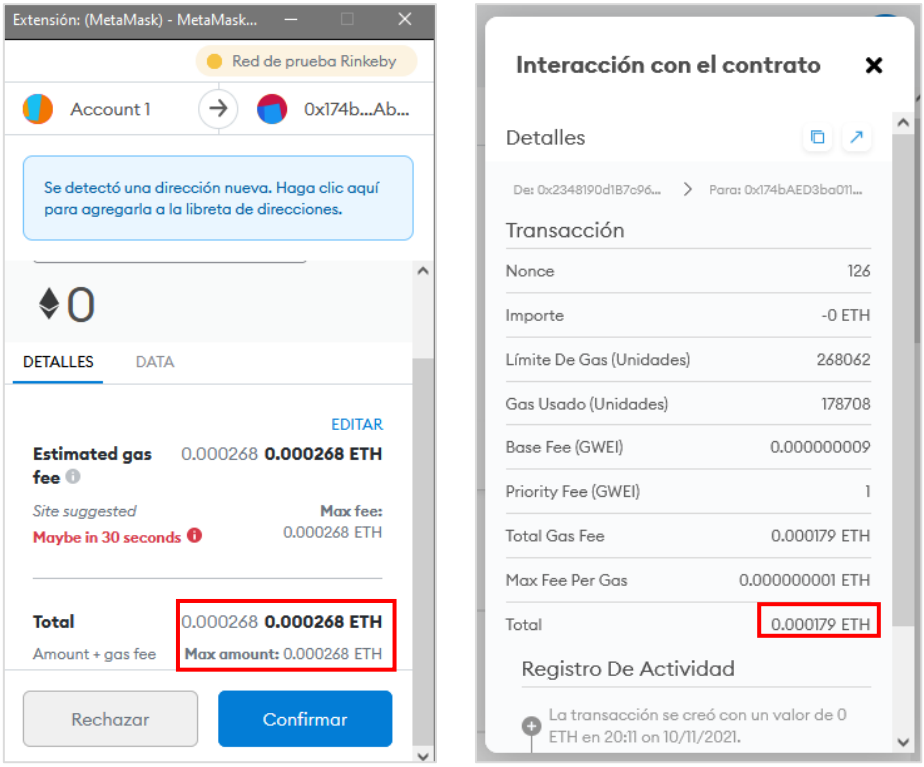


Figura 30. Costo máximo de gas y final en ETH al registrar un certificado académico digital en la testnet Rinkeby.

Finalmente se generó dos manuales que son:

- **Manual de programador** que se observa en la siguiente dirección web: https://github.com/EdgarPatricio/M-dulo-de-software-de-VCDA-por-Blockchain (README.md)
- **Manual de usuario** que se ubica en el **Anexo 12.** Manual de Usuario.

7. Discusión

7.1. Desarrollo de la propuesta alternativa

El Proyecto de Trabajo de Titulación denominado “Implementación de la tecnología Blockchain para la validación de autenticidad de los certificados académicos digitales”, busca implementar una solución informática en base a la formación académica. Se desarrolló con el fin de cumplir los objetivos planteados.

7.1.1. Objetivo 1: Definir el módulo de software para la validación de certificados académicos digitales usando la Ingeniería de Requisitos.

Para cumplir el objetivo se analizó la metodología de desarrollo de software a usarse, inicialmente se pretendió usar DevOps basados en la recomendación de Truffle Suite, pero generaba el conflicto de que las fases y entregables no son claras, así mismo de que no es considerada por muchos como una metodología. Además, las metodologías conocidas hasta el momento por el director y el estudiante se enfocaban en sistemas tradicionales y no en sistemas que usen tecnología Blockchain, debido a que la misma es emergente. Eventualmente durante el proceso con base a investigación y tutorías de expertos en la tecnología Blockchain, se encontró la metodología de desarrollo de software ABCDE (ver el punto **5.1 Metodología de desarrollo de software**), teniendo en cuenta que es la única metodología conocida en su momento que toma en cuenta a las DApp y los contratos inteligentes para poder diseñarlos, codificarlos y realizar pruebas. La metodología seleccionada permitió que el módulo de software tenga una base sólida que le permite ejecutarse de manera correcta.

Las 4 primeras fases de la metodología de desarrollo de software ABCDE fueron completadas con base al documento de Especificación de Requisitos de Software. En donde se procedió a reemplazar la fase 3 historias de usuario por requerimientos, debido a que era más factible y no genera conflictos en fases futuras, aunque de igual manera se deja en evidencia la realización de las mismas (ver **Anexo 2.** Descripción de Historias de usuario). Además, de plantear un diagrama de la arquitectura (ver el punto **6.1.4 de la sección de Resultados**) y poder definir los dos subsistemas del módulo de software. Los resultados obtenidos en este punto fueron enviados a la Tercera Conferencia Internacional del IEEE sobre Computación y Aplicaciones de Blockchain (BCCA 2021), se observa en el **Anexo 3.** Paper BCCA 2021.

7.1.2. Objetivo 2: Desarrollar el módulo de software para la validación de certificados académicos digitales por medio de la plataforma Blockchain Ethereum Virtual Machine.

En este objetivo se tenía dividido al módulo de software en el subsistema: contratos inteligentes y de aplicaciones, en los cuales se procedió a diseñar, codificar y ejecutar pruebas

unitarias (correspondientes a las fases 5 a 8 de la metodología de desarrollo de software ABCDE). La estructura general del proyecto (ver el punto **6.2.4 de la sección de Resultados**) se generó en base a las tecnologías de Node js y React js, justamente del box de React ofrecido por Truffle Suite. Aunque existen más box o paquetes dentro de la plataforma de Truffle Suite con diferentes tecnologías, el seleccionado es el más favorable, luego de la capacitación adquirida hasta ese momento y que las tecnologías contenidas eran más factibles para poder cumplir el objetivo.

En el subsistema de contratos inteligentes, la fase de diseño se completó por medio de un diagrama de clases (ver el punto **6.2.5 de la sección de Resultados**), la metodología de desarrollo de software informa de que debe usar elementos diferenciadores para las clases que sean de tipo contratos inteligentes. La fase de codificación se realizó bajo el lenguaje de programación denominado Solidity (ver **Anexo 5.** Código de programación de subsistema de contratos inteligentes), un lenguaje de programación en el que es importante definir el tamaño adecuado de las variables por cuestiones de costo a futuro en las transacciones. La fase de Pruebas Unitarias (ver **Anexo 6.** Plan de Pruebas Unitarias del subsistema de contratos inteligentes) se ejecutaron de manera correcta por medio de la herramienta que brinda Truffle, funcionalidad que facilita el realizarlas, codificarlas y ejecutarlas en tiempo real cada unidad de código de los contratos inteligentes.

El subsistema de aplicaciones en la fase de diseño se realizó la arquitectura/vista física y una relación con el MVC (ver el punto **6.2.7 de la sección de Resultados**), debido a que se desarrolló bajo la forma de una SPA (Single Page Application). La fase de codificación se desarrolló por medio de React js (ver **Anexo 7.** Código de programación de subsistema de aplicaciones), además fue necesario la instalación de MetaMask como extensión en el navegador y generar una cuenta en la wallet (cartera) de la misma. Es preciso indicar que la cartera debe estar conectada a la testnet local o sino emitirá errores al iniciar el sitio web. En la fase de Pruebas Unitarias (ver **Anexo 8.** Plan de Pruebas Unitarias del subsistema de aplicaciones) se realizó con tecnologías de Jest y Enzyme, esta última fue necesaria debido a que se usa la extensión de MetaMask en el navegador web.

Para completar cada subsistema del módulo de software se los realizó en sprints (ver el punto **6.2.1** Iteraciones (sprints) del módulo de software), indicados dentro de la metodología de desarrollo de software ABCDE, con el fin de proceder a la última fase que está contemplada en el siguiente objetivo.

7.1.3. Objetivo 3: Probar la solución en una testnet de blockchain para el módulo de software de validación de certificados académicos digitales

Para cumplir este objetivo se procedió ejecutar la fase 9 de la metodología de desarrollo de software ABCDE. Aquí se procedió a identificar la testnet blockchain Rinkeby en la que se realizó el despliegue, se usó la herramienta de Infura para obtener las credenciales de conexión como nodo de Ethereum. Se realizaron Pruebas de Integración (ver **Anexo 9.** Plan de Pruebas de Integración) para determinar la integración de los dos subsistemas por medio de la herramienta Cypress arrojando resultados positivos, el usar esta herramienta permitió simular el uso de la aplicación en tiempo real (ver el punto **6.3.1.1 de la sección de Resultados**).

Las Pruebas Funcionales (ver **Anexo 10.** Plan de Pruebas Funcionales) permitió determinar que los requisitos funcionales planteados se cumplieron a cabalidad, probando la validación en diversas situaciones, generando resultados positivos en la autenticidad de los certificados digitales académicos. Además, en las Pruebas de Aceptación (ver **Anexo 11.** Plan de Pruebas de Aceptación), donde se pudo obtener información por partes de los estudiantes de la carrera de Ingeniería en Sistemas/Computación para mejoras que algunas se observan en el punto de trabajos futuros de las conclusiones y en base con los criterios de las pruebas se dan como positivo (ver el punto **6.3.1.2 de la sección de Resultados**).

Además, se desplegó el módulo de software en el servicio de computación de Microsoft Azure (ver el punto **6.3.1.3 de la sección de Resultados**), debido al convenio de la universidad con esta entidad. Aunque está contemplado para ser usado solo en el navegador web con la wallet de Metamask instalada como extensión, una vez desplegado se identificó que se puede usar en la aplicación móvil “Metamask” disponible para iOS y Android, en la sección de navegador web a través de la dirección web del despliegue.

Eventualmente, el costo de la transferencia al registrar un certificado académico digital en la testnet Rinkeby de Ethereum, varía entre 0.00016 a 0.0002 ether que equivale a 0.50 y 0.63 dólares respectivamente, hay que tener en cuenta que el valor en dólares varía de acuerdo al a la tasa de conversión ETH/USD. Y actualmente para la certificación de documentos en una notaría según el Consejo de Judicatura, en el Reglamento del sistema notarial integral de la función judicial (art. 74), se debe pagar por foja (dos lados) el costo de $ 1.79 [38]. Por lo tanto, el módulo de software del TT genera un costo beneficio óptimo, ya que una vez registrado el certificado académico puede ser validado el número de veces necesario y no una sola vez como es al tener una copia en físico por parte de una notaría. En vista a que la validación no tiene un costo en la red Blockchain y se puede hacer el número de veces requerido, derivando en la sostenibilidad del proyecto.

8. Conclusiones

Una vez culminado el Trabajo de Titulación, se concluye que:

- El uso de la tecnología Blockchain permite otorgar características de seguridad, trazabilidad, inmutabilidad y transparencia en el proceso de gestión de cualquier bien digital que aplique dicha tecnología; el mismo que ha podido ser implementado a través del módulo de software de autenticidad de certificados académicos digitales del presente TT.
- La aplicación de la metodología ABCDE en sus cuatro primeras fases en bloque con el estándar de la IEEE 830 que se fundamentan en la Ingeniería de Requisitos, permitió definir de forma idónea el módulo de software para la validación de certificados académicos digitales.
- El uso de la metodología ABCDE permitió que se segmente el diseño y codificación del subsistema de contratos inteligentes y el subsistema de aplicaciones, de manera eficiente.
- La ejecución de Pruebas Unitarias permitió evaluar y comprobar el código de cada subsistema, arrojando resultados positivos por medio de las herramientas Jest, Enzyme y Truffle.
- Con la última fase de la metodología ABCDE permitió integrar las Pruebas de Integración, Funcionales y Aceptación, se verificó el despliegue en las testnet Rinkeby en conjunto con el servicio de computación en la nube Microsoft Azure, permitiéndonos realizar una comprobación del aplicativo de validación de autentificación de certificados académicos digitales de forma positiva.
- El costo de la transferencia al registrar un certificado académico digital en la testnet Rinkeby de Ethereum, varía entre 0.00016 a 0.0002 ether que equivale a 0.50 y 0.63 dólares respectivamente, hay que tener en cuenta que el valor en dólares varía de acuerdo al a la tasa de conversión ETH/USD.
- Con base en la actual pandemia de COVID-19, que obliga a digitalizar todos los procesos como en la educación y la forma de representar el conocimiento adquirido, sea a través de certificados académicos digitales. En el que se encuentra el problema de emitir y validarlos, la solución planteada y ejecutada en el TT permite registrar y validar certificados académicos digitales de forma segura, transparente, y confiable.

9. Recomendaciones

Una vez culminado el Trabajo de Titulación, se recomienda que:

- Hacer uso de la metodología ABCDE para el desarrollo de sistemas descentralizados en los que intervenga la tecnología Blockchain, debido a que las metodologías ágiles o tradicionales conocidas a lo largo de la formación académica no cumplen con la necesidad de desarrollo de una DApp, a tal punto que no se considera en ninguna el término de contrato inteligente.
- A la carrera de Ingeniería en Sistemas/Computación fomentar el desarrollo en DApps con tecnología Blockchain. Teniendo en cuenta que los avances tecnológicos y la necesidad de los estudiantes de tener una formación actualizada, con la finalidad de estar a la vanguardia y poder brindarle herramientas que le permitan enfrentarse al campo laboral y profesional en un futuro cercano.
- Explorar el Trabajo de Titulación como referencia para generar módulos de software o sistemas en el que se implemente la tecnología Blockchain, en el campo de la salud en el registro de historias médicas o en la industria farmacéutica para verificar medicamentos, entre otros. Tomando en cuenta que los diferentes trabajos relacionados encontrados son muy escasos llegando al punto de nulos a nivel local o nacional.
- A la Unidad de Telecomunicaciones y la Información de la Universidad Nacional de Loja la aplicación de la tecnología Blockchain no solo para emitir o verificar los certificados académicos digitales, sino en otros procesos internos para que ayude a mejorar la experiencia de la comunidad universitaria y apuntar a estar a la vanguardia de este mundo cada vez más globalizado.

Se recomienda para trabajos a futuros:

- Investigar e impulsar en proyectos que generen metodologías de desarrollo de desarrollo de software, marcos de trabajo, patrones de diseño o modelado, entre otros, que permita tener un catálogo de opciones al momento de un desarrollo de software con tecnología Blockchain.
- Integrar de forma sustancial el módulo de software a un sistema de una entidad privada o pública como la Universidad Nacional de Loja, para que se brinde un servicio que a nivel de país no se da, generando un reconocimiento no solo nacional sino internacional.
- Al sistema que implemente el módulo de software al momento de emitir el archivo en formato PDF del certificado académico digital se lo generé en un patrón único para brindar mayor seguridad.

- En el módulo de software al momento de registrar un certificado académico digital generé un código QR, para que los usuarios finales puedan incluirlo en sus hojas de vidas o en donde sea de su interés, que redireccione automáticamente a la validación de autenticidad del certificado.
- Para el despliegue en la red Blockchain se puede indagar en otras en el que los costos puedan ser más económicos con respecto a los generados en la testnet de Rinkeby de Ethereum, en las que se fundamenta el Trabajo de Titulación o eventualmente indagar en la posibilidad de realizar el despliegue en la red mainnet de Ethereum en donde el pago de las transferencias es real.

10. Bibliografía

Referencias Bibliográficas

[1] NU. CEPAL and UNESCO, “La educación en tiempos de la pandemia de COVID-19,” CEPAL, 2020. [Online]. Available: https://www.siteal.iiep.unesco.org/respuestas_educativas_covid_19.

[2] C. Belloch, “Recursos Tecnologicos: TIC.” 2016. Accessed: Feb. 07, 2021. [Online]. Available: https://www.uv.es/bellochc/pedagogia/EVA2.wiki?0

[3] D. Lizcano, “Aproximación basada en Blockchain para crear un modelo de confianza en la enseñanza superior abierta y ubicua,” *Centro de Estudios Financieros, CEF*.

[4] R. Estudillo, “Certificados académicos digitales en la cadena de bloques | Monitor Educativo,” *Instituto de Investigación, Innovación y Estudios de Posgrado para la Educación*. Jun. 2018. Accessed: Feb. 07, 2021. [Online]. Available: https://monitor.iiiepe.edu.mx/notas/certificados-académicos-digitales-en-la-cadena-de-bloques

[5] F. Bond, F. Amati, and G. Blousson, “Blockchain, caso práctico de verificación académica,” 2015.

[6] C. Pastorino, “Blockchain: qué es, cómo funciona y cómo se está usando en el mercado | WeLiveSecurity.” Sep. 2018. Accessed: Feb. 28, 2021. [Online]. Available: https://www.welivesecurity.com/la-es/2018/09/04/blockchain-que-es-como-funciona-y-como-se-esta-usando-en-el-mercado/

[7] Banco Interamericano de Desarrollo, “Regulación de blockchain e identidad digital en América Latina.” 2020. Accessed: Feb. 28, 2021. [Online]. Available: https://publications.iadb.org/publications/spanish/document/Regulacion-de-blockchain-e-identidad-digital-en-America-Latina-El-futuro-de-la-identidad-digital.pdf

[8] J. Valencia, “Contrato inteligentes,” *Revista de Investigación en Tecnologías de la Información*, vol. 7, no. 14, pp. 1–10, 2019, doi: 10.36825/riti.07.14.001.

[9] S. Moscoso, D. Suárez, and D. Martin, “Certificación digital de documentos académicos usando Blockchain Formato IEEE,” *Tecnología Investigación y Academia*, vol. 7, no. 2, pp. 21–27, 2020, [Online]. Available: https://revistas.udistrital.edu.co/index.php/tia/article/view/12757

[10] D. Plaza, “Diseño y desarrollo de diplomas académicos digitales mediante la tecnología blockchain,” Cali, 2019. [Online]. Available:

http://vitela.javerianacali.edu.co/bitstream/handle/11522/11222/Diplomas_académicos_digitales_Blockchain.pdf?sequence=1&isAllowed=y

[11] C. Piris, A. Cabellos, and J. Vilanova, “Design and implementation of the Transactional and Communication layer of a Blockchain to secure IP prefixes Autor,” 2018.

[12] E. Piscini, D. Dalton, and L. Kehoe, “Panorama de Blockchain y Ciberseguridad Blockchain & Ciberseguridad,” 2018. [Online]. Available: https://www.bloomberg.com/news/articles/2016

[13] Universidad del Rosario, “Smart Contracts.”

[14] V. Palacios, “Explorando la Blockchain de Ethereum y el desarrollo de smart contracts,” Catalunya, Feb. 2018.

[15] Blockchain Academy México, “Blockchain 2.0 | Smart Contract,” 2020, Accessed: Mar. 14, 2021. [Online]. Available: https://blockchainacademy.mx/

[16] Escuela Crypto ES, *Programando Ethereum: Crea el futuro*, Kindle. 2021.

[17] Blockchain Academy México, “Blockchain 2.0 | Smart Contract | DAapp,” 2020. Accessed: Mar. 14, 2021. [Online]. Available: https://blockchainacademy.mx/

[18] Bit2Me Academy, “¿Qué es la Ethereum Virtual Machine (EVM)? | Bit2Me Academy.” https://academy.bit2me.com/que-es-ethereum-virtual-machine-evm/ (accessed Mar. 13, 2021).

[19] Jesús Lucas, “Qué es NodeJS y para qué sirve,” *OpenWebinars*, Sep. 04, 2019. https://openwebinars.net/blog/que-es-nodejs/ (accessed Aug. 07, 2021).

[20] A. Muñoz, “Introduccion-a-Nodejs.pdf,” Madrid, May 2013. Accessed: Aug. 07, 2021. [Online]. Available: https://www.academia.edu/38559025/Introduccion_a_Nodejs_pdf

[21] Web Age Solutions Inc., “Chapter 1-Introduction to Single Page Applications”.

[22] R. Bermejo, “Single-Page Applications (SPA),” Jun. 10, 2016. https://itblogsogeti.com/2014/06/10/single-page-applications-roberto-bermejo-sogeti/ (accessed Apr. 10, 2022).

[23] Facebook Inc., “React – Una biblioteca de JavaScript .” https://es.reactjs.org/ (accessed Aug. 07, 2021).

[24] Tutoriales en PDF, “React JS ,” 2018. https://tutorialesenpdf.com/reactjs/ (accessed Aug. 07, 2021).

[25] Desde Linux, “Truffle Suite: Herramientas de código abierto para Blockchain.” https://blog.desdelinux.net/truffle-framework-herramientas-codigo-abierto-blockchain/#Ganache (accessed Mar. 13, 2021).

[26] Microsoft, “Documentation for Visual Studio Code.” https://code.visualstudio.com/docs (accessed Aug. 07, 2021).

[27] R. Crespo, “Funciones Hash: MD5 y SHA-256,” 2020. http://www.robertocrespo.net/kaizen/funciones-hash-md5-y-sha256/ (accessed Aug. 07, 2021).

[28] D. Rachmawati, T. Tarigan, and A. B. C. Ginting, “A comparative study of Message Digest 5(MD5) and SHA256 algorithm,” 2017, doi: 10.1088/1742-6596/978/1/012116.

[29] Ask Any Difference, “Diferencia entre SHA y MD5.” https://askanydifference.com/es/diferencia-entre-sha-y-md5/ (accessed Aug. 07, 2021).

[30] L. Marchesi, M. Marchesi, and R. Tonelli, “ABCDE-agile block chain DApp engineering,” Cagliari, Nov. 2020. Accessed: Aug. 22, 2021. [Online]. Available: https://reader.elsevier.com/reader/sd/pii/S2096720920300026?token=C7ABB69531EE9D9BF844D0FB4CD0972FC4845DABFA6CE14E39B70797AB33CC2194C8A67287483155BD184992500CF4BA&originRegion=us-east-1&originCreation=20210824012952

[31] P. David, “Certificados Académicos Digitales mediante Blockchain,” Feb. 2019, Accessed: Aug. 28, 2021. [Online]. Available: http://vitela.javerianacali.edu.co/bitstream/handle/11522/11222/Articulo_cientifico.pdf?sequence=2&isAllowed=y

[32] E. Fernandez, J. Gutierrez, R. Delgado, and R. Lopez, “Aplicación web para la gestión de diplomas digitales en centros de capacitación mediante firma electrónica y blockchain,” 2020.

[33] D. Suárez and S. Moscoso, “Diseño de la Arquitectura de un Sistema de Contratos Inteligentes Basada en la Tecnología Blockchain Aplicada al Proceso de Registro de Estudiantes en el Sistema de Educación Colombiano,” 2018. [Online]. Available: http://repository.udistrital.edu.co/handle/11349/13640

[34] C. Narváez and J. Carrión, “Smart Contracts for user registration on Ethereum technology: Systematic Literature Review,” 2021.

[35] L. Rosero, M. Morales, and S. Morales, “Registro de títulos académicos mediante una aplicación basada en Blockchain y Smart Contracts,” Quito, May 2020. Accessed: Aug.

28, 2021. [Online]. Available: https://revistadigital.uce.edu.ec/index.php/CATEDRA/article/view/2200/2801

[36] G. Hernández, “Método Analítico,” Pachuca, 2017. Accessed: May 24, 2021. [Online]. Available: https://www.uaeh.edu.mx/docencia/P_Presentaciones/b_huejutla/2017/Metodo_Analitico.pdf

[37] G. Barchini, “Métodos ‘I + D’ de la Informática,” Santiago del Estero. Accessed: May 24, 2021. [Online]. Available: http://laboratorios.fi.uba.ar/lie/Revista/Articulos/020205/A2ago2005.pdf

[38] Consejo de la Judicatura, *Reglamento del sistema notarial integral de la función* . Consejo de la Judicatura, 2017. Accessed: Dec. 07, 2021. [Online]. Available: https://www.funcionjudicial.gob.ec/www/pdf/resoluciones/2017/216-2017.pdf

11. Anexos

Anexo 1. Especificación de requisitos de software

Especificación de requisitos de software

Proyecto: Módulo de software para la validación de autenticidad de certificados académicos digitales por tecnología Blockchain

Versión: 3.0

Fecha: 28/06/2021

Ficha del documento

Versión	Fecha de revisión	Cambios	Motivos del cambio
1.0	18/06/2021	El requisito no funcional de rendimiento (RNF01)	Debe incluirse en la descripción del requisito el tiempo de una transacción en la testnet de Ethereum.
2.0	25/06/2021	Agregar actores en la funcionalidad del producto	Agregar los actores del contrato inteligente y de usuario administrador.
3.0	28/06/2021	N/A	N/A

Índice de Contenidos

Índice de Figuras

Índice de Tablas

1. Introducción

Este documento es una Especificación de Requisitos Software (ERS) para el Módulo de software para la validación de autenticidad de certificados académicos digitales por tecnología Blockchain. Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE Práctica Recomendada para Especificaciones de Requisitos Software ANSI/IEEE 830, 1998.

1.1. Propósito

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales para el desarrollo de un módulo de software que permitirá la validación de autenticidad de certificados académicos digitales por tecnología blockchain.

1.2. Alcance

Esta especificación de requisitos está dirigida al usuario del sistema, que tiene como objetivo validar los certificados académicos digitales por tecnología blockchain.

1.3. Personal involucrado

Tabla 1.
Personal involucrado estudiante de la CIS

Nombre	Edgar Patricio Sánchez Malla
Rol	Analista y Desarrollador de Software
Categoría Profesional	Estudiante de la CIS
Responsabilidad	Análisis de información, diseño y programación del módulo de software
Información de contacto	edgar.sanchez@unl.edu.ec

Tabla 2.
Personal involucrado docente de la CIS

Nombre	Cristian Ramiro Narváez Guillen
Rol	Director del trabajo de titulación
Categoría Profesional	Docente de la de la CIS/C
Responsabilidad	Supervisar y asesorar en el desarrollo del Trabajo de Titulación
Información de contacto	cristian.narvaez@unl.edu.ec

1.4. Definiciones, acrónimos y abreviaturas

Tabla 3.
Definiciones, acrónimos y abreviaturas

Nombre	Descripción
BC	Blockchain
CAD	Certificado académico digital

CIS	Carrera de Ingeniería en Sistemas
DNI	Documento nacional de identificación
ERS	Especificación de Requisitos Software
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional
SM-VC	Contrato Inteligente para la validación de certificados UNL
Usuario	Persona que usará el sistema para validar los certificados académicos digitales
SPA-VC	Subsistema de aplicaciones para interactuar con el usuario final

1.5. Referencias

Tabla 4.
Referencias ERS

Título del Documento	Referencia
IEEE Std 830-1998	IEEE Recommended Practice for Software Requirements Specifications

1.6. Resumen

Este documento consta de tres secciones. En la primera sección se realiza una introducción al mismo y se proporciona una visión general de la especificación de recursos del sistema.

En la segunda sección del documento se realiza una descripción general del sistema, con el fin de conocer las principales funciones que éste debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles.

Por último, la tercera sección del documento es aquella en la que se definen detalladamente los requisitos que debe satisfacer el sistema.

2. Descripción general

2.1. Perspectiva del producto

El módulo de software de validación de CAD será un producto diseñado para trabajar en entornos WEB, lo que permitirá su utilización de forma rápida y eficaz, además de dar características de transparencia, inmutabilidad, seguridad y trazabilidad.

2.2. Funcionalidad del producto

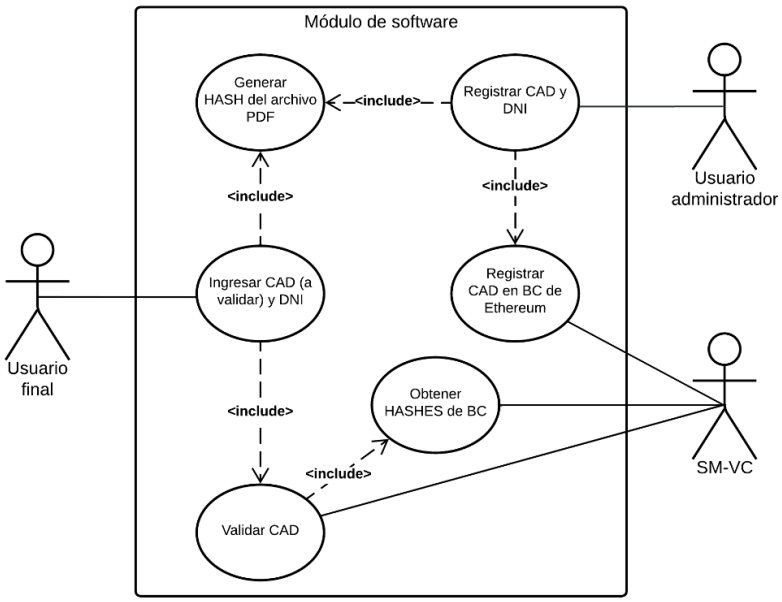


Figura 1. Roles y actividades.

2.3. Características de los usuarios

Tabla 5.
Características usuario de usuario final

Tipo de usuario	Usuario final
Formación	N/A
Actividades	• Ingresar DNI del usuario. • Cargar el certificado académico digital a validar.

Tabla 6.
Características usuario administrador

Tipo de usuario	Usuario administrador
Formación	N/A
Actividades	Registra la información del certificado académico digital y DNI del usuario al que le pertenezca.

Tabla 7.
Características usuario SM-VC

Tipo de usuario	SM-VC
Formación	N/A
Actividades	El contrato inteligente que contiene las funcionalidades para interactuar con al testnet de Blockchain.

2.4. Restricciones

- Interfaz para ser usada con internet.
- Se utilizará las herramientas de Truffle Suite.

- El módulo de software podrá ser utilizado en los navegadores Chrome, Mozilla o Edge.
- El módulo de software requiere internet para su correcto funcionamiento.
- Lenguajes y tecnologías en uso: HTML, CSS, React JS, JavaScript, Infura, Rinkeby, Truffle Suite y Solidity.
- Los servidores deben ser capaces de atender consultas concurrentemente.
- El sistema deberá tener un diseño e implementación sencilla, independiente de la plataforma o del lenguaje de programación.

2.5. Suposiciones y dependencias

- Se asume que los requisitos aquí descritos son estables.
- Los equipos en los que se vaya a ejecutar el sistema deben cumplir los requisitos antes indicados para garantizar una ejecución correcta de la misma

3. Requisitos específicos

3.1. Requisitos comunes de las interfaces

3.1.1. Interfaces de usuario

La interfaz con el usuario consistirá en un conjunto de ventanas con botones, listas y campos de textos. Ésta deberá ser construida específicamente para el sistema propuesto y, será visualizada desde un navegador de internet.

Figura 2. Interfaz gráfica inicial.

Figura 3. Interfaz gráfica en caso de obtener un resultado positivo.

Figura 4. Interfaz gráfica en caso de obtener un resultado negativo.

3.1.2. Interfaces de hardware

Será necesario disponer de equipos de cómputos en perfecto estado con las siguientes características:

- Computador:
    - Pantalla de al menos 14 pulgadas, 800 dpi.
    - Mínimo un procesador core i3, Pentium dual core.
    - Mínimo memoria RAM de 2GB.
    - Periféricos de Entrada/Salida.
- Conectividad:
    - Conexión a internet.

3.1.3. Interfaces de software

- Sistema Operativo: Windows 7 o superior.
- Explorador: Mozilla o Chrome.

3.2. Requisitos funcionales

Tabla 8.
Requisito funcional generar HASH de certificado académico digital

Identificación del requerimiento:	RF01
Nombre del Requerimiento:	Generar HASH de certificado académico digital
Descripción del requerimiento:	El módulo de software generará un HASH 256 del certificado académico digital (archivo PDF)
Dependencias:	El usuario administrador debe tener un archivo PDF.
Requerimiento NO funcional:	• RNF03 • RNF04
Prioridad del requerimiento:	Alta

Tabla 9.
Requisito funcional obtener DNI del usuario

Identificación del requerimiento:	RF02
Nombre del Requerimiento:	Obtener DNI del usuario
Descripción del requerimiento:	El módulo de software recibirá el DNI del usuario al que pertenece el certificado académico digital
Dependencias:	El usuario administrador debe conocer el DNI al que le pertenece el CDA del RF01
Requerimiento NO funcional:	• RNF01 • RNF04
Prioridad del requerimiento:	Alta

Tabla 10.
Requisito funcional registrar CAD en la blockchain

Identificación del requerimiento:	RF03
Nombre del Requerimiento:	Registrar CAD en la Blockchain
Descripción del requerimiento:	El módulo de software por medio de un formulario y del contrato inteligente (smart contract) almacenará el hash del CAD y el DNI del usuario en la Blockchain de

	Ethereum, solo por la cuenta que despliegue los contratos inteligentes
Dependencias:	• RF01 • RF02
Requerimiento NO funcional:	• RNF04
Prioridad del requerimiento:	Alta

Tabla 11.
Requisito funcional obtener DNI y CAD a validar

Identificación del requerimiento:	RF04
Nombre del Requerimiento:	Obtener DNI y CAD a validar
Descripción del requerimiento:	El módulo de software permitirá recibir a través de un formulario web el DNI del usuario y un archivo en formato PDF del CAD a validar.
Dependencias:	N/A
Requerimiento NO funcional:	RNF04
Prioridad del requerimiento:	Alta

Tabla 12.
Requisito funcional generar HASH del CAD a validar

Identificación del requerimiento:	RF05
Nombre del Requerimiento:	Generar HASH del CAD a validar
Descripción del requerimiento:	El módulo de software generará un HASH 256 del CAD a validar en formato PDF, recibido del formulario web.
Dependencias:	RF04
Requerimiento NO funcional:	RNF04
Prioridad del requerimiento:	Alta

Tabla 13.
Requisito funcional obtener CADs registrados

Identificación del requerimiento:	RF06
Nombre del Requerimiento:	Obtener CADs registrados
Descripción del requerimiento:	El módulo de software por medio del contrato inteligente (smart contract) consultará el HASH o HASHES registrados
Dependencias:	RNF03
Requerimiento NO funcional:	RNF04
Prioridad del requerimiento:	Alta

Tabla 14.
Requisito funcional validar el certificado académico digital

Identificación del requerimiento:	RF07
Nombre del Requerimiento:	Validar el certificado académico digital
Descripción del requerimiento:	El módulo de software buscará en los CADs obtenidos del RF06, si existe un CAD idéntico al generado en el RF04.
Dependencias:	• RF04 • RF06

Requerimiento NO funcional:	• RNF03 • RNF04
Prioridad del requerimiento:	Alta

Tabla 15.
Requisito funcional mostrar resultado

Identificación del requerimiento:	RF08
Nombre del Requerimiento:	Mostrar resultado
Descripción del requerimiento:	El módulo de software alertará con un mensaje en el sitio web, el resultado de la validación del certificado académico digital.
Dependencias:	RF05
Requerimiento NO funcional:	• RNF02 • RNF04
Prioridad del requerimiento:	Alta

3.3. Requisitos no funcionales

Tabla 16.
Requisito no funcional rendimiento

Identificación del requerimiento:	RNF01
Nombre del Requerimiento:	Rendimiento
Descripción del requerimiento:	El módulo de software debe proporcionar un tiempo de respuesta aceptable aproximadamente entre 2 a 7 segundos. La transacción tarda de 2 a 5 segundos en una testnet de Blockchain de Ethereum y en un ambiente real de 15 segundos a 5 minutos.
Prioridad del requerimiento:	Alta

Tabla 17.
Requisito no funcional usabilidad

Identificación del requerimiento:	RNF02
Nombre del Requerimiento:	Usabilidad
Descripción del requerimiento:	El módulo de software debe proporcionar una interfaz amigable e intuitiva, haciendo que el proceso sea comprensible y fácil de llevar a cabo. Además, debe permitir ser utilizado en cualquier navegador web.
Prioridad del requerimiento:	Alta

Tabla 18.
Requisito no funcional fiabilidad

Identificación del requerimiento:	RNF03
Nombre del Requerimiento:	Fiabilidad
Descripción del requerimiento:	El módulo de software debe permitir la disponibilidad las 24 horas del día y los 7 días de la semana, y en caso de que el módulo de software presente algún error, se debe recuperar en el menor tiempo posible.

	El módulo de software debe permitir recuperar los datos que se vean afectados en el caso de alguna falla en el módulo de software respecto al tiempo y esfuerzo que este genere.
Prioridad del requerimiento:	Alta

Tabla 19.
Requisito no funcional seguridad

Identificación del requerimiento:	RNF04
Nombre del Requerimiento:	Seguridad
Descripción del requerimiento:	El módulo de software debe garantizar disminuir las vulnerabilidades de ataques de fuerza bruta. Garantizar la seguridad del módulo de software con respecto a la información y datos que se manejan tales sean documentos o archivos.
Prioridad del requerimiento:	Alta

Anexo 2. Descripción de Historias de usuario

	Enunciado de la historia				Criterios de aceptación			
Identificador (ID) de la historia	Rol	Característica / Funcionalidad	Razón / Resultado	Número (#) de escenario	Criterio de aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
HU01	Como usuario final	Necesito validar un CAD	Con la finalidad de validar la autenticidad del CAD	1	Validar un CAD auténtico	En caso que se vaya a validar un CAD registrado	Cuando haga clic en el botón validar	El módulo de software debe presentar un mensaje donde indique el que el certificado es autentico
				2	Validar un CAD no auténtico	En caso que se vaya a validar un CAD registrado	Cuando haga clic en el botón validar	El módulo de software debe presentar un mensaje donde indique el que el certificado no es autentico
				3	Validar la copia de un CAD auténtico	En caso que se vaya a validar un CAD registrado	Cuando haga clic en el botón validar	El módulo de software debe presentar un mensaje donde indique el que el certificado es autentico
				4	Validar un CAD modificado	En caso que se vaya a validar un CAD registrado	Cuando haga clic en el botón validar	El módulo de software debe presentar un mensaje donde indique el que el certificado no es autentico
HU02	Como usuario administrador	Necesito registrar un CAD	Con la finalidad de registrar un CAD en la red Blockchain	1	Registrar un CAD	En caso que se vaya a registrar un CAD	Cuando haga clic en el botón registrar en la blockchain	El módulo de software debe presentar una ventana de metamask para confirmar la transacción y un mensaje de que el

								certificado académico digital fue registrado con éxito
				2	Registrar un CAD ya registrado	En caso que se vaya a registrar un CAD	Cuando haga clic en el botón registrar en la blockchain	El módulo de software debe presentar un mensaje donde indique el CAD ya se encuentra registrado
HU03	Como Contrato inteligente (SM-VC)	Necesito validar un CAD	Con la finalidad de validar si el CAD se encuentra registrado en la red Blockchain	1	Validar un CAD registrado	En caso que se requiera validar si el CAD se encuentra registrado	Cuando se llame a la función validateCAD del contrato inteligente	El contrato inteligente en la función debe devolver un valor de TRUE
				2	Validar un CAD no registrado	En caso que se requiera validar si el CAD se encuentra registrado	Cuando se llame a la función validateCAD del contrato inteligente	El contrato inteligente en la función debe devolver un valor de FALSE
HU04	Como Contrato inteligente (SM-VC)	Necesito registrar un CAD	Con la finalidad de validar si el CAD se encuentra registrado en la red Blockchain	1	Registrar un CAD	En caso que se vaya a registrar un CAD	Cuando se llame a la función registerCAD del contrato inteligente	El contrato inteligente debe registrar en la red Blockchain el DNI y el SHA-256 del CAD
				2	Registrar un CAD ya registrado	En caso que se vaya a registrar un CAD	Cuando se llame a la función registerCAD del contrato inteligente	El contrato inteligente en la función debe devolver un mensaje de que el CAD ya se encuentra registrado

HU05	Como SPA-VC	Necesito generar el SHA-256 del CAD	Con la finalidad de generar el SHA-256 del CAD	1	En caso que se genere el SHA-256 de un CAD (documento que si se puede leer)	En caso que se genere el SHA-256 de un CAD	Cuando se cargue el documento PDF en el formulario	La función readfile debe leer el documento y convertirlo en una cadena binaria el contenido del documento para generar el SHA-256
				2	En caso que se genere el SHA-256 de un CAD (documento que no se puede leer)	En caso que se genere el SHA-256 de un CAD	Cuando se cargue el documento PDF en el formulario	Emitir un mensaje de error en caso de una lectura fallida del documento del CAD

Anexo 3. Paper BCCA 2021

En la siguiente página se muestra el documento en formato IEEE.

Blockchain technology to validate the authenticity of academic digital certificates

Edgar Patricio Sánchez Malla
Facultad de Energía, CIS Universidad Nacional de Loja
Loja, Ecuador
edgar.sanchez@unl.edu.ec
0000-0003-0784-7370

Cristian R. Narváez Guillén
Facultad de Energía, CIS
Universidad Nacional de Loja
Loja, Ecuador
cristian.narvaez@unl.edu.ec
0000-0002-9096-1010

Ruperto A. López
Facultad de Energía, CIS
Universidad Nacional de Loja
Loja, Ecuador
ruperto.lopez@unl.edu.ec 0000-0003-0202-2361

Pablo F. Ordoñez-Ordoñez
CIS, Universidad Nacional de Loja
ETSISI, Universidad Politécnica de Madrid
Loja, Ecuador
pfordonez@unl.edu.ec 0000-0001-8079-7694

***Abstract*—Educational institutions or organisations in a glob- alised world the way to represent the acquired knowledge is by means of an academic digital certificate, which grants the same academic and legal guarantees that later can be shared with any employer, university or governmental entity. But the trust of the users is deficient due to the multiple ways of falsifying them and the centralisation of the information that gives way to fraud. It is for them that a solution is presented that includes Blockchain technology that gives transparency, immutability and traceability. This paper presents the results of the first phase in which the software module for the authenticity validation of digital academic certificates using Requirements Engineering is defined. In which related work is indicated. In addition to the functional and non-functional requirements, the diagrams of: use cases and the architecture of the software module. Being these the basis to develop the software module in the future.**

***Index Terms*—Architecture, DApp, Decentralise, Ethereum, Requirements.**

I. Introduction

Blockchain technology allows the transfer of digital data with a very sophisticated encryption and in a completely secure way, i.e., this transfer or procedure does not require a centralised intermediary (a bank or a notary) to identify and certify the information contained therein, but this one is distributed in multiple nodes independent of each other, which register and validate it. Once the one is on the blockchain, "the one cannot be deleted, only new records can be added, and it will not be legitimised unless the majority of them agree to do so" [1]. In which the following characteristics can be identified taken from [2]:

- Decentralise.
- Open source
- Autonomy
- Immutability
- Anonymity
- Traceability

A. Blockchain Types

Blockchain technology can be divided into two types to below:

- **Public Blockchain**. It is one that has no restrictions in terms of reading its data and viewing transactions for inclusion in the blockchain, all participants have the right to send transactions and in case they are valid, these will be included in the Blockchain. For the consensus process all nodes have the possibility to participate [3]. It is considered "fully decentralized" architecture. For instance, Bitcoin and Ethereum are public Blockchain [2].
- **Private blockchain**. It is one that has direct access to blockchain data. Furthermore, the view of transactions is limited due to a predefined list of entities, write permission is held by only one organisation, and read per- missions can be public or somewhat arbitrarily restricted. By its nature, in most cases it does not provide significant advantages over the traditional centralised trust model [3].

B. Applications of Blockchain

This technology, basically any type of information that needs to be preserved intact and must remain available can be stored in blockchain in a secure, decentralised and cheaper way than through intermediaries. The following is a description of some of the uses according to [4]:

- **Health**. Health records could be unified and stored in blockchain. In this manner, the medical history of each patient would be secure and at the same time available to each authorised physician, regardless of the health centre where the patient has been treated. In the pharmaceutical industry, it can be used to verify drugs and prevent counterfeiting.
- **For documents**. It would be very useful for the manage- ment of digital goods and documents. Blockchain makes it possible to register purchases, deeds, documents or any

type of digital asset that cannot be falsified.

- **Other uses**. It can revolutionise the Internet of Things (IoT) market, where there are millions of Internet-connected devices that need to be managed by the provider companies. In the near future, the centralised model will not support many devices, not to mention that many of them are not secure enough. With blockchain technology, devices can communicate over the network directly, securely and reliably, with not intermediaries.

C. Smart contracts

Smart contracts are computer programs that run on the larger blockchain, have become a feature of blockchain. This type of program can be used to facilitate, verify or enforce rules between parties, allowing straight-through processing and interaction with other smart contracts [5].

These are the main characteristics that make it possible to rely on smart contracts according to [6]:

- Immutability
- Distributed
- Deterministic
- Verifiable

Currently, there are a wide variety of blockchain platforms that support Smart Contracts, but the largest and most adaptable is Ethereum. Bitcoin also has support for Smart Contracts, but these ones are very limited compared to Ethereum.

D. Ethereum

It is the second best known blockchain platform. It was created by the well-known Vitalik Buterin, a Russian programmer and writer. The purpose of Ethereum is to develop a platform that allows and facilitates other programmers to create decentralised applications (DApp) with Smart Contracts. At the same time, it serves as a global platform where these applications are carry out [6].

Ethereum Virtual Machine (EVM) is a virtual machine that form part of the Ethereum Blockchain Ecosystem. The main function is to allow the execution of programs or Smart Contracts, to deploy on the blockchain a series of added functionalities hence that users can enjoy them.

EVM built a specialised high-level language called Solidity. This language facilitates the creation of Smart Contracts. The process to follow is: first Solidity is transformed to the operation codes (OP_CODES) and then to a bytecode. This is finally executed by the EVM to perform the operations specified in a Smart Contract [7]. Thus, EVM can execute from the simplest to the most complex operations.

E. Event System of National University of Loja

The Events system of National University of Loja has the purpose of managing the events around the university and in some cases a digital academic certificate is issued. This is the current form in which the acquired knowledge is represented.

Nowadays, digital academic certificates are validated gives rise to the fraud that occurs both by forgery and by the complicity of authorities and staff of the study centers. Because of currently several institutions verify the authenticity of their academic certificates through an online consultation. Others delegate the task to third parties (notaries). Finally, there is no alternative other than direct contact with the study centre and its academic secretariat to confirm or not the validity of an academic certificate [8]. Furthermore, all the identification systems deployed until now present, to a greater or lesser extent, the problem of depending on a centralized entity and, consequently, removing the focus from the user. In other words, all these systems follow a scheme where an entity decides on the personal data, with some minor interventions by the user that generate a false sense of control over the identity data [9].

Based on this, it is proposed to implement a software module for the validation of digital academic certificates by blockchain technology in the Events system of the National University of Loja.

The following sections represent the related works; the materials and methods used; and the experimentation with their respective results of the first phase that indicates to define the software module for the validation of digital academic certificates using Requirements Engineering.

Likewise, it is necessary to know the following concepts in order to understand the software module architecture diagram that will be shown later.

- Truffle Suite. A set of tools to help developers create, test and deploy various software solutions on Ethereum Blockchain. The ones and their definitions are obtained from [10]:
  - Truffle Teams. It allows to manage and monitor the status of applications enabled on a Blockchain.
  - Truflee. It is a development environment, which provides a testing framework and a portfolio of assets for Blockchain using the Ethereum Virtual Machine.
  - Ganache. It is a personal Ethereum Blockchain to build and test developments such as DApps and Smart Contracts.
  - Drizzle. It is a collection of front-end libraries that make front-end development for DApps easier and more predictable.
- Solidity. Object-oriented programming language for writing smart contracts.
- MetaMask. Extension for web browsers, it is the gateway to Blockchain applications.
- Web3.js. Ethereum JavaScript API, a library that allows to interact with a local or remote Ethereum node using HTTP, IPC or WebSocket.

II. Related Works

The related works found according to the reviewed bibliography. In Latin America, there are projects that use Blockchain technology to issue and/or validate documents, providing

greater reliability and security to users, which are referred to below:

A. Digital Academic Certificates through Blockchain

In [2] is mentioned the software development of the platform Stampo and also is details the design and architecture of the generation of digital academic certificates on the activities of the students.

B. Blockchain-based digital certification of academic documents

According with [11], it is describe the main features of Blockchain and the diverse way that could be used in the process of issuing official documents, such as the generation of documents that required to be authenticated and/or verify.

C. Web application for managing digital diplomas in training centers using electronic signature and blockchain

Fernandez et at. [12] worked in a web application that allows managed the registry, issue and delivery of training certificates in an efficient way.

D. Design of a Blockchain-based smart contract system architecture applied to the student registration process in the Colombian education system.

On another hand, Suarez [13] looked for applying the Blockchain technology in a information system that could support to the solution of the people registry problems in the Colombian education system.

E. Smart Contracts for user registration on Ethereum technology: Systematic Literature Review

In [14] look for the way to development smart contracts inside of user registry module of a system. Moreover of detecting what are the best tool that contribute a best application of the smart contracts.

III. Materials and Methods

In this work, the set of materials, methods, techniques, standards and methodologies to be carried out during the development are elaborated, defined and systematised.

In which interview techniques and meetings are applied to obtain information on the requirements, the use of analytical, deductive and inductive methods to generate the requirements specification document by means of the IEEE 830 standard. Given, the coronavirus pandemic (COVID-19) has provoked an unprecedented crisis in all areas, in which multiple technological challenges are being faced. Consequently, the use of Information and Communication Technologies (ICTs) made it possible to carry out the experimentation of the first phase. The materials used are listed in Table I below:

TABLE I
MATERIALS USED

Software	
Detail	**Description**
Zoom	Video chat software that allows collaborative work, reviews, interviews, among others.
OneDrive	It is a service that allowed the storage and collaborative work of documents.
Lucidchart	Software that allows the construction of different diagrams or figures.
InVision	Prototype development tool that allowed the development of the graphical interface.

IV. Experimentation and Results

Based on the purpose of 'Define the software module for the validation of digital academic certificates using Requirements Engineering'. The results obtained are detailed below.

A. Software Requirements

On base of the IEEE 830 Standard, it could determined the functional requirements as shown in Table II and non functional requirements as shown in Table III as well as the use cases diagram as shown in Fig. 2.

TABLE II
FUNCTIONAL REQUIREMENTS

Codification	Requirement	Description
FR01	Generate HASH of Digital Academic Certificate (DAC)	The software module will generate a HASH 256 of the academic digital certificate (PDF file).
FR02	Get user National Identity Document (NID)	The software module will receive the NID of the user to whom the academic digital certificate belongs.
FR03	Register on the Blockchain	The software module via smart contract will store the DAC hash and the user's NID on the Ethereum Blockchain.
FR04	Get NID and DAC to validate	The software module will allow to receive through a web form the user's NID and a PDF file of the DAC to be validated.
FR05	Generate HASH of the DAC to be validated	The software module shall generate a HASH 256 of the DAC to be validated in PDF format, received from the web form.
FR06	Get HASHES from user	The software module by means of the smart contract will query the HASH or HASHES linked to the user's NID.
FR07	Validate the academic digital certificate	The software module shall search the HASHES obtained from FR06, if there is a HASH identical to the one generated in FR05.
FR08	Show result	The software module will alert with a message on the website, the result of the validation of the academic digital certificate.

B. Software Architecture for the application

For understanding the architecture, it is necessary to remember that a tradition web application uses HTML, CSS and JS to create a web page as well as taking into account of uses an API to connect with a DB.

On another hand, an DApp uses the similar Frontend technology than a tradition web application. Nevertheless, there exist a radical difference with the Backend, instance of using

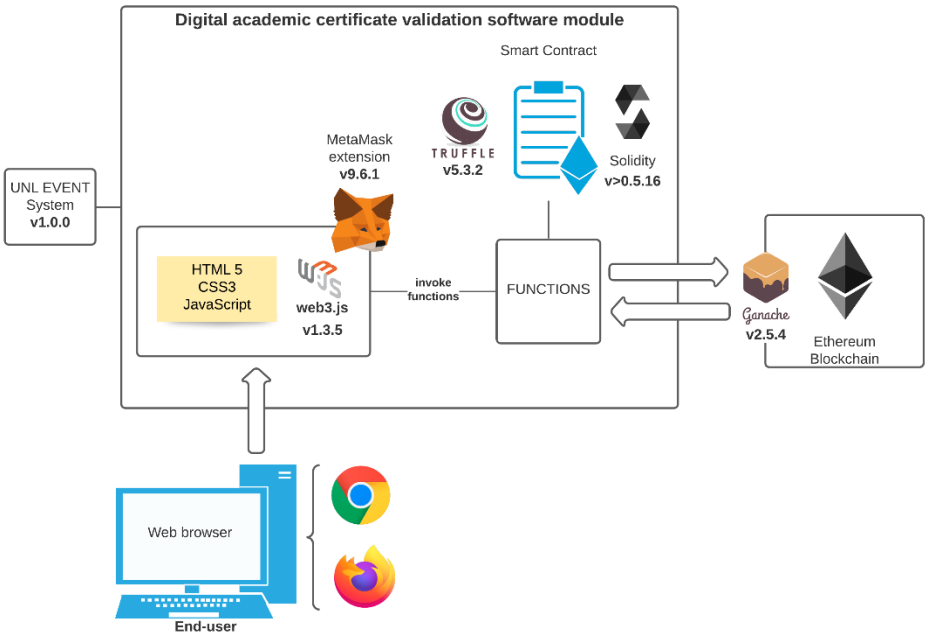


Fig. 1. Software Architecture Diagram proposed for the application

TABLE III
NON FUNCTIONAL REQUIREMENTS

Codification	Requirement	Description
NFR01	Performance	The application should provide a response time acceptable near between 2and 7 seconds. Given a transaction used to delay between 2 and 5 second in a testnet of Ethereum Blockchain and a production environment used to take time between 15 and 300 seconds.
NFR02	Usability	The application should provide an UI friendly and intuitive, allowing that the process will be compressible and easy to carry out. Moreover, it must be used in whatever browser web.
NFR03	Reliability	The software application should be available 24 hours on day, 7 day on week and 365 days on year. The response time should be the minus possible when one presented any error. Also, the one should allow a fast recovery the data when this will be affected in any failure cases respect to the time and human effort to solve the problem.
NFR04	Security	The application must be in the capacity to warranty and supporting any brute-force attack. Also it should warrant the security of the application with respect to the information and data managed.

an API to connect with a DB, the DApp uses a Smart Contract to connect with a Blockchain as shown in Fig. 3.

Consequently, the software architecture as shown in Fig. 1 is used for the validation of academic digital certificates: By means of the Frontend (HTML, CSS and JavaScript), web3.js and the MetaMask extension, it allows to invoke the functions of a smart contract developed under the Solidity programming language and the Truffle tool. The implementation of this tools

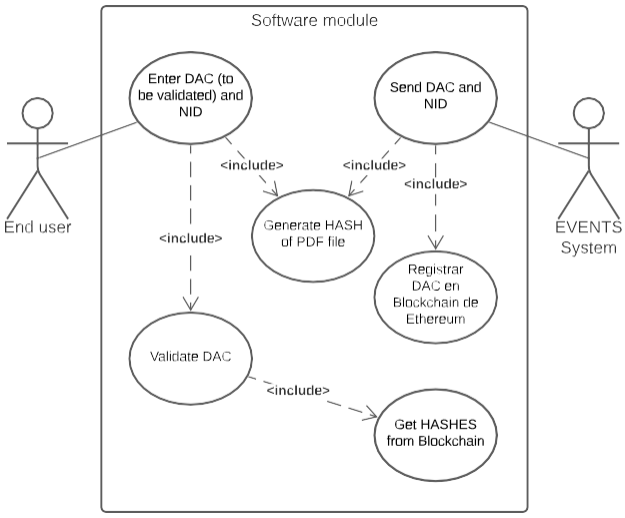


Fig. 2. Use Case Diagram

allow to connect to the local Ethereum Blockchain (testnet) generated with Ganache.

V. CONCLUSIONS AND FUTURE WORKS

According to the results obtained, it is determined that the functional and non-functional requirements together with the use case diagram and the architecture of the software module are a fundamental piece and the starting point for the next phase of the project, which is the development of the software module. The importance of the software module architecture diagram lies in the fact that it is a guide that helps

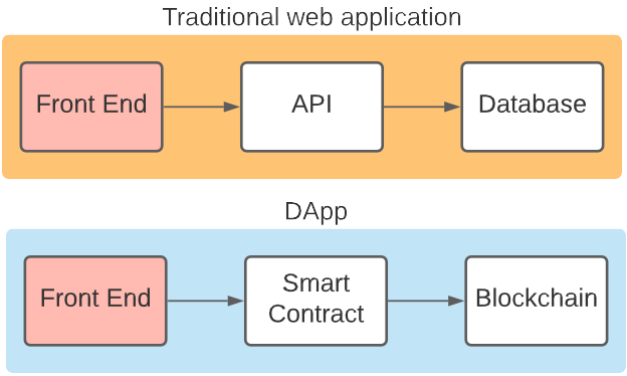


Fig. 3. Comparative between a web app and a DApp

to understand how Blockchain is involved in a real project, such as the Events system of the National University of Loja.

Once the architecture and the tools to be used have been defined, the next step would be to develop of the solution and testing the solution in a blockchain testnet for the validation of the authenticity of academic digital certificates, where the different tools and technologies mentioned will be put into practice. This will allow us to demonstrate that the implementation of a software project where Blockchain technology is implemented generates advantages in relation to traditional projects where it is not used, from the transparency and trust that is given to the end user to the traceability and immutability that can be given to the different information that is generated

REFERENCES

[1] J. Valencia, "Contrato inteligentes," *Revista de Investigación en Tecnologías de la Información*, vol. 7, no. 14, pp. 1–10, 2019.

[2] S. Moscoso, D. Su árez, and D. Martin, "Certificación digital de documentos académicos usando Blockchain Formato IEEE," *Tecnología Investigación y Academia*, vol. 7, no. 2, pp. 21–27, 2020.

[3] D. Plaza, *Diseño y desarrollo de diplomas académicos digitales mediante la tecnolog´ıa blockchain*. PhD thesis, Pontificia Universidad Javeriana Cali, Cali, feb 2019.

[4] C. Pastorino, "Blockchain: qu é es, cómo funciona y cómo se está usando en el mercado — WeLiveSecurity," sep 2018.

[5] E. Piscini, D. Dalton, and L. Kehoe, "Panorama de Blockchain y Ciberseguridad Blockchain Ciberseguridad," tech. rep., Deloitte, 2018.

[6] V. Palacios, "Explorando la Blockchain de Ethereum y el desarrollo de smart contracts," tech. rep., Universitat Politècnica de Catalunya, Catalunya, feb 2018.

[7] Bit2Me Academy, "¿Qu e´ es la Ethereum Virtual Machine (EVM)? — Bit2Me Academy."

[8] F. Bond, F. Amati, and G. Blousson, "Blockchain, caso práctico de verificación académica," tech. rep., Amazon Web Services, 2015.

[9] Banco Interamericano de Desarrollo, "Regulación de blockchain e identidad digital en América Latina," 2020.

[10] Desde Linux, "Truffle Suite: Herramientas de código abierto para Blockchain."

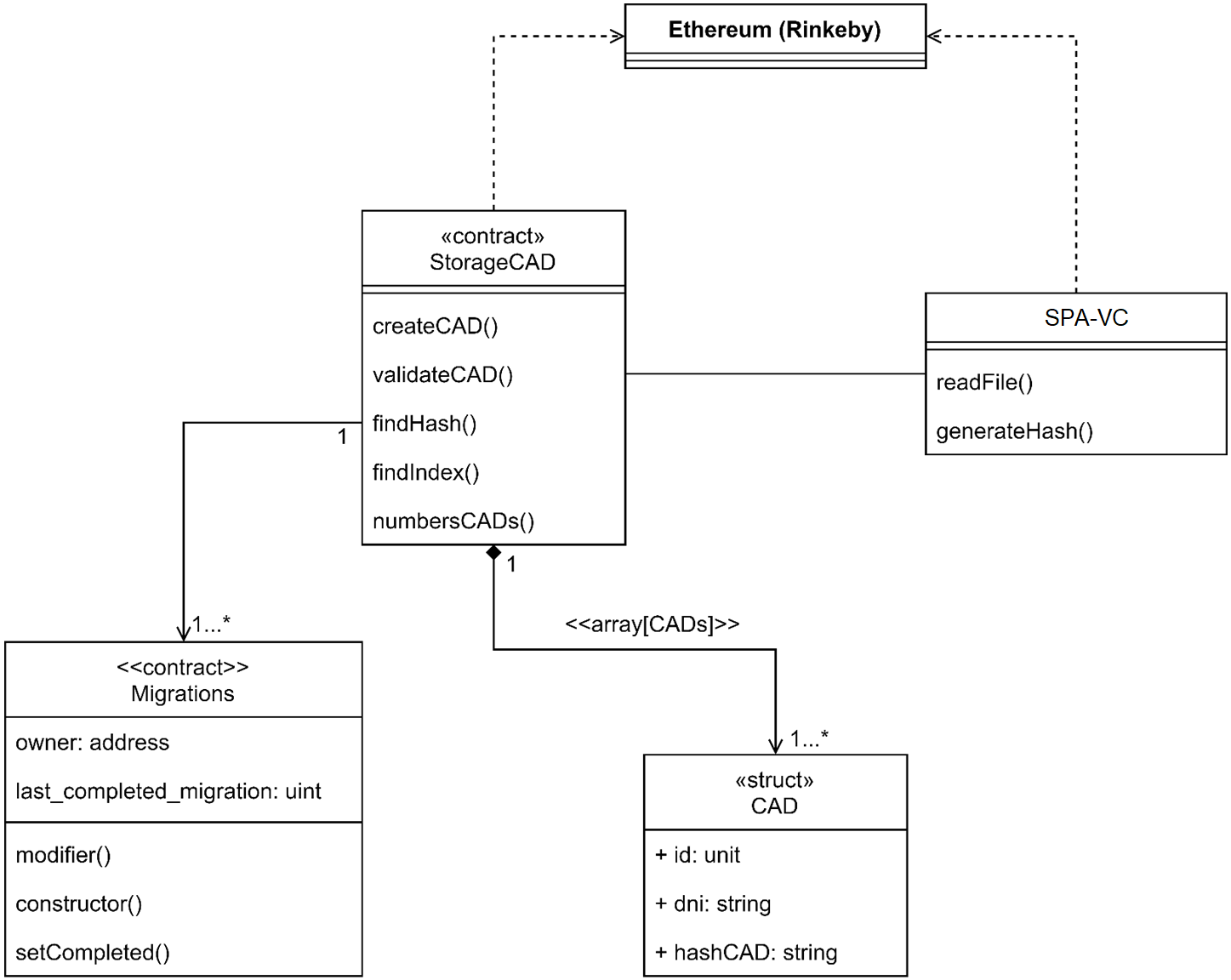
[11] P. David, "Certificados Académicos Digitales mediante Blockchain," tech. rep., 2019.

[12] E. Fernandez, J. Gutierrez, R. Delgado, and R. Lopez, "Aplicación web para la gestión de diplomas digitales en centros de capacitación mediante firma electrónica y blockchain," tech. rep., 2020.

[13] D. Su árez and S. Moscoso, "Diseño de la Arquitectura de un Sistema de Contratos Inteligentes Basada en la Tecnolog´ıa Blockchain Aplicada al Proceso de Registro de Estudiantes en el Sistema de Educación Colombiano," tech. rep., 2018.

[14] C. Narváez and J. Carrión, "Smart Contracts for user registration on Ethereum technology: Systematic Literature Review," 2021.

Anexo 4. Descripción del Diagrama de Clases



Anexo 5. Código de programación de subsistema de contratos inteligentes

Código de programación
del subsistema de
Contratos Inteligentes

Proyecto: Módulo de software para la validación de autenticidad de certificados académicos digitales por tecnología Blockchain

Fecha: 02/08/2021

Índice

1. Código fuente del contrato inteligente Migrations.sol

El código fuente se observa en la Tabla 1.

Tabla 1.
Código fuente del contrato Migrations.sol

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.4.21 <0.7.0;

contract Migrations {
  address public owner;
  uint public last_completed_migration;

  modifier restricted() {
    if (msg.sender == owner) _;
  }

  constructor() public {
    owner = msg.sender;
  }

  function setCompleted(uint completed) public restricted {
    last_completed_migration = completed;
  }
}
```

2. Código fuente del contrato inteligente StorageCAD.sol

El código fuente se observa en la Tabla 2.

Tabla 2.
Código fuente del contrato StorageCAD.sol

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.4.21 <0.7.0;

contract StorageCAD {

    address public owner;
    modifier restricted() {
        if (msg.sender == owner) _;
    }

    constructor() public {
        owner = msg.sender;
    }

    uint256 nextId;
    struct CAD {
        uint256 id;
        string dni;
```

```
        string hashCAD;
    }

    CAD[] cads;

    function registerCAD( string memory _dni, string memory _hashCAD, address _owner ) public {
        bool registered = findHash(_hashCAD);
        if (registered == false) {
            if (owner == _owner) {
                cads.push(CAD(nextId, _dni, _hashCAD));
                nextId++;
            } else {
                revert("Intento registrar con una cuenta diferente al propietario del despliegue del
contrato");
            }
        } else {
            revert("El certificado academico digital esta duplicado");
        }
    }

    function validateCAD(string memory _dni, string memory _hashCAD) public view returns (bool) {
        for (uint256 i = 0; i < cads.length; i++) {
            if (
                keccak256(abi.encodePacked((cads[i].dni))) == keccak256(abi.encodePacked((_dni))) &&
                keccak256(abi.encodePacked((cads[i].hashCAD))) ==
keccak256(abi.encodePacked((_hashCAD)))
            ) {
                return true;
            }
        }
    }

    function findHash(string memory _hashCAD) public view returns (bool) {
        for (uint256 i = 0; i < cads.length; i++) {
            if (
                keccak256(abi.encodePacked((cads[i].hashCAD))) ==
                keccak256(abi.encodePacked((_hashCAD)))
            ) {
                return true;
            }
        }
    }

    function findIndex(string memory _dni, string memory _hashCAD) internal view returns (uint256) {
        for (uint256 i = 0; i < cads.length; i++) {
            if (
                keccak256(abi.encodePacked((cads[i].dni))) ==
                keccak256(abi.encodePacked((_dni))) &&
                keccak256(abi.encodePacked((cads[i].hashCAD))) ==
                keccak256(abi.encodePacked((_hashCAD)))
            ) {
                return i;
```

```
            }
        }
        revert("El certificado academico digital no fue encontrado");
    }

    function readCAD(string memory _dni, string memory _hashCAD) public view returns (
            uint256,
            string memory,
            string memory
        )
    {
        uint256 index = findIndex(_dni, _hashCAD);
        return (cads[index].id, cads[index].dni, cads[index].hashCAD);
    }

    function updateCAD(string memory _dni, string memory _hashCAD) public {
        uint256 index = findIndex(_dni, _hashCAD);
        cads[index].dni = _dni;
        cads[index].hashCAD = _hashCAD;
    }

    function deleteCAD(string memory _dni, string memory _hashCAD) public {
        uint256 index = findIndex(_dni, _hashCAD);
        delete cads[index];
    }

    function numbersCADs() public view returns (uint256) {
        uint256 size = cads.length;
        return size;
    }
}
```

Anexo 6. Plan de Pruebas Unitarias del subsistema de contratos inteligentes

Plan de Pruebas Unitarias subsistema de Contratos Inteligentes

Proyecto: Módulo de software para la validación de autenticidad de certificados académicos digitales por tecnología Blockchain

Versión: 2.0

Fecha: 02/08/2021

Hoja de control

Organismo	Universidad Nacional de Loja		
Proyecto	Módulo de software para la validación de certificados académicos digitales por tecnología blockchain		
Entregable	Planes de Pruebas Unitarias del subsistema de contratos inteligentes		
Autor	Edgar Sánchez		
Versión/Edición	2.0	**Fecha Versión**	02/08/2021
Aprobado por	Cristian Ramiro Narváez Guillen, Mg.Sc.	**Fecha Aprobación**	05/08/2021
		Nº Total de Páginas	7

Registro de cambios

Versión doc	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Versión inicial del Plan de Pruebas Unitarias subsistema de Contratos Inteligentes	Edgar Patricio Sánchez Malla	30/07/2021
2.0	Versión final del Plan de Pruebas Unitarias subsistema de Contratos Inteligentes	Edgar Patricio Sánchez Malla	01/08/2021

Control de distribución

Nombre y Apellidos
Edgar Patricio Sánchez Malla
Cristian Ramiro Narváez Guillen, Mg. Sc

Índice

1. Introducción

Objeto

El objetivo de este documento es para verificar la funcionalidad correcta del subsistema de contratos inteligentes aislando cada parte del código y mostrar que las partes individuales son correctas. Esto respecto al código principal que se ejecuta en el módulo.

Propósito

Comprobar el correcto funcionamiento del subsistema de contratos inteligentes del módulo de software validación de autenticidad de certificados académicos digitales por tecnología Blockchain , por unidad de código principal, para asegurar que cada unidad funcione correctamente y eficientemente por separado.

2. Definición de los casos de pruebas

En este apartado se describe en detalle cada uno de los casos de pruebas que se identificaron:

Número del Caso de Prueba	Componente	Descripción de lo que se probará	Prerrequisitos
CP01	Contrato inteligente	Comprobar que enviado los parámetros de DNI y SHA-256 del CAD se registren correctamente	Contrato compilado correctamente
CP02	Contrato inteligente	Comprobar que enviado los parámetros de DNI y SHA-256 del CAD a validar se valide la autenticidad del CAD autentico, correctamente	Contrato compilado correctamente
CP03	Contrato inteligente	Comprobar que enviado los parámetros de DNI y SHA-256 del CAD a validar se valide la autenticidad del CAD no autentico, correctamente	Contrato compilado correctamente
CP04	Contrato inteligente	Comprobar que enviado el parámetro de SHA-256 del CAD registrado valide que si se encuentra	Contrato compilado correctamente
CP05	Contrato inteligente	Comprobar que enviado el parámetro de SHA-256 del CAD no registrado valide que no se encuentra	Contrato compilado correctamente
CP06	Contrato inteligente	Comprobar que devuelva el número correcto de CADs registrados	Contrato compilado correctamente

CP01					
Nº	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
1	Se registra el dni y el hash del CAD, y se verifica que se encuentre registrado, retornado el valor de verdadero	testItRegisterCAD()	dni, hashCAD	✓	N/A

CP02					
Nº	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
1	Se valida el dni y el hash del CAD a validar auténticos, y obtener el valor de verdadero	testItValidateCADAuthentic()	dni, hashCAD	✓	N/A

CP03					
Nº	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
1	Se valida el dni y el hash del CAD a validar no auténticos, y obtener el valor de falsos	testItValidateCADNotAuthentic()	dni, hashCAD	✓	N/A

CP04					
Nº	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
1	Se busca el hash del CAD que se encuentra registrado y obtener el valor de verdadero	testItFindHashCADRegistered()	hashCAD	✓	N/A

CP05					
Nº	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
1	Se busca el hash del CAD que no se encuentra registrado y obtener el valor de falso	testItFindHashCADNotRegistered()	hashCAD	✓	N/A

CP06					
Nº	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
1	Se comprueba que devuelva el valor correcto de CADs registrados	testItNumbersCADs()	N/A	✓	N/A

3. Glosario

A continuación, se muestra la definición de todos los términos utilizados en el presente documento.

Término	Descripción
CAD	Certificado académico digital
DNI	Documento nacional de identificación
PTT	Proyecto de Trabajo de Titulación

Anexo 7. Código de programación de subsistema de aplicaciones

Código de programación de subsistema de Aplicaciones

Proyecto: Módulo de software para la validación de autenticidad de certificados académicos digitales por tecnología Blockchain

Fecha: 03/08/2021

Índice

1. Estados y conexión con web3, metamask, contrato y cuentas

El código fuente se observa en la Tabla 1.

Tabla 1.
Estados y conexión con web3, metamask, contrato y cuentas

```
state = { storageDNI: "", storageHashCAD: "", validate: null, showValidate: true, showRegister: false, numberOfRegistrations: 0, web3: null, accounts: null, contract: null, isDeploymentOwner: null, balance: null, addressContract: null };

constructor(props) {
    super(props);
    this.fileInput = React.createRef();
    this.textFileInput = React.createRef();
    this.textLabelDNI = React.createRef();
    this.cardValidate = React.createRef();
  }

  componentDidMount = async () => {
    Swal.fire({
      title: 'Cargando...',
      html: '<b>Inicia sesión </b> en la wallet de MetaMask<br />',
      allowEscapeKey: false,
      didOpen: () => {
        Swal.showLoading()
      },
      backdrop:true,
      allowOutsideClick: () => {
        const popup = Swal.getPopup()
        popup.classList.remove('swal2-show')
        setTimeout(() => {
          popup.classList.add('animate__animated', 'animate__headShake')
        })
        setTimeout(() => {
          popup.classList.remove('animate__animated', 'animate__headShake')
        }, 500)
        return false
      }
    });

    try {
      const web3 = await getWeb3();
      const accounts = await web3.eth.getAccounts();
      const networkId = await web3.eth.net.getId();
      const deployedNetwork = StorageCADContract.networks[networkId];
      const instanceContract = new web3.eth.Contract(
        StorageCADContract.abi, deployedNetwork && deployedNetwork.address, );
      const addressContract = deployedNetwork.address.toString();
      const balance = await web3.eth.getBalance(accounts[0]);
      const balanceETHER = web3.utils.fromWei(balance, 'ether');
      const deploymentOwner = await instanceContract.methods.owner().call();
      if (accounts[0] === deploymentOwner) {
```

```
        this.setState({ isDeploymentOwner: true });
      } else {
        this.setState({ isDeploymentOwner: false });
      };

      this.setState({ web3, accounts, contract: instanceContract, balance: balanceETHER, addressContract: addressContract }, this.run);
    } catch (error) {
      Swal.fire({
        icon: 'error',
        title: '¡Atención!',
        html: 'No se ha podido cargar',
        timer: 3000,
        showConfirmButton: false
      });
      console.error(error);
    }
  };
```

2. Función de registrar y validar un CAD

El código fuente se observa en la Tabla 2.

Tabla 2.
Función de registrar y validar un CAD

```
  // Función para registrar un CAD
  handleSubmit = async () => {
    const { storageDNI, storageHashCAD, accounts, contract } = this.state;
    const registered = await contract.methods.findHash(storageHashCAD).call();
    if (storageDNI === "" || storageHashCAD === "") {
      Swal.fire({
        icon: 'warning',
        title: '¡Atención!',
        text: 'Los campos deben estar completos para registrar el certificado académico digital en la red Blockchain',
        confirmButtonText: 'Aceptar',
      });
    }
    else if (registered === false) {
      Swal.fire({
        title: 'Espere...',
        html: '<b>Confirmar</b> la transacción y esperar el mensaje de confirmación (esto tomará un tiempo)...
                <br />',
        didOpen: () => {
          Swal.showLoading()
        },
        backdrop:true,
        allowOutsideClick: () => {
          const popup = Swal.getPopup()
          popup.classList.remove('swal2-show')
```

```
        setTimeout(() => {
          popup.classList.add('animate__animated', 'animate__headShake')
        })
        setTimeout(() => {
          popup.classList.remove('animate__animated', 'animate__headShake')
        }, 500)
        return false
      }
    });
    try {
      let result = await contract.methods.registerCAD(storageDNI, storageHashCAD, accounts[0]).send({ from: accounts[0] });
      Swal.fire({
        icon: 'success',
        title: '¡Correcto!',
        text: 'El certificado académico digital se registró en la red Blockchain con éxito',
        confirmButtonText: 'Aceptar',
      });
      this.resetForm();
      console.log(result);
    } catch (error) {
      Swal.fire({
        icon: 'warning',
        title: '¡Atención!',
        html: 'Error al registrar el certificado académico digital <br />' +
          'Inténtelo más tarde o revise el siguiente mensaje:' +
          '<hr style="border:1px dashed #dfdfdf; width:90%"></hr>' +
          '<span style="color:red">Error: </span>' +
          error.message,
        confirmButtonText: 'Aceptar'
      });
    };
  } else {
    Swal.fire({
      icon: 'warning',
      title: '¡Atención!',
      text: 'El certificado académico digital ya se encuentra registrado en la red Blockchain',
      confirmButtonText: 'Aceptar',
    });
  }
  const number = await contract.methods.numbersCADs().call();
  this.setState({ numberOfRegistrations: number });
};

// Función para validar si el CAD es o no autentico
handleValidate = async () => {
  const { storageDNI, storageHashCAD, contract } = this.state;
  if (storageDNI === "" || storageHashCAD === "") {
    Swal.fire({
      icon: 'warning',
      title: '¡Atención!',
```

```
        text: 'Los campos deben estar completos para validar la autenticidad del certificado académico digital',
        confirmButtonText: 'Aceptar',
      });
    } else {
      try {
        const response = await contract.methods.validateCAD(storageDNI, storageHashCAD).call();
        const Toast = Swal.mixin({
          toast: true,
          position: 'top-end',
          showConfirmButton: false,
          timer: 5000,
          timerProgressBar: true,
          didOpen: (toast) => {
            toast.addEventListener('mouseenter', Swal.stopTimer)
            toast.addEventListener('mouseleave', Swal.resumeTimer)
          }
        });
        Toast.fire({
          icon: 'info',
          title: 'Validación de autenticidad realizada, observe el resultado en pantalla'
        });
        this.setState({ validate: response });
        const card = this.cardValidate.current;
        card.classList.add('animate__animated', 'animate__backInRight');
        setTimeout(() => card.classList.remove('animate__animated', 'animate__backInRight'), 1000);
      } catch (error) {
        Swal.fire({
          icon: 'warning',
          title: '¡Atención!',
          html: 'Error al validar el certificado académico digital <br />' +
            'Inténtelo más tarde o revise el siguiente mensaje:' +
            '<hr style="border:1px dashed #dfdfdf; width:90%"></hr>' +
            '<span style="color:red">Error: </span>' +
            error.message,
          confirmButtonText: 'Aceptar'
        });
      };
    }
  };
```

3. Función de leer el CAD y generar SHA-256

El código fuente se observa en la Tabla 3.

Tabla 3.
Función de leer el CAD y generar SHA-256

```
readFile = (e) => {
    const input = this.fileInput.current;
    const inputTextFile = this.textFileInput.current;
    const extValidate = /(.pdf)$/i;
```

```
    if (!extValidate.exec(input.value)) {
      inputTextFile.classList.remove('invalid');
      inputTextFile.classList.add('invalid');
      this.setState({ storageHashCAD: "" });
    } else {
      inputTextFile.classList.remove('invalid');
      inputTextFile.classList.add('valid');
      const file = e.target.files[0];
      if (!file) return;
      const fileReader = new FileReader();
      fileReader.readAsBinaryString(file);
      fileReader.onload = () => {
        const hash256File = sha256(fileReader.result);
        this.setState({ storageHashCAD: hash256File });
      }
      fileReader.onerror = () => {
        Swal.fire({
          title: 'Error',
          icon: 'error',
          text: fileReader.error,
          allowEscapeKey: false,
          confirmButtonText: 'Aceptar',
          backdrop:true,
          allowOutsideClick: () => {
            const popup = Swal.getPopup()
            popup.classList.remove('swal2-show')
            setTimeout(() => { popup.classList.add('animate__animated', 'animate__headShake') })
            setTimeout(() => { popup.classList.remove('animate__animated', 'animate__headShake') },
500)
            return false
          }
        }).then((result) => {
          if (result.isConfirmed) { window.location.reload(true); }
        });
        console.log(fileReader.error);
      }
    }
  };
```

Anexo 8. Plan de Pruebas Unitarias del subsistema de aplicaciones

Plan de Pruebas Unitarias
subsistema de Aplicaciones

Proyecto: Módulo de software para la validación de autenticidad de certificados académicos digitales por tecnología Blockchain

Versión: 2.0

Fecha: 04/08/2021

Hoja de control

Organismo	Universidad Nacional de Loja		
Proyecto	Módulo de software para la validación de certificados académicos digitales por tecnología blockchain		
Entregable	Planes de Pruebas Unitarias del subsistema de Aplicaciones		
Autor	Edgar Sánchez		
Versión/Edición	2.0	**Fecha Versión**	04/08/2021
Aprobado por	Cristian Ramiro Narváez Guillen, Mg.Sc.	**Fecha Aprobación**	05/08/2021
		Nº Total de Páginas	7

Registro de cambios

Versión doc	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Versión inicial del Plan de Pruebas Unitarias subsistema de Aplicaciones	Edgar Patricio Sánchez Malla	01/08/2021
2.0	Versión final del Plan de Pruebas Unitarias subsistema de Aplicaciones	Edgar Patricio Sánchez Malla	04/08/2021

Control de distribución

Nombre y Apellidos
Edgar Patricio Sánchez Malla
Cristian Ramiro Narváez Guillen, Mg. Sc

Índice

1. Introducción

Objeto

El objetivo de este documento es para verificar la funcionalidad correcta del subsistema de Aplicaciones aislando cada parte del código y mostrar que las partes individuales son correctas. Esto respecto al código principal que se ejecuta en el módulo.

Propósito

Comprobar el correcto funcionamiento del módulo de software validación de autenticidad de certificados académicos digitales por tecnología Blockchain, por unidad de código principal, para asegurar que cada unidad funcione correctamente y eficientemente por separado.

2. Definición de los casos de pruebas

En este apartado se describe en detalle cada uno de los casos de pruebas que se identificaron:

Número del Caso de Prueba	Componente	Descripción de lo que se probará	Prerrequisitos
CP01	Sistema de aplicaciones	Comprobar que al no tener conexión con web3, contratos o cuentas, la página renderice adecuadamente	Navegador web
CP02	Sistema de aplicaciones	Comprobar que al tener conexión con web3, contratos o cuentas, la página renderice adecuadamente	Navegador web con extensión de metamask
CP03	Sistema de aplicaciones	Comprobar que enviado los parámetros de DNI y SHA-256 del CAD a validar se registre el CAD correctamente	Contrato compilado y migrado en formato JSON al subsistema de aplicaciones
CP04	Sistema de aplicaciones	Comprobar que enviado los parámetros de DNI y SHA-256 del CAD a validar se valide la autenticidad del CAD correctamente	Contrato compilado y migrado en formato JSON al subsistema de aplicaciones
CP05	Sistema de aplicaciones	Comprobar que pase de un formulario a otro adecuadamente	Página correctamente renderizada
CP06	Sistema de aplicaciones	Comprobar que se habilite el Input para cargar adecuadamente el CAD en formato PDF y generar su SHA-256	Página correctamente renderizada

CP01					
Nº	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
1	Comprobar que al no tener conexión con web3, contratos o cuentas, la página renderice adecuadamente	test('Renderizar <App /> al no estar conectado a web3, cuentas, contrato o metamask ')	N/A	✓	N/A

CP02					
Nº	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
1	Comprobar que al tener conexión con web3, contratos o cuentas, la página renderice adecuadamente	test('Renderizar <App /> correctamente al estar conectado correctamente')	N/A	✓	N/A
2	Comprobar que se llama a la función que realiza la petición de web3, contratos y cuentas	test('Llamar a componentDidMount()')	N/A	✓	N/A
3	Comprobar que llama a la función run()	test('Llamar a la función run()'	N/A	✓	N/A

CP03					
Nº	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
1	Mostrar adecuadamente el formulario de registro de un CAD	test('Mostrar formulario para registrar un CAD'	N/A	✓	N/A
2	Comprobar de que se llama correctamente a la función de registrar	test('Llamar a la función handleSubmit()')	N/A	✓	N/A

CP04					
Nº	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
1	Mostrar adecuadamente el formulario de validación de un CAD	test('Mostrar formulario para validar un CAD')	N/A	✓	N/A
2	Comprobar de que se llama correctamente a la función de validar	test('Llamar a la función handleValidate()'	N/A	✓	N/A
3	Mostrar el mensaje adecuado si el CAD es autentico	test('Mostrar el mensaje adecuado cuando el CAD es autentico')	N/A	✓	N/A
4	Mostrar el mensaje adecuado si el CAD no es autentico	test('Mostrar el mensaje adecuado cuando el CAD no es autentico')	N/A	✓	N/A

CP05					
Nº	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
1	Se comprueba que se cambia al formulario de validar certificados correctamente	test('Llamar a la función handleChangeValidate()')	N/A	✓	N/A
2	Se comprueba que se cambia al formulario de registrar certificados correctamente	test('Llamar a la función handleChangeRegister()'	N/A	✓	N/A

CP06					
Nº	**Descripción**	**Método**	**Datos Entrada**	**¿OK?**	**Observaciones**
1	Comprobar que se habilite el Input para cargar adecuadamente el CAD en formato PDF y generar su SHA-256	test('Llamar a la función readFile')	N/A	✓	N/A

3. Glosario

A continuación, se muestra la definición de todos los términos utilizados en el presente documento.

Término	Descripción
CAD	Certificado académico digital
DNI	Documento nacional de identificación
PDF	Formato de documento portátil
PTT	Proyecto de Trabajo de Titulación

Anexo 9. Plan de Pruebas de Integración

Plan de Pruebas de Integración

Proyecto: Módulo de software para la validación de autenticidad de certificados académicos digitales por tecnología Blockchain

Versión: 2.0

Fecha: 11/08/2021

Hoja de control

Organismo	Universidad Nacional de Loja		
Proyecto	Módulo de software para la validación de certificados académicos digitales por tecnología blockchain		
Entregable	Planes de Pruebas de Integración		
Autor	Edgar Sánchez		
Versión/Edición	2.0	**Fecha Versión**	11/08/2021
Aprobado por	Cristian Ramiro Narváez Guillen, Mg.Sc.	**Fecha Aprobación**	13/08/2021
		Nº Total de Páginas	9

Registro de cambios

Versión doc	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Versión inicial del Plan de Pruebas de Integración	Edgar Patricio Sánchez Malla	09/08//2021
2.0	Versión final del Plan de Pruebas de Integración	Edgar Patricio Sánchez Malla	13/08//2021

Control de distribución

Nombre y Apellidos
Edgar Patricio Sánchez Malla
Cristian Ramiro Narváez Guillen, Mg. Sc

Índice

1. Introducción

Objeto

El objetivo de este documento es recoger los casos de pruebas que verifican que el módulo de software a partir de los requerimientos funcionales del módulo para verificar el correcto ensamblaje y funcionalidad respecto a sus distintos componentes. Las pruebas de integración se elaboran después de la correcta ejecución de las pruebas unitarias; es decir una vez verificado el software de manera unitaria, verificar que no existan problemas en la combinación de elementos unitarios.

Alcance

Los casos de pruebas están dirigidos para el docente Ing. Cristián Narváez (director del TT), docente de la Carrera de Ingeniería en Sistemas / Computación de la Universidad Nacional de Loja, que valida los diferentes casos de pruebas. Y el estudiante Edgar Sánchez que es el que genera y registra los diferentes casos de pruebas.

2. Definición de los casos de pruebas

En este apartado se describe en detalle cada uno de los casos de pruebas que se identificaron:

Número del Caso de Prueba	Componentes	Descripción de lo que se probará	Prerrequisitos
CP01	Subsistema de contratos inteligentes – Subsistema de aplicaciones	Se registre correctamente el DNI y el SHA-256 del CAD en la testnet de Blockchain	• Ingresar a la página. • Inicio de sesión y conexión con la wallet de metamask (la contraseña de la wallet para el TT o frase de recuperación según corresponda). • Tener el certificado académico digital en formato PDF y el número de DNI.
CP02	Subsistema de contratos inteligentes – Subsistema de aplicaciones	Se valide el DNI y EL SHA-256 del CAD, si se encuentra en la testnet de Blockchain	• Ingresar a la página. • Inicio de sesión y conexión con la wallet de metamask (la contraseña de la wallet para el TT o frase de recuperación según corresponda). • Tener registrados certificados académicos digitales.
CP03	Wallet - Subsistema de aplicaciones	Conexión con la wallet de Metamask	• Ingresar a la página • Conocer la frase de recuperación de la wallet o la contraseña si ya inicio sesión previamente
CP04	Subsistema de aplicaciones-Blockchain	Al hacer clic en el botón de VALIDAR, se comunique con la blockchain	• Ingresar a la página. • Inicio de sesión y conexión con la wallet de metamask. • Tener el certificado académico digital en formato PDF y el número de DNI. • Completar el formulario de validación con lo mencionado anteriormente.
CP05	Subsistema de aplicaciones-Blockchain	Al hacer clic en el botón de REGISTRAR, se comunique con la blockchain	• Ingresar a la página. • Inicio de sesión y conexión con la wallet de metamask. • Tener el certificado académico digital en formato PDF y el número de DNI. • Completar el formulario de registro con lo mencionado anteriormente.

CP01					
Paso	**Descripción de pasos a seguir**	**Datos Entrada**	**Salida Esperada**	**¿OK?**	**Observaciones**
1	Hacer clic en el tab denominado “REGISTRAR CERTIFICADOS”.	Clic	Vista de validar	✓	N/A
2	Completar el campo DNI con el DNI del usuario	String DNI	String DNI	✓	
3	Cargar el documento del CAD en formato PDF	Archivo PDF	Cadena string que contiene el SHA-256	✓	
4	Hacer clic en el botón “REGISTRAR EN BLOCKCHAIN”.	Clic	Respuesta de un mensaje de espera y que indique que debo confirmar la transacción	✓	
5	Hacer clic en el botón confirmar de la ventana de metamask	Clic	Respuesta de un mensaje de que el registró fue realizado con éxito	✓	

CP02					
Paso	**Descripción de pasos a seguir**	**Datos Entrada**	**Salida Esperada**	**¿OK?**	**Observaciones**
1	Hacer clic en el tab denominado “VALIDAR CERTIFICADOS”, aunque por defecto se encuentra en esa vista.	Clic	Vista de registrar	✓	N/A
2	Completar el campo DNI con el DNI del usuario	String DNI	String DNI	✓	
3	Cargar el documento del CAD en formato PDF	Archivo PDF	Cadena string que contiene el SHA-256	✓	
4	Hacer clic en el botón “VALIDAR”	Clic	Respuesta de un mensaje con el resultado de la validación	✓	

CP03					
Paso	**Descripción de pasos a seguir**	**Datos Entrada**	**Salida Esperada**	**¿OK?**	**Observaciones**
1	Ingresar a la página	Clic	Ventana de la wallet para ingresar la contraseña	✓	N/A
2	Completar el campo con la contraseña	String	Habilitar el botón de DESBLOQUEAR	✓	
3	Hacer clic en el botón DESBLOQUEAR	Clic	Un mensaje de alerta con el saldo disponible y el sistema cargado completamente	✓	

CP04					
Paso	**Descripción de pasos a seguir**	**Datos Entrada**	**Salida Esperada**	**¿OK?**	**Observaciones**
1	Ingresar a la página	Clic	Ventana de la wallet para ingresar la contraseña	✓	N/A
2	Iniciar sesión en la wallet	String y clic	Un mensaje de alerta con el saldo disponible y el sistema cargado completamente	✓	
3	Llenar el formulario de validación	DNI Y archivo PDF	Habilitar el botón de VALIDAR	✓	
4	Hacer clic en el botón “VALIDAR”	Clic	Respuesta de un mensaje con el resultado de la validación	✓	

CP05					
Paso	**Descripción de pasos a seguir**	**Datos Entrada**	**Salida Esperada**	**¿OK?**	**Observaciones**
1	Ingresar a la página	Clic	Ventana de la wallet para ingresar la contraseña	✓	N/A
2	Iniciar sesión en la wallet	String y clic	Un mensaje de alerta con el saldo disponible y el sistema cargado completamente	✓	
3	Llenar el formulario de registro	DNI Y archivo PDF	Habilitar el botón de REGISTRAR	✓	
4	Hacer clic en el botón “REGISTRAR EN BLOCKCHAIN”	Clic	Ventana de la wallet de metamask	✓	
5	Aceptar la transacción	Clic	Mensaje de espera y finalizada la espera una alerta con el resultado del registro	✓	
6	Hacer clic sobre el mensaje del resultado	Clic	Una página nueva del Etherscan en que se detalla la transacción del registro	✓	

3. Glosario

A continuación, se muestra la definición de todos los términos utilizados en el presente documento.

Término	Descripción
CAD	Certificado académico digital
DNI	Documento nacional de identificación
PDF	Portable Document Format, «formato de documento portátil»
PTT	Proyecto de Trabajo de Titulación

4. Bibliografía y referencias

Referencia	Título
Anexo 1 (del Documento del Trabajo de Titulación)	Especificación de requisitos de Software IEEE 830.

Anexo 10. Plan de Pruebas Funcionales

Plan de Pruebas Funcionales

Proyecto: Módulo de software para la validación de autenticidad de certificados académicos digitales por tecnología Blockchain

Versión: 2.0

Fecha: 17/08/2021

Hoja de control

Organismo	Universidad Nacional de Loja		
Proyecto	Módulo de software para la validación de autenticidad de certificados académicos digitales por tecnología Blockchain		
Entregable	Planes de Pruebas Funcionales		
Autor	Edgar Sánchez		
Versión/Edición	1.0	**Fecha Versión**	17/08/2021
Aprobado por	Cristian Ramiro Narváez Guillen, Mg.Sc.	**Fecha Aprobación**	19/08/2021
		Nº Total de Páginas	19

Registro de cambios

Versión doc	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Versión inicial del Plan de Pruebas Funcionales	Edgar Patricio Sánchez Malla	14/10/2021
2.0	Versión final del Plan de Pruebas Funcionales	Edgar Patricio Sánchez Malla	17/08/2021

Control de distribución

Nombre y Apellidos
Edgar Patricio Sánchez Malla
Cristian Ramiro Narváez Guillen, Mg. Sc

Índice

1. Introducción

Objeto

El objetivo de este documento es recoger los casos de pruebas que verifican que el módulo de software satisface los requisitos especificados. Deberá contener la definición de los casos de prueba, la matriz de trazabilidad entre casos de pruebas y requisitos, y la estrategia a seguir en la ejecución de las pruebas.

Alcance

Los casos de pruebas están dirigidos para el docente Ing. Cristián Narváez (director del TT), docente de la Carrera de Ingeniería en Sistemas / Computación de la Universidad Nacional de Loja, que valida los diferentes casos de pruebas. Y el estudiante Edgar Sánchez que es el que genera y registra los diferentes casos de pruebas.

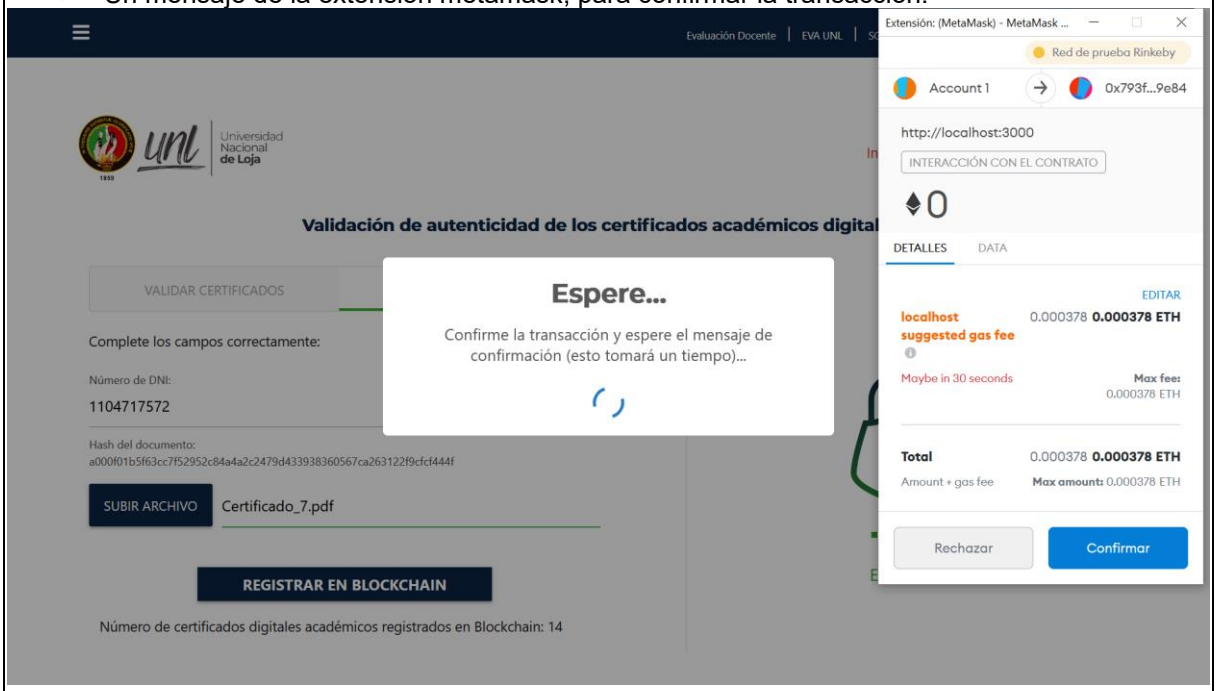
2. Trazabilidad de casos de pruebas – requisitos

En la siguiente MATRIZ se indica la correspondencia entre los casos de pruebas definidos y los requisitos funcionales de la especificación de requisitos. Las filas representan cada uno de los casos de pruebas definidos, y las columnas los requisitos funcionales. La “x” marca la relación existente.

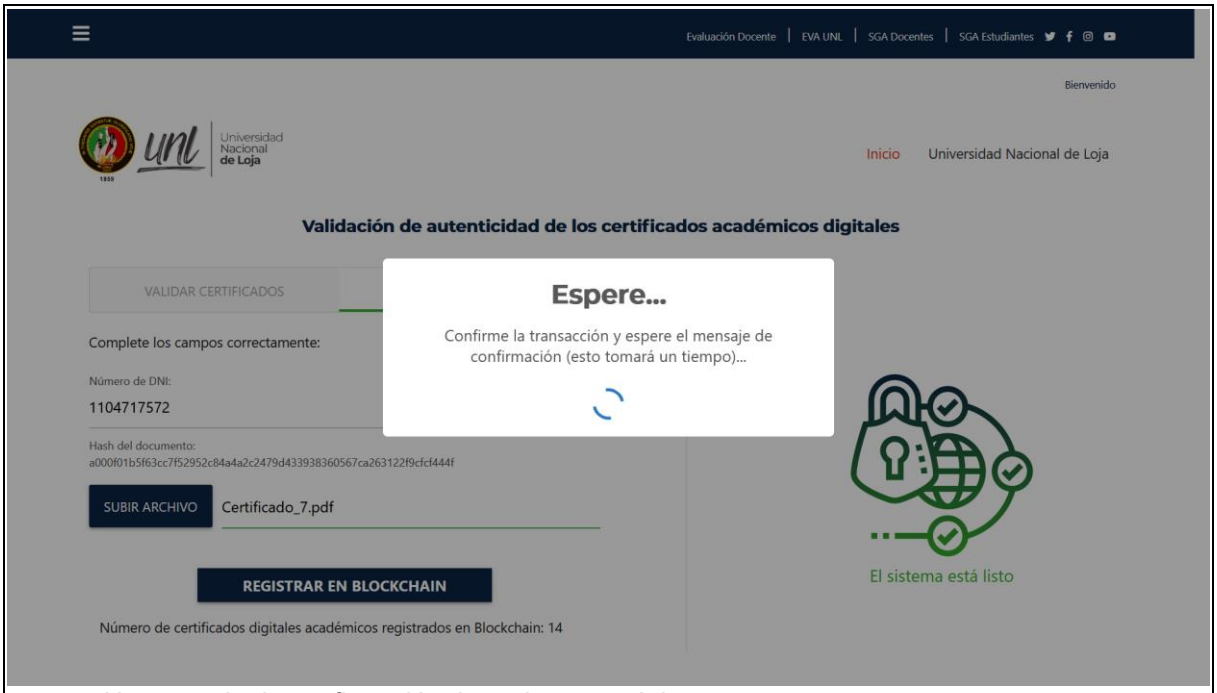
	RF01	RF02	RF03	RF04	RF05	RF06	RF07	RF08
CP01	x	x	x					x
CP02	x	x	x					x
CP03	x	x	x					x
CP04				x	x	x	x	x
CP05				x	x	x	x	x
CP06				x	x	x	x	x
CP07				x	x	x	x	x
CP08				x	x	x	x	x
CP09				x	x	x	x	x

3. Definición de los casos de pruebas

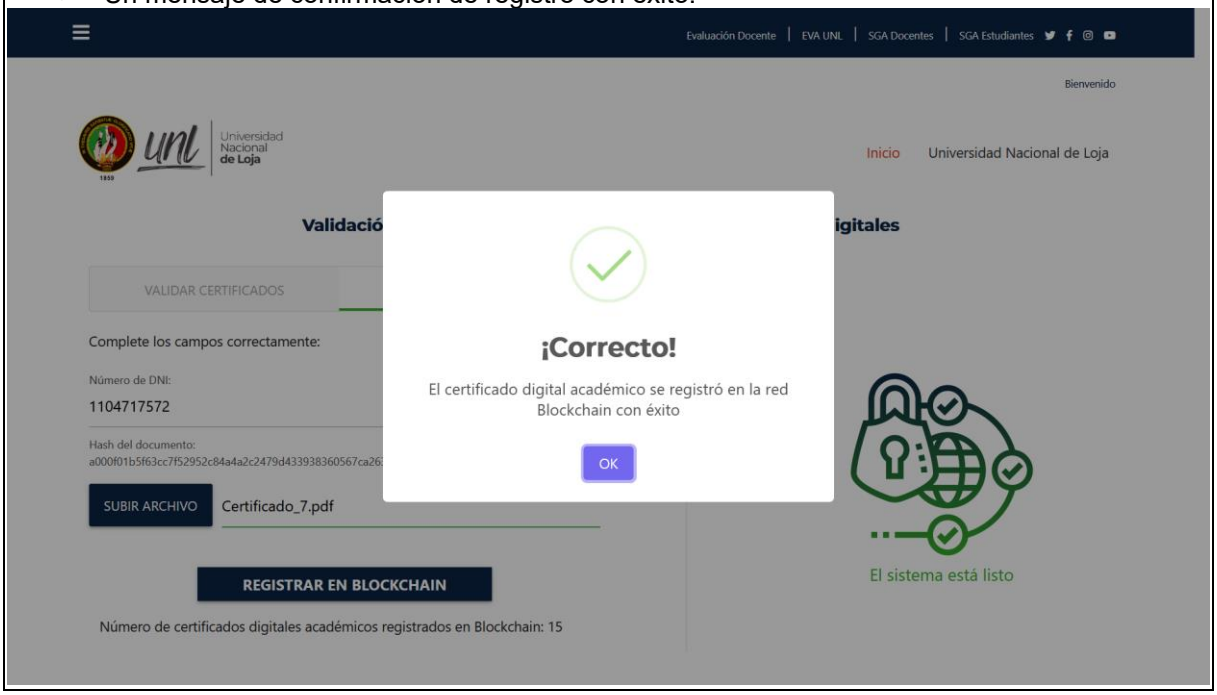
En este apartado se describe en detalle cada uno de los casos de pruebas que se identificaron para verificar la funcionalidad del sistema. Además de indicarse aquellos que deben realizarse para asegurar el correcto despliegue del módulo de software.

<table>
<tr><td rowspan="2">Registrar un DNI y un CAD</td><td>CP01</td><td></td></tr>
<tr><td>¿Prueba de despliegue?</td><td>Sí</td></tr>
<tr><td colspan="3">Descripción:
Se probará la respuesta del módulo de software al registrar un DNI con un certificado académico digital en formato PDF</td></tr>
<tr><td colspan="3">Prerrequisitos
• Ingresar a la página.
• Inicio de sesión y conexión con la wallet de metamask (la contraseña de la wallet para el TT o frase de recuperación según corresponda).
• Tener el certificado académico digital en formato PDF y el número de DNI.</td></tr>
<tr><td colspan="3">Pasos:
1. Hacer clic en el tab denominado “REGISTRAR CERTIFICADOS”.
2. Completar los campos de DNI y cargar el documento del CAD en formato PDF.
3. Hacer clic en el botón “REGISTRAR EN BLOCKCHAIN”.
4. Confirmar la transacción en la ventana que se genera automáticamente por metamask.
5. Esperar el mensaje de confirmación.</td></tr>
<tr><td colspan="3">Resultado esperado:
• Un mensaje de la extensión metamask, para confirmar la transacción.
• Un mensaje de espera.
• Un mensaje de confirmación de registro con éxito.</td></tr>
<tr><td colspan="3">Resultado obtenido:
• Un mensaje de la extensión metamask, para confirmar la transacción.
 

• Un mensaje de espera.</td></tr>
</table>

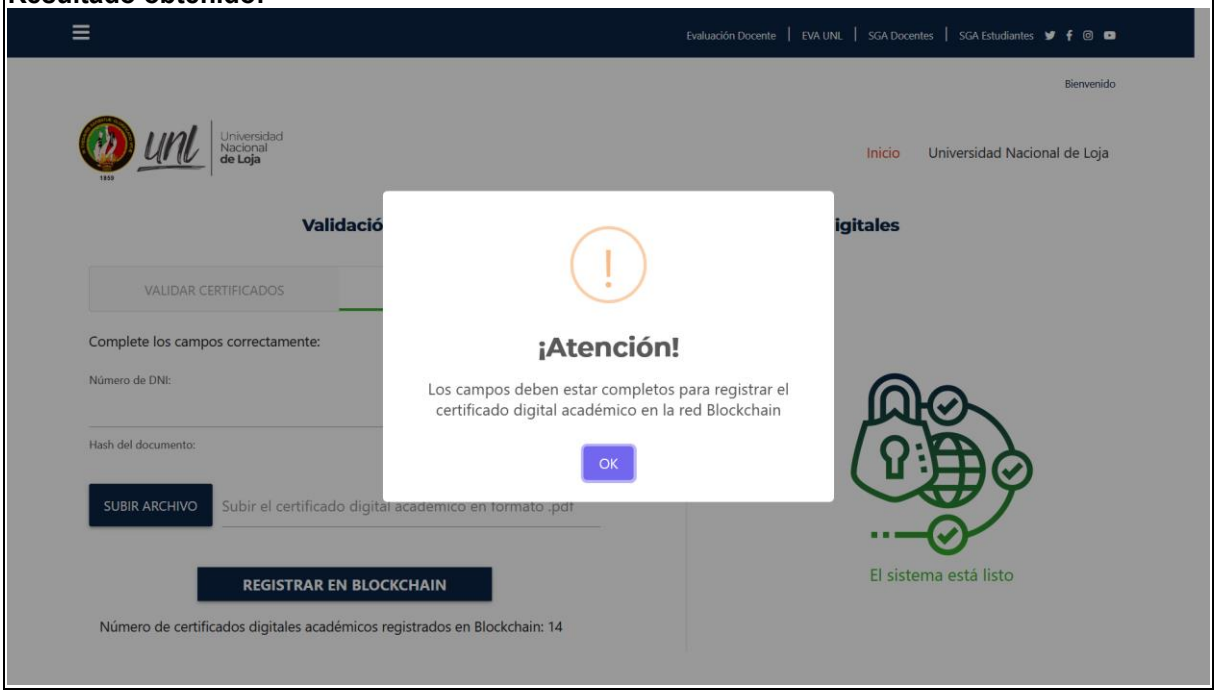


- Un mensaje de confirmación de registro con éxito.

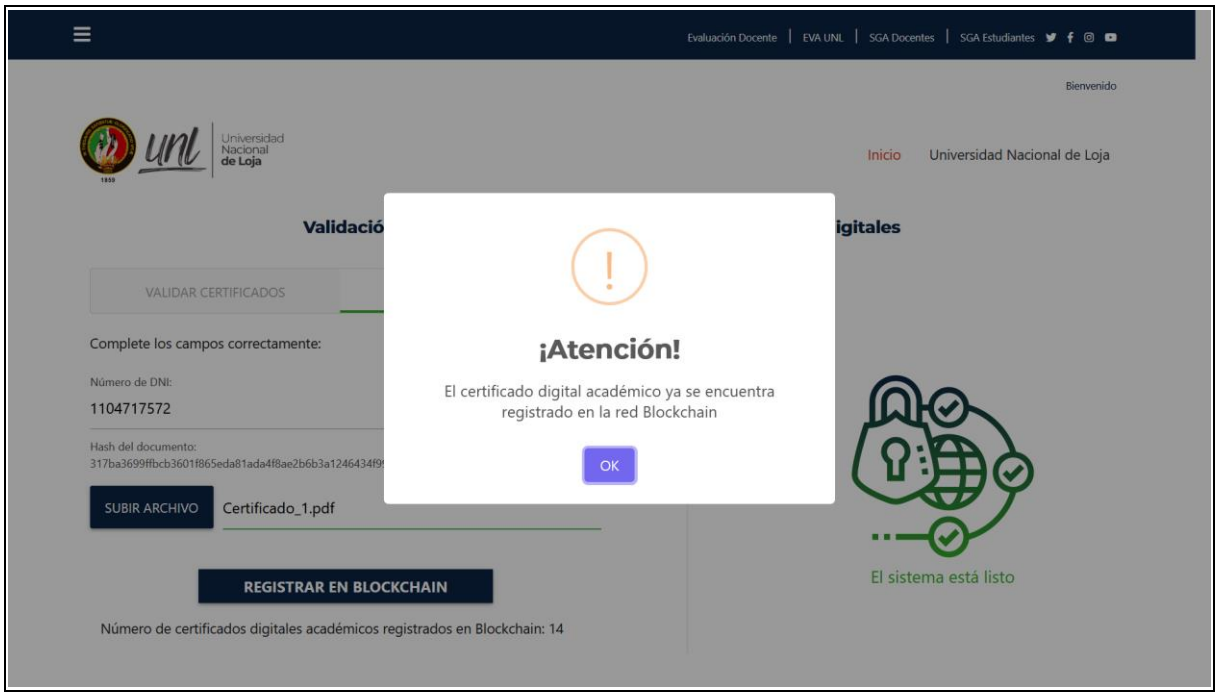


Registrar con campos en blanco	CP02	
	¿Prueba de despliegue?	Sí
Descripción: Se probará la respuesta del módulo de software al registrar, pero con los campos vacíos.		
Prerrequisitos • Ingresar a la página. • Inicio de sesión y conexión con la wallet de metamask (la contraseña de la wallet para el TT o frase de recuperación según corresponda).		
Pasos: 1. Hacer clic en el tab denominado “REGISTRAR CERTIFICADOS”. 2. No completar los campos de DNI o cargar el documento del CAD en formato PDF, a la vez. 3. Hacer clic en el botón “REGISTRAR EN BLOCKCHAIN”.		

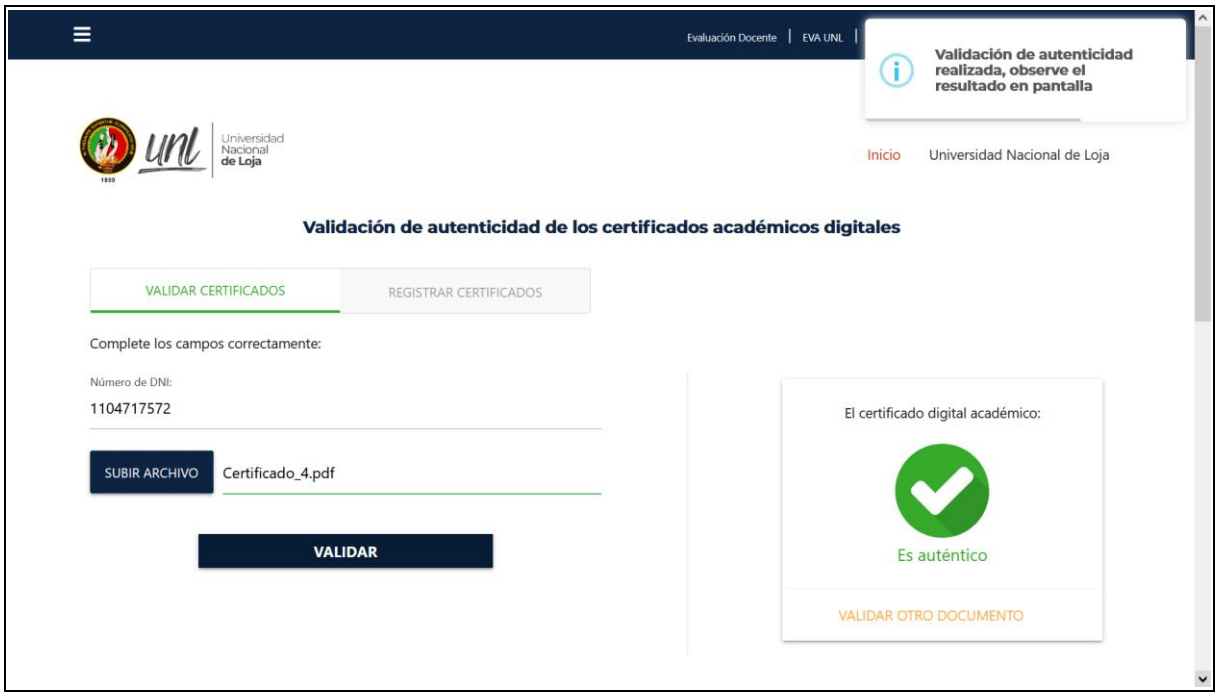
Resultado esperado: Un mensaje que indique que los campos no deben estar en blanco para registrar el certificado académico digital.
Resultado obtenido:



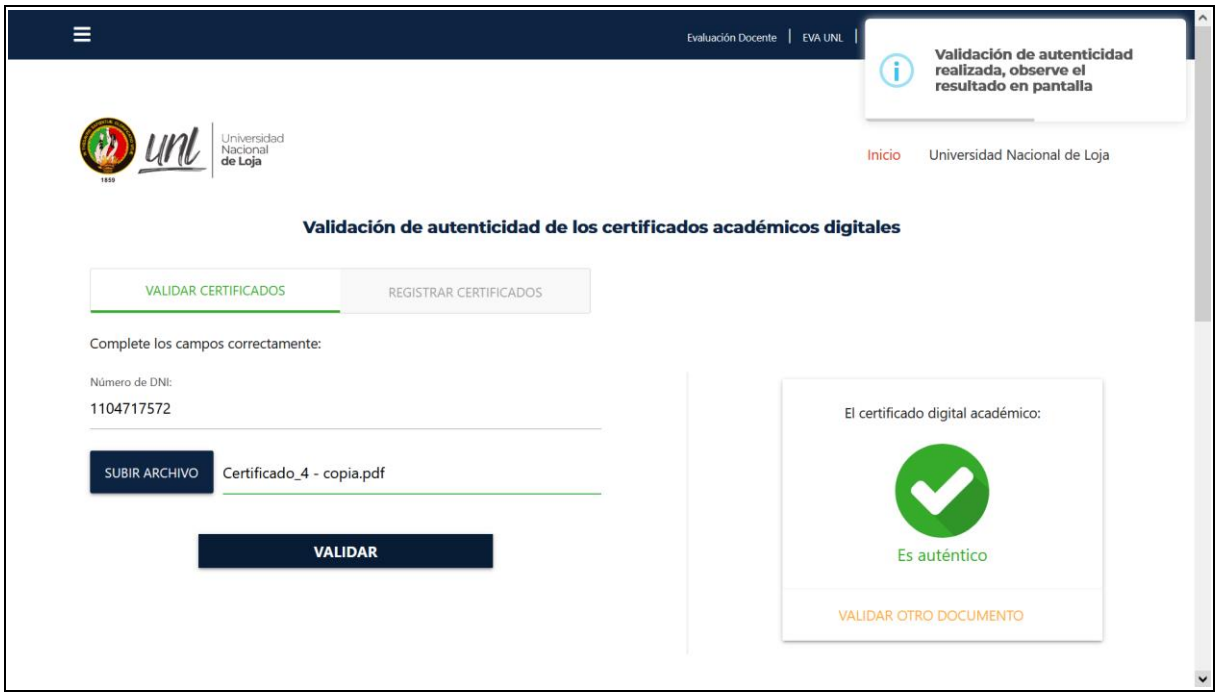
Registrar un CAD ya registrado	CP03	
	¿Prueba de despliegue?	Sí
Descripción: Se probará la respuesta del módulo de software al registrar un certificado académico digital ya registrado con anterioridad.		
Prerrequisitos • Ingresar a la página. • Inicio de sesión y conexión con la wallet de metamask (la contraseña de la wallet para el TT o frase de recuperación según corresponda).		
Pasos: 1. Hacer clic en el tab denominado “REGISTRAR CERTIFICADOS”. 2. Completar los campos de DNI y cargar el documento del CAD en formato PDF (ya registrado con anterioridad). 3. Hacer clic en el botón “REGISTRAR EN BLOCKCHAIN”.		
Resultado esperado: Un mensaje que indique que el certificado académico digital ya se encuentra registrado.		
Resultado obtenido:		



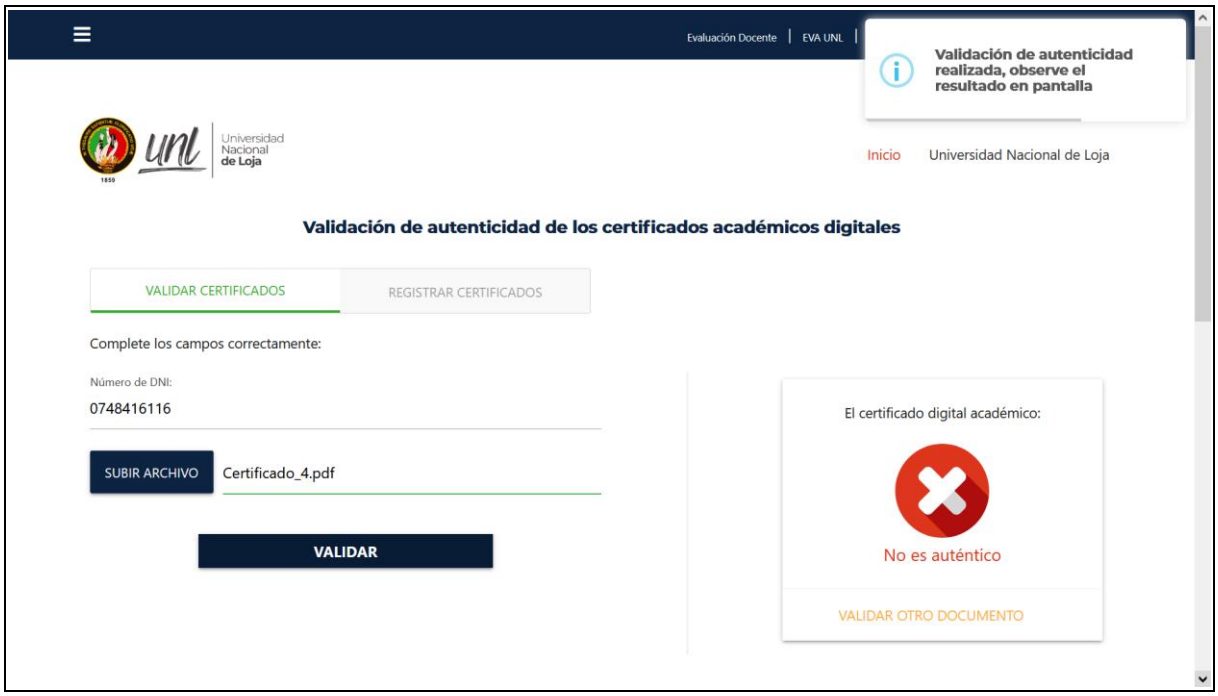
Validar con DNI correcto y CAD auténtico	CP04	
	¿Prueba de despliegue?	Sí
Descripción: Se probará la respuesta del módulo de software al validar el DNI correcto con un certificado académico digital auténtico en formato PDF.		
Prerrequisitos • Ingresar a la página. • Inicio de sesión y conexión con la wallet de metamask (la contraseña de la wallet para el TT o frase de recuperación según corresponda). • Tener registrados certificados académicos digitales.		
Pasos: 1. Hacer clic en el tab denominado “VALIDAR CERTIFICADOS”, aunque por defecto se encuentra en esa vista. 2. Completar los campos con el DNI correcto y cargar el documento del CAD en formato PDF. 3. Hacer clic en el botón “VALIDAR”.		
Resultado esperado: • Visualizar un mensaje indicando que la validación fue realizada. • Visualizar en pantalla junto al formulario el resultado de “El certificado académico digital: Es auténtico”.		
Resultado obtenido:		



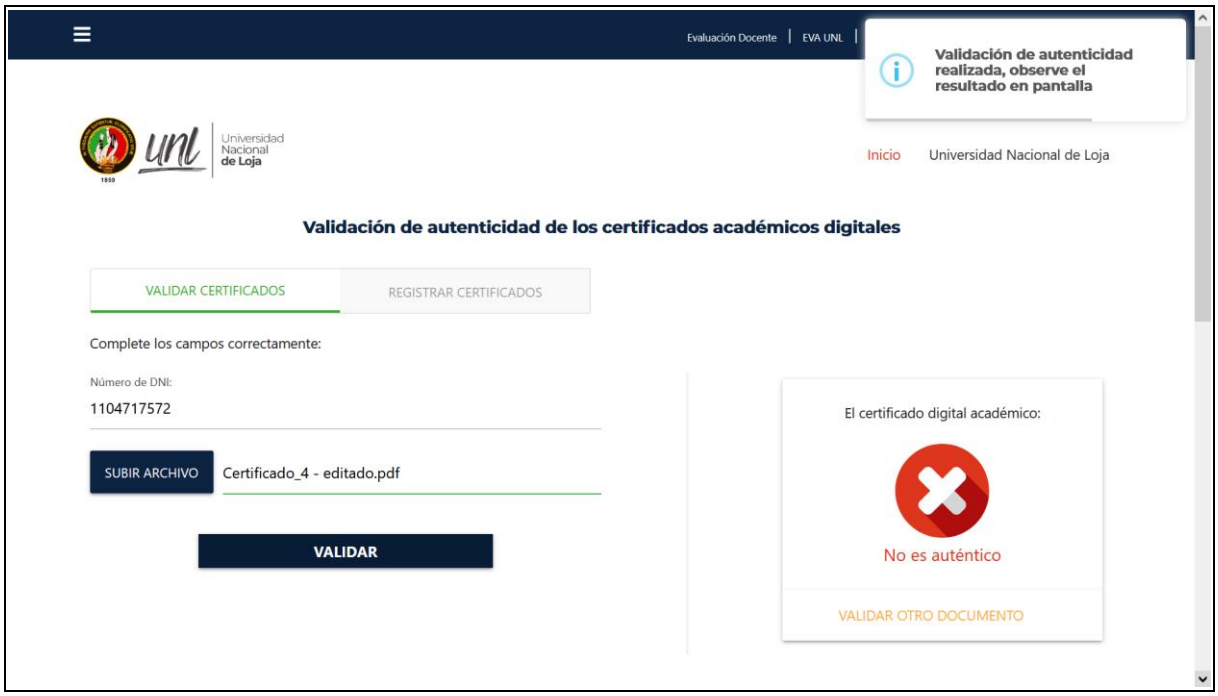
Validar con DNI correcto y copia del CAD auténtico	**CP05**	
	¿Prueba de despliegue?	**Sí**
Descripción: Se probará la respuesta del módulo de software al validar el DNI correcto con una copia del certificado académico digital auténtico en formato PDF.		
Prerrequisitos • Ingresar a la página. • Inicio de sesión y conexión con la wallet de metamask (la contraseña de la wallet para el TT o frase de recuperación según corresponda). • Tener registrados certificados académicos digitales.		
Pasos: 1. Hacer clic en el tab denominado “VALIDAR CERTIFICADOS”, aunque por defecto se encuentra en esa vista. 2. Completar los campos de DNI y cargar el documento de la copia del CAD autentico en formato PDF. 3. Hacer clic en el botón “VALIDAR”.		
Resultado esperado: • Visualizar un mensaje indicando que la validación fue realizada. • Visualizar en pantalla junto al formulario el resultado de “El certificado académico digital: Es auténtico”.		
Resultado obtenido:		



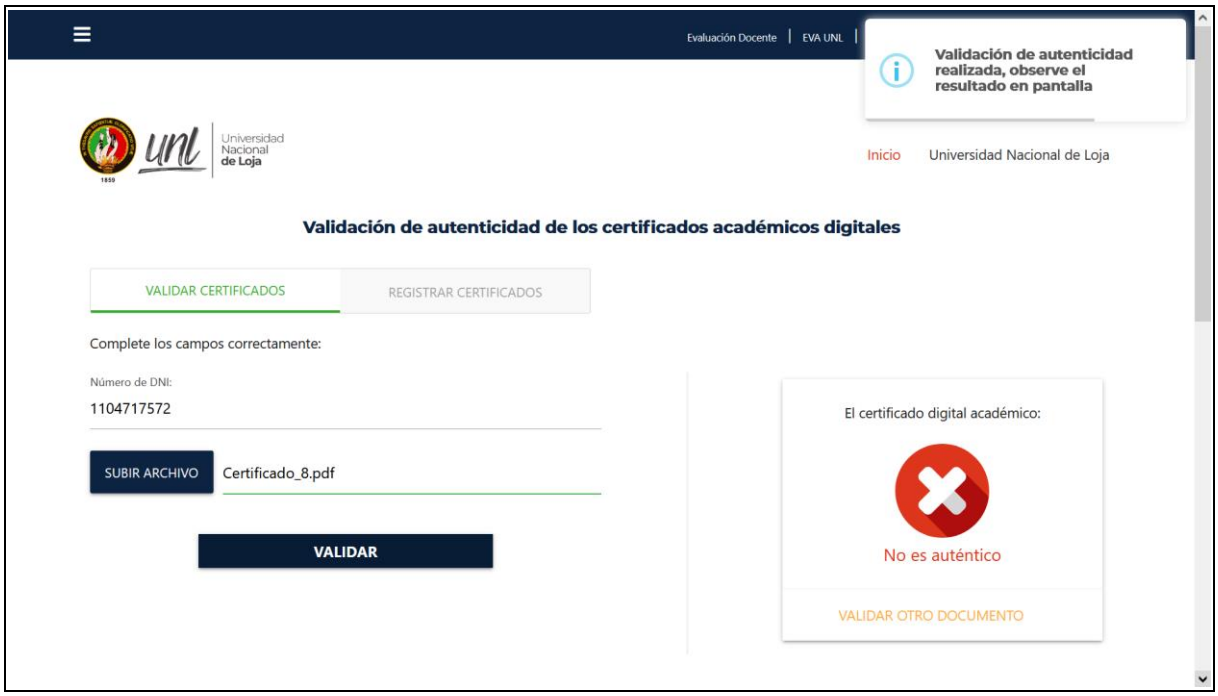
Validar con DNI incorrecto y CAD auténtico	**CP06**	
	¿Prueba de despliegue?	**Sí**
Descripción: Se probará la respuesta del módulo de software al validar el DNI incorrecto con un certificado académico digital auténtico en formato PDF.		
Prerrequisitos • Ingresar a la página. • Inicio de sesión y conexión con la wallet de metamask (la contraseña de la wallet para el TT o frase de recuperación según corresponda). • Tener registrados certificados académicos digitales.		
Pasos: 1. Hacer clic en el tab denominado “VALIDAR CERTIFICADOS”, aunque por defecto se encuentra en esa vista. 2. Completar los campos de DNI incorrecto y cargar el documento del CAD autentico en formato PDF. 3. Hacer clic en el botón “VALIDAR”.		
Resultado esperado: • Visualizar un mensaje indicando que la validación fue realizada. • Visualizar en pantalla junto al formulario el resultado de “El certificado académico digital: No es auténtico”.		
Resultado obtenido:		



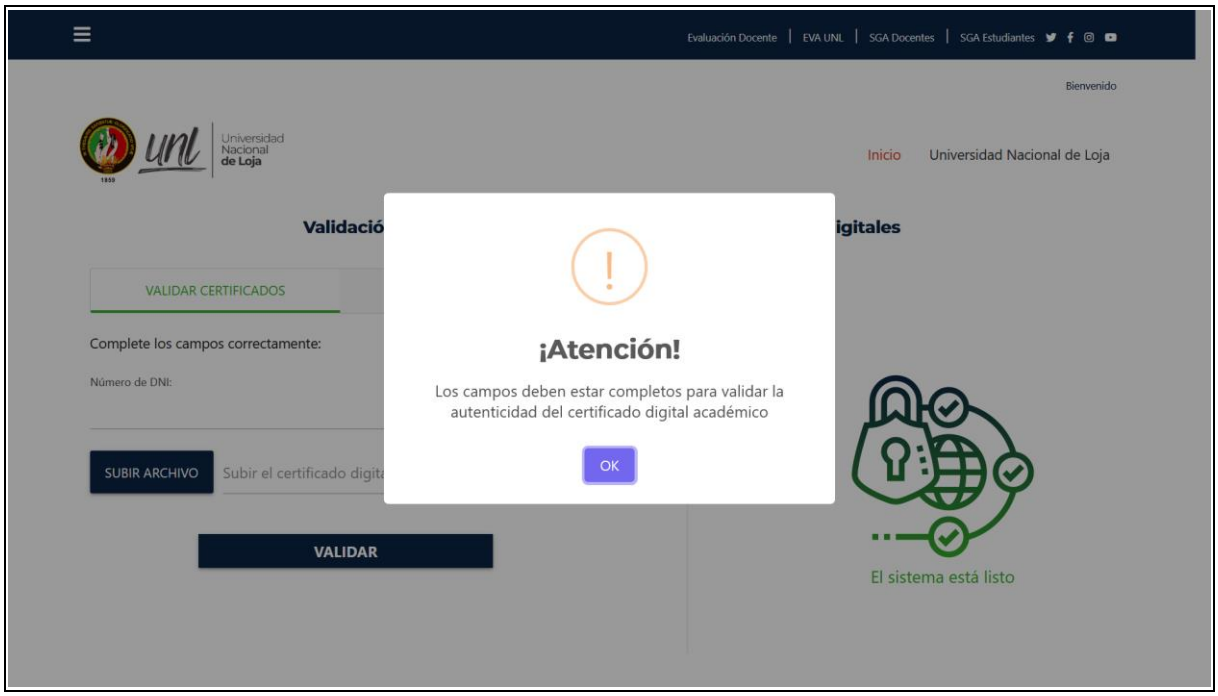
Validar con DNI correcto y CAD editado	CP07	
	¿Prueba de despliegue?	Sí
Descripción: Se probará la respuesta del módulo de software al validar el DNI correcto con un certificado académico digital editado en formato PDF.		
Prerrequisitos • Ingresar a la página. • Inicio de sesión y conexión con la wallet de metamask (la contraseña de la wallet para el TT o frase de recuperación según corresponda). • Tener registrados certificados académicos digitales.		
Pasos: 1. Hacer clic en el tab denominado “VALIDAR CERTIFICADOS”, aunque por defecto se encuentra en esa vista. 2. Completar los campos de DNI correcto y cargar el documento del CAD editado en formato PDF. 3. Hacer clic en el botón “VALIDAR”.		
Resultado esperado: • Visualizar un mensaje indicando que la validación fue realizada. • Visualizar en pantalla junto al formulario el resultado de “El certificado académico digital: No es auténtico”.		
Resultado obtenido:		



Validar con DNI correcto y CAD diferente al registrado	**CP08**	
	¿Prueba de despliegue?	**Sí**
Descripción: Se probará la respuesta del módulo de software al validar el DNI correcto con un certificado académico digital diferente al registrado en formato PDF.		
Prerrequisitos • Ingresar a la página. • Inicio de sesión y conexión con la wallet de metamask (la contraseña de la wallet para el TT o frase de recuperación según corresponda). • Tener registrados certificados académicos digitales.		
Pasos: 1. Hacer clic en el tab denominado “VALIDAR”, aunque por defecto se encuentra en esa vista. 2. Completar los campos de DNI correcto y cargar el documento del CAD diferente al registrado en formato PDF. 3. Hacer clic en el botón “VALIDAR”.		
Resultado esperado: • Visualizar un mensaje indicando que la validación fue realizada. • Visualizar en pantalla junto al formulario el resultado de “El certificado académico digital: No es auténtico”.		
Resultado obtenido:		



Validar con campos en blanco	CP09	
	¿Prueba de despliegue?	Sí
Descripción: Se probará la respuesta del módulo de software al validar, pero con los campos vacíos.		
Prerrequisitos • Ingresar a la página. • Inicio de sesión y conexión con la wallet de metamask (la contraseña de la wallet para el TT o frase de recuperación según corresponda).		
Pasos: 1. Hacer clic en el tab denominado “VALIDAR”, aunque por defecto se encuentra en esa vista. 2. No completar los campos de DNI o cargar el documento del CAD en formato PDF, a la vez. 3. Hacer clic en el botón “VALIDAR”.		
Resultado esperado: Un mensaje que indique que los campos no deben estar en blanco para validar el certificado académico digital.		
Resultado obtenido:		


Evaluación Docente | EVA UNL | SGA Docentes | SGA Estudiantes
Bienvenido
Universidad Nacional de Loja
Inicio
Universidad Nacional de Loja
VALIDAR CERTIFICADOS
Complete los campos correctamente:
Número de DNI:
SUBIR ARCHIVO
VALIDAR
¡Atención!
Los campos deben estar completos para validar la autenticidad del certificado digital académico
OK
El sistema está listo

4. Estrategia de ejecución de pruebas

En este apartado se indica los diferentes casos de prueba que se realizan en tres ciclos.

.

	Ciclo 1	Ciclo 2	Ciclo 3
CP01	x		
CP02			x
CP03	x		
CP04		x	
CP05		x	
CP06		x	
CP07		x	
CP08		x	
CP09			x

5. Anexos

A continuación, se adjunta capturas en las que se muestra que las diferentes transacciones en las que se puede observar su trazabilidad por medio de Etherscan de la testnet Rinkeby en la Figura 1. Las estadísticas de Infura con hora y fecha en las que se realizaron las pruebas en la Figura 2.

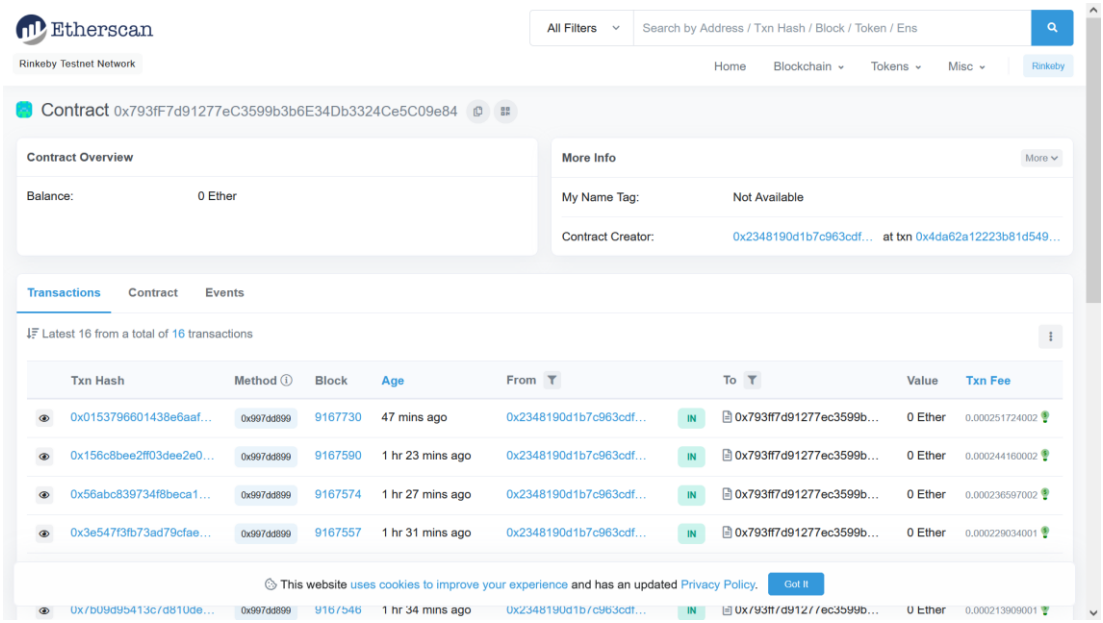


Figura 1. Trazabilidad del contrato inteligente en Etherscan Rinkeby.

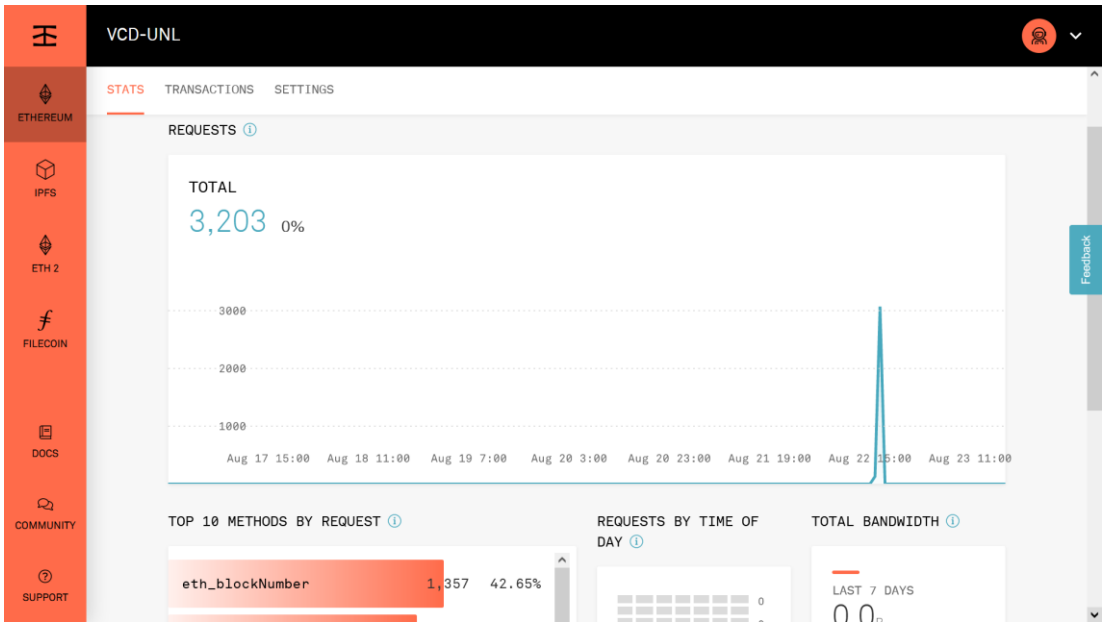


Figura 2. Estadística del uso de la testnet en Infura.

6. Glosario

A continuación, se muestra la definición de todos los términos utilizados en el presente documento.

Término	Descripción
CAD	Certificado académico digital
DNI	Documento nacional de identificación
PDF	Portable Document Format, «formato de documento portátil»
PTT	Proyecto de Trabajo de Titulación

7. Bibliografía y referencias

Referencia	Título
Ref. 1	Plantilla Plan de Pruebas Funcionales del portal del marco de desarrollo de software de la Junta de Andalucía (MADEJA)
Ref. 2	Documento de Especificación de Requisitos del Sistema

Anexo 11. Plan de Pruebas de Aceptación

Plan de Pruebas de Aceptación

Proyecto: Módulo de software para la validación de autenticidad de certificados académicos digitales por tecnología Blockchain

Versión: 2.0

Fecha: 14/09/2021

Hoja de control

Organismo	Universidad Nacional de Loja		
Proyecto	Módulo de software para la validación de certificados académicos digitales por tecnología blockchain		
Entregable	Planes de Pruebas de Aceptación		
Autor	Edgar Sánchez		
Versión/Edición	2.0	**Fecha Versión**	14/09/2021
Aprobado por	Cristian Ramiro Narváez Guillen, Mg.Sc.	**Fecha Aprobación**	15/09/2021
		Nº Total de Páginas	11

Registro de cambios

Versión doc	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Versión inicial del Plan de Pruebas de Aceptación	Edgar Patricio Sánchez Malla	24/08/2021
2.0	Versión final del Plan de Pruebas de Aceptación	Edgar Patricio Sánchez Malla	14/09/2021

Control de distribución

Nombre y Apellidos
Edgar Patricio Sánchez Malla
Cristian Ramiro Narváez Guillen, Mg.Sc

Índice

1. Introducción

Objeto

El objetivo de este documento es que a partir de la obtención de información (encuesta) que valida la funcionalidad y requerimientos del módulo de software por medio de la Aceptación de una muestra de personas de la Universidad Nacional de Loja.

Propósito

Validar que el módulo de software por una muestra de personas de la Universidad Nacional de Loja de la misma, mediante la aplicación de encuestas, verificar que cuenta con la respectiva Aceptación.

2. Parámetros de evaluación

Nro.	Pregunta
1	¿Es simple el vocabulario utilizado?
2	¿Se proporciona tiempo suficiente para realizar las entradas por teclado?
3	¿Se entienden la interfaz y su contenido?
4	¿Resulta fácil identificar un objeto o una acción?
5	¿Resulta fácil entender el resultado de una acción?
6	¿Está diseñada la interfaz para facilitar la realización eficiente de las tareas de la mejor forma posible?
7	¿Son apropiados los mensajes presentados por el sistema?
8	¿Actúa el sistema en la prevención de errores?
9	¿El sistema informa claramente sobre los errores presentados?
10	¿Permite una cómoda navegación dentro del producto y una fácil salida de éste?
11	¿Se presenta al usuario la información que sólo necesita?

Así mismo, se plantea una tabla para determinar el estado de las Pruebas de Aceptación:

Estado de las Pruebas de Aceptación	Criterio		
	Sí	Parcialmente	No
Positivo	>= 80%	<= 20%	<1%
Negativo	< 80%	> 20%	>1%

3. Encuesta

La presente encuesta tiene por objetivo conocer si el módulo de Software desarrollado cumple con la aceptación de funcionamiento y de requerimientos por parte de una muestra de personas de 62 estudiantes de la Universidad Nacional de Loja, específicamente de la carrera de Ingeniería en Sistemas. Misma que se realizó por medio de los formularios del Google Suite de la Universidad, como se muestra en la Figura 1.

En el que se obtuvo los siguientes resultados que se representan en los gráficos generados automáticamente de Google Form.

Figura 1. Construcción de la encuesta en Google Form.

En la encuesta están contemplada los parámetros de evaluación, realizada a un grupo de estudiantes de 62 personas. Adicionalmente se agregaron las preguntas de nombres y apellidos; número de cedula; a que parte de la comunidad estudiantil pertenece y el correo institucional para mayor fiabilidad de las respuestas. A continuación, se detallan los resultados para cada parámetro convertido en pregunta.

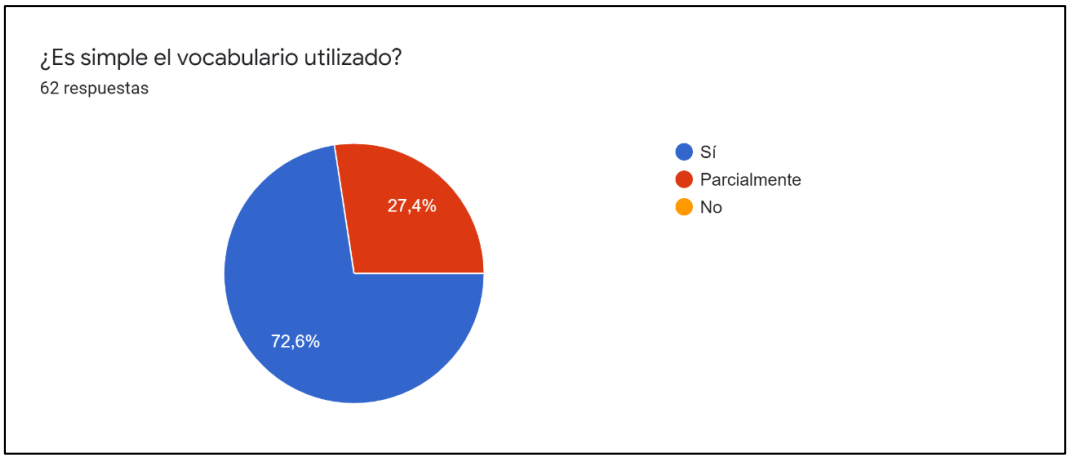


Figura 2. Parámetro a evaluar número 1.

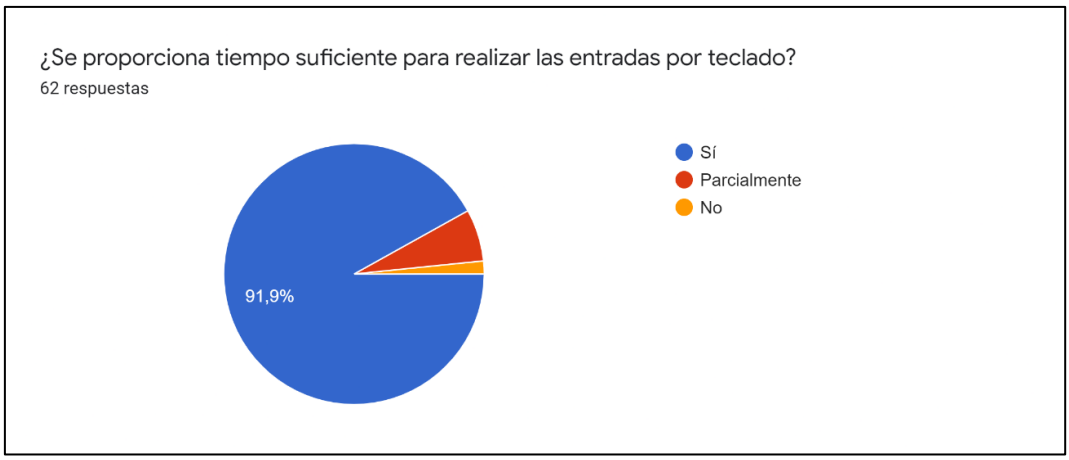


Figura 3. Parámetro a evaluar número 2.

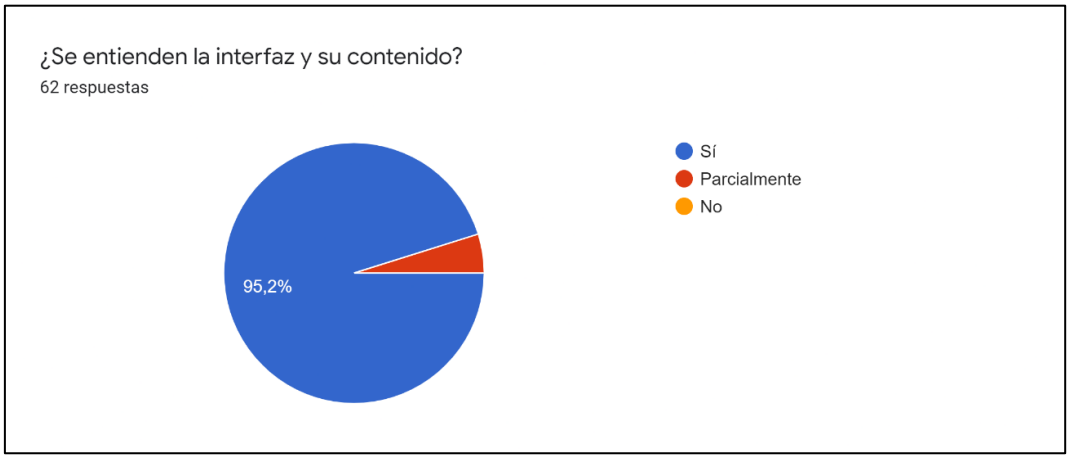


Figura 4. Parámetro a evaluar número 3.

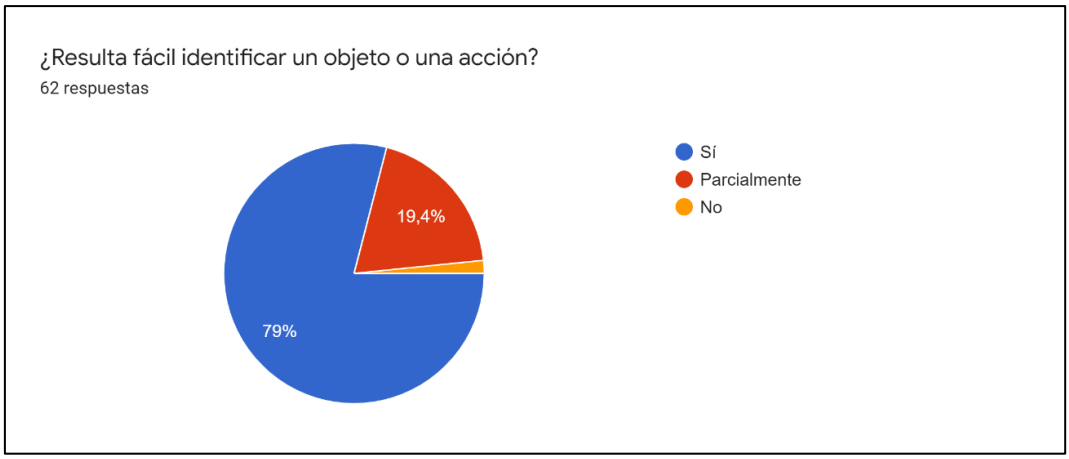


Figura 5. Parámetro a evaluar número 4.

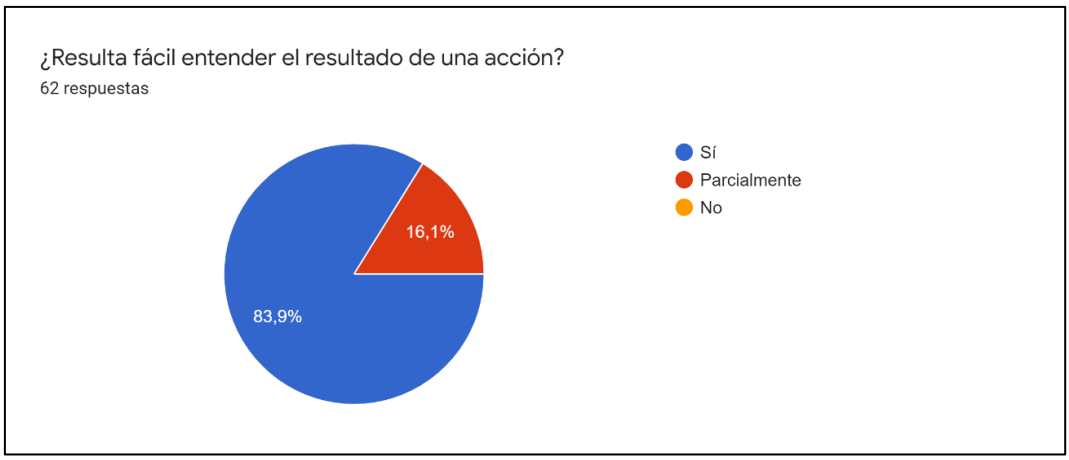


Figura 6. Parámetro a evaluar número 5.

Figura 7. Parámetro a evaluar número 6.

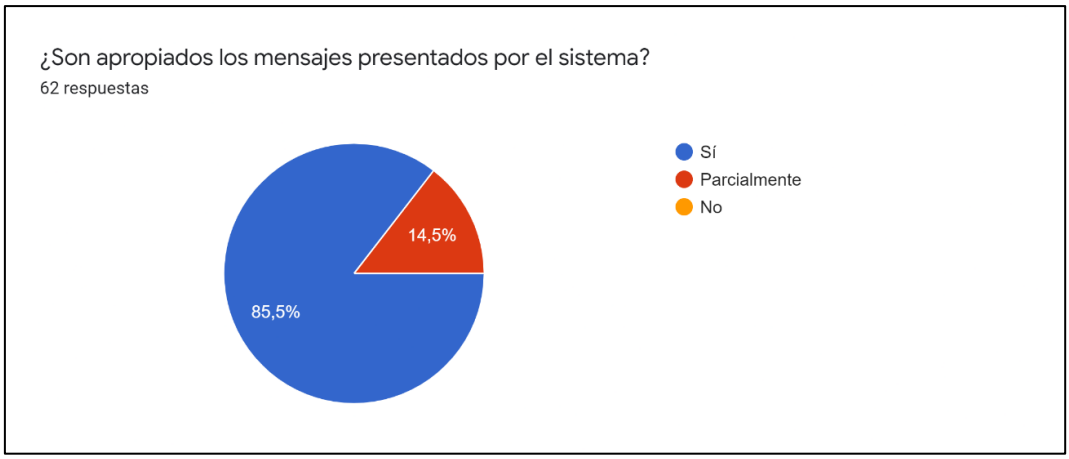


Figura 8. Parámetro a evaluar número 7.

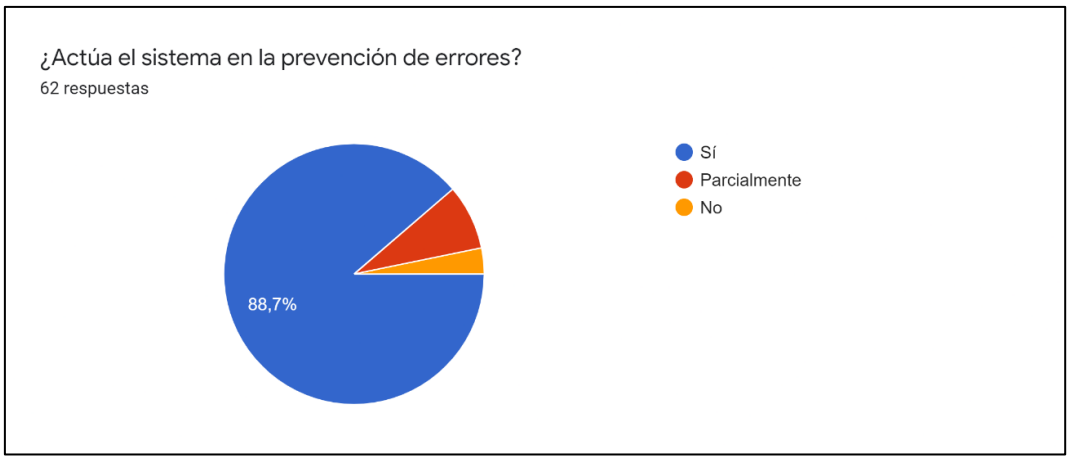


Figura 9. Parámetro a evaluar número 8.

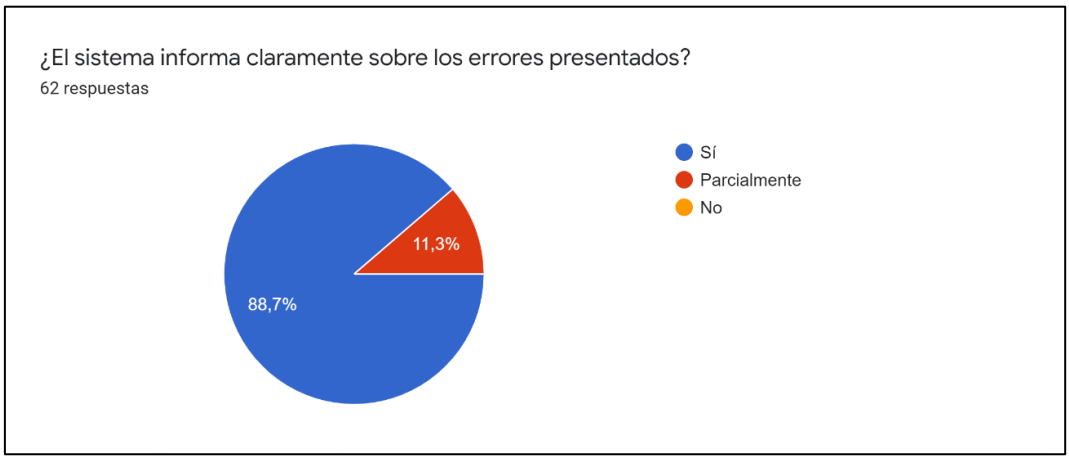


Figura 10. Parámetro a evaluar número 9.

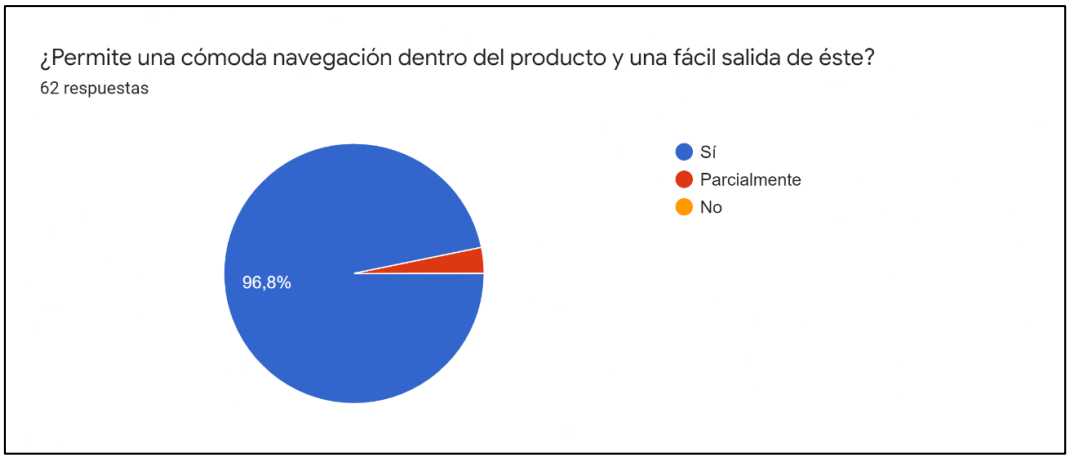


Figura 11. Parámetro a evaluar número 10.

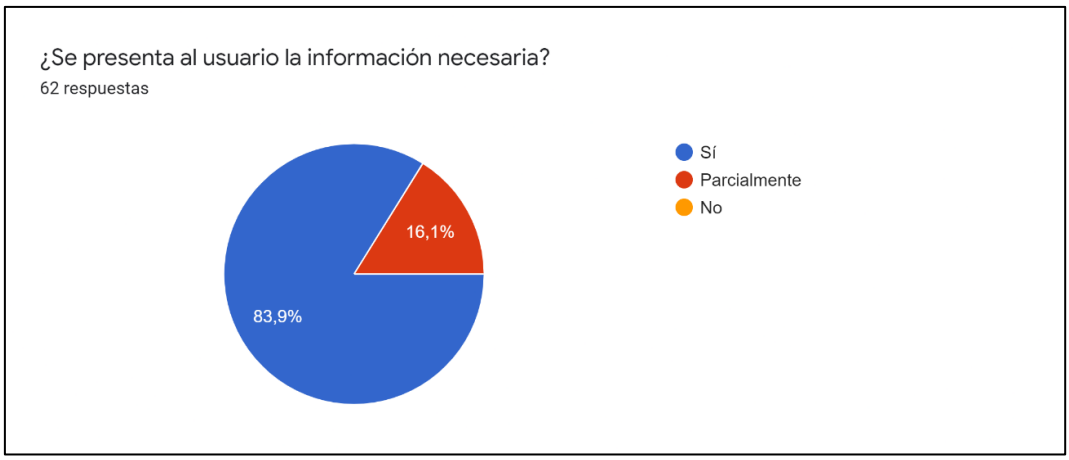


Figura 12. Parámetro a evaluar número 11.

4. Bibliografía y referencias

Referencia	Título
Anexo 1 (del Documento del Trabajo de Titulación)	Especificación de requisitos de Software IEEE 830.

Anexo 12. Manual de Usuario

Manual de Usuario

Proyecto: Módulo de software para la validación de autenticidad de certificados académicos digitales por tecnología Blockchain

Versión: 2.0

Fecha: 11/10/2021

Hoja de control

Organismo	Universidad Nacional de Loja		
Proyecto	Módulo de software para la validación de certificados académicos digitales por tecnología blockchain		
Entregable	Manual de Usuario		
Autor	Edgar Sánchez		
Versión/Edición	2.0	**Fecha Versión**	11/10/2021
Aprobado por	Cristian Ramiro Narváez Guillen, Mg.Sc.	**Fecha Aprobación**	12/10/2021
		Nº Total de Páginas	16

Registro de cambios

Versión doc	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Versión inicial del Manual de Usuario	Edgar Patricio Sánchez Malla	07/10/2021
2.0	Versión final del Manual de Usuario	Edgar Patricio Sánchez Malla	11/10/2021

Control de distribución

Nombre y Apellidos
Edgar Patricio Sánchez Malla
Cristian Ramiro Narváez Guillen, Mg.Sc

Índice

1. Introducción

El módulo de software permite registrar y validar la autenticidad de los certificados académicos digitales de forma correcta a través de tecnología Blockchain. Por lo tanto, se procede a genera un manual de usuario.

2. Requisitos

El módulo de software para la validación de autenticidad de certificados académicos digitales por tecnología Blockchain permite su funcionamiento en los navegadores:

- Chrome.
- Mozilla Firefox.
- Brave.
- Navegador de la aplicación móvil Metamask.

Para los navegadores Chrome, Mozilla Firefox y Brave, instalar la extensión MetaMask y crear o iniciar sesión en una cuenta. Además de obtener fondos a través de una Ethereum Rinkeby Faucet Testnet, más información de cómo realizarlo aquí: https://medium.com/@julgq/c%C3%B3mo-enviar-eth-a-metamask-en-la-red-rinkeby-f3bbf388ba54

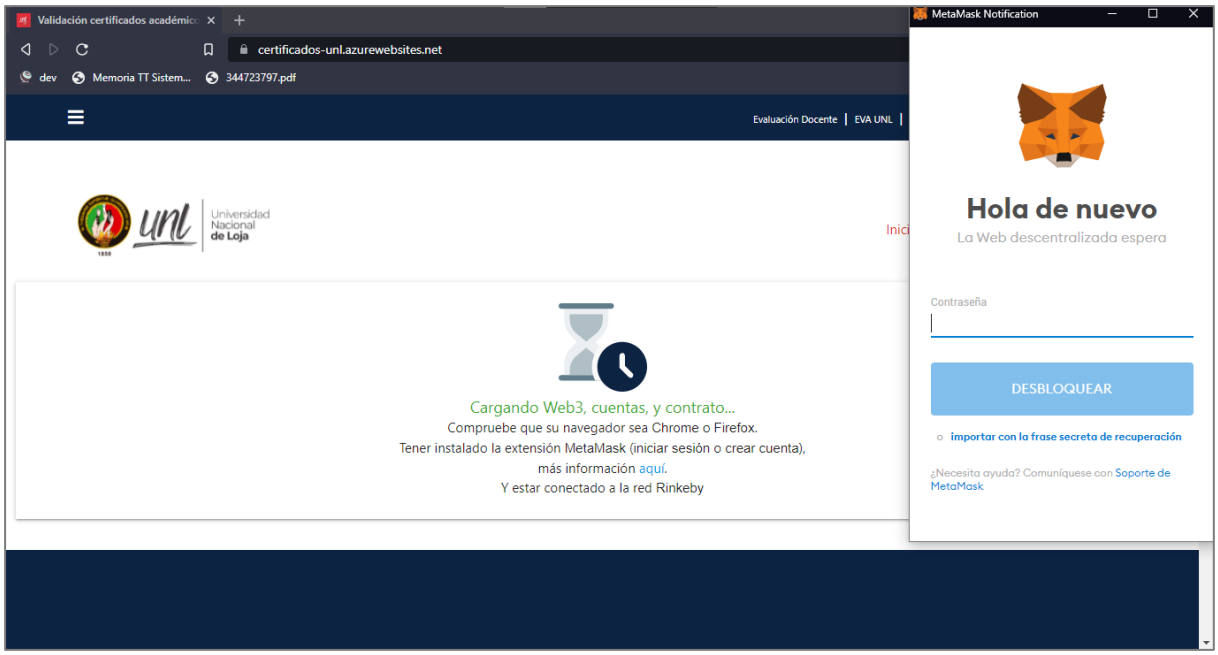
En el caso de la aplicación móvil se debe iniciar sesión una vez creada ya la cuenta.

NOTA: Es importante tener fondos para poder realizar las transacciones.

3. Ingreso al Sistema

A través de un navegador web:

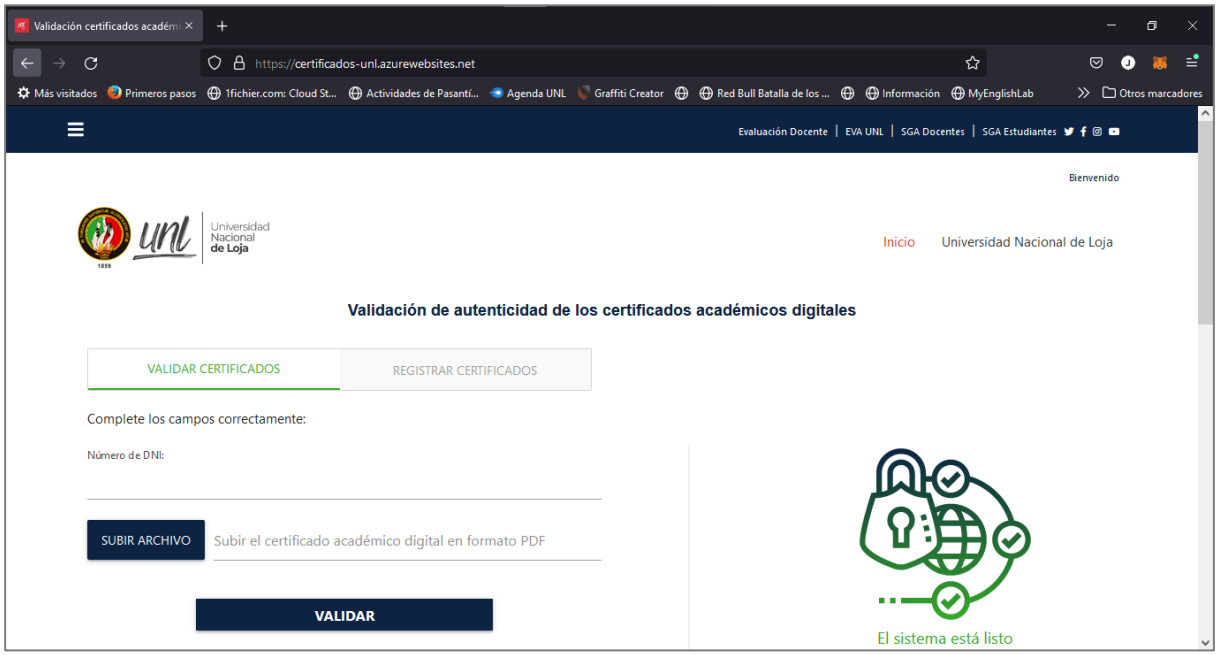
1. En el navegador seleccionado e instalada la extensión, debe ingresar a la siguiente dirección web: https://certificados-unl.azurewebsites.net/
2. Al cargar la página automáticamente se le lanzará una venta de la extensión MetaMask, como se observa en la siguiente imagen.



En la ventana de la extensión de MetaMask, ingresar la contraseña o importar con la frase de recuperación según sea el caso.

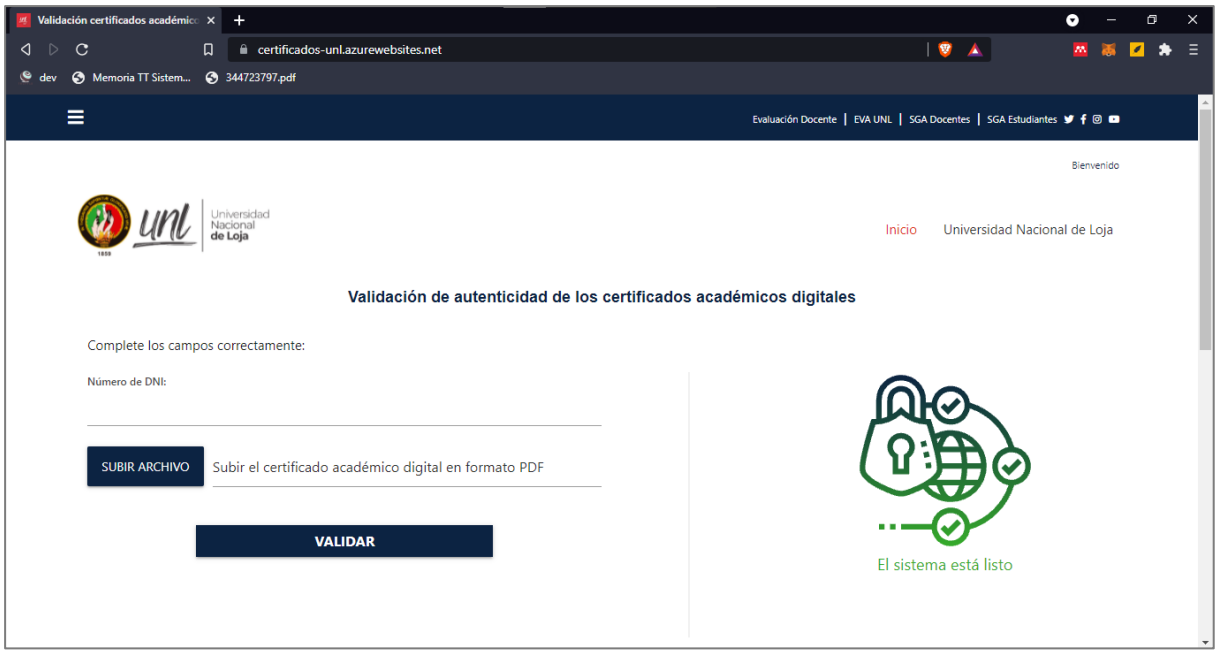
3. Una vez ingresado correctamente su contraseña se presentarán dos tipos de vistas según su cuenta, que le permitirá validar, registrar o ambos:

   3.1. Si puede registrar y validar un certificado académico digital, la siguiente vista:

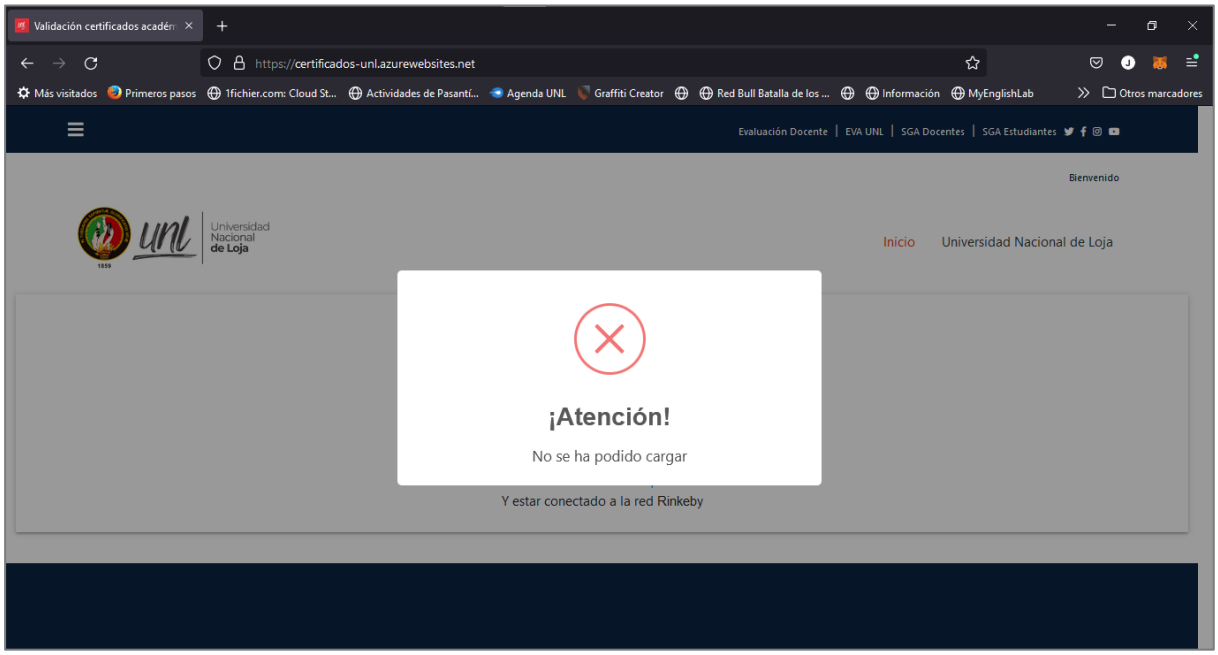


Se observa dos opciones que le permiten navegar entre los formularios de validar y registrar.

   3.2. Si puede validar un certificado académico digital, la siguiente vista:

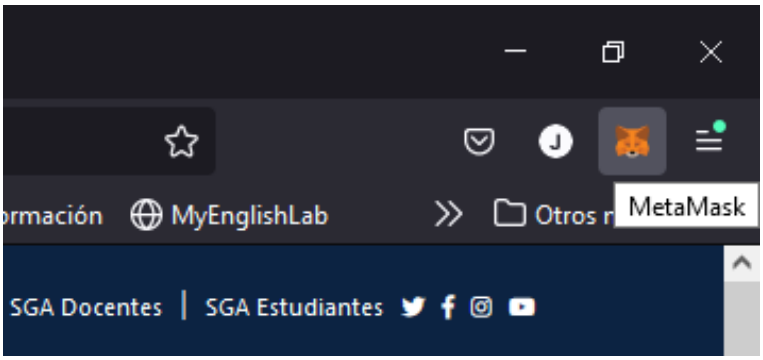


NOTA: Es importante en los dos casos que MetaMask esté conectado a la red de pruebas **Rinkeby** o si no se le presentará el siguiente mensaje de alerta.

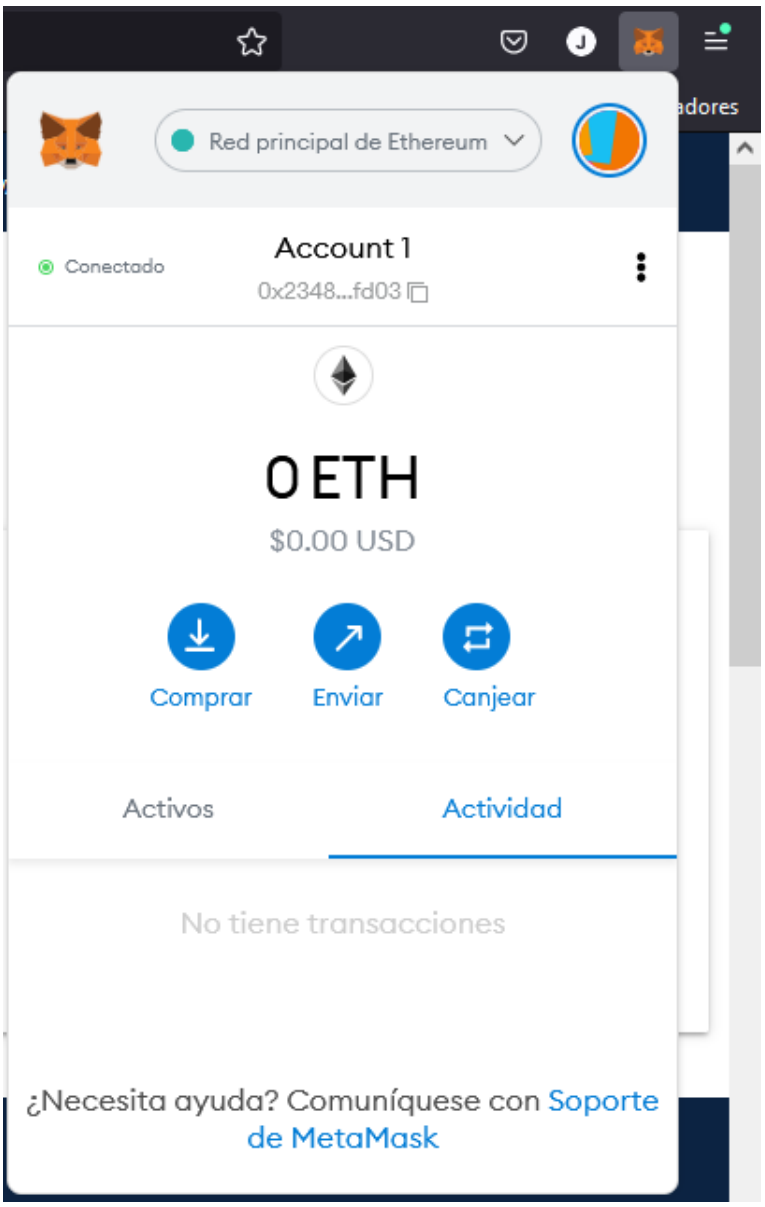


Para solucionar el siguiente error, hay que realizar lo siguiente:

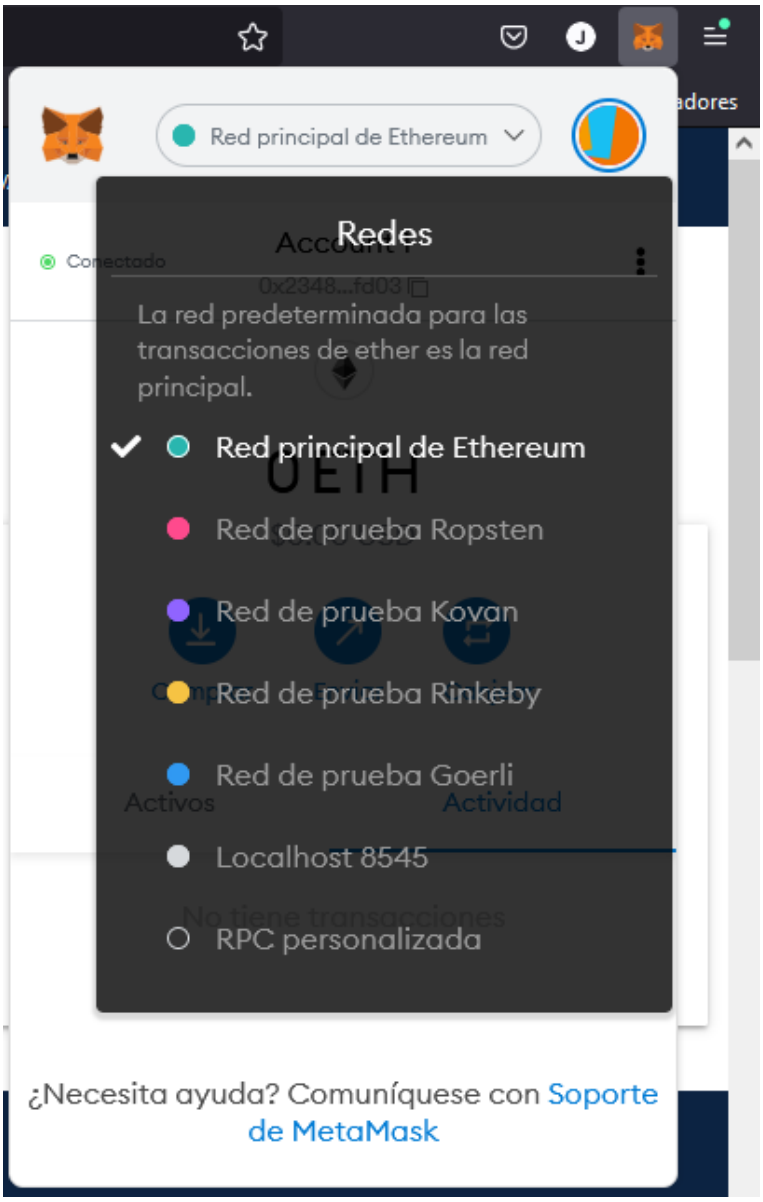
- Hacer clic sobre el icono  de la extensión de Metamask que, según el navegador, en este caso se puede observar en la parte superior derecha.



Y se despliega la siguiente ventana:



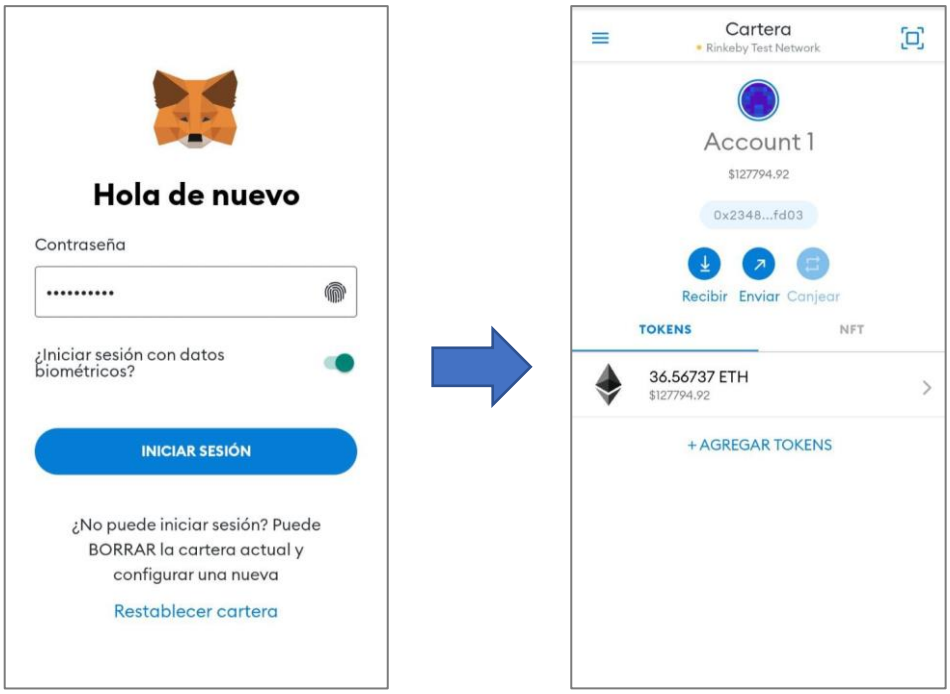
- Se hace clic sobre  y se despliega las diferentes redes de la siguiente forma:



- Aquí se debe seleccionar  y recargar la página.

A través de la app móvil de MetaMask:

1. Abrir la aplicación móvil e iniciar sesión, a través de su huella digital (si tiene activa esta opción) o a través de su contraseña, como se observa:

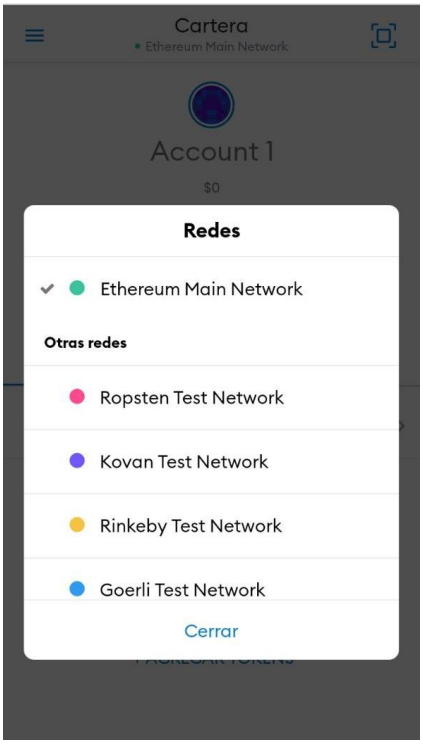


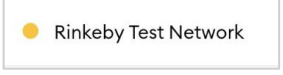
NOTA: Es importante verificar que, en la parte superior debajo del título de Cartera, este conectado a **testnet Rinkeby**, con el fin de evitar un error a posterior.

Si no se encuentra así realizar lo siguiente:

- Pulsar sobre 

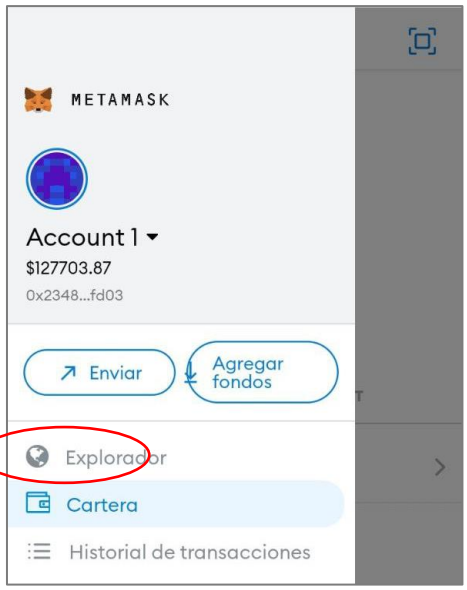
y se mostrará lo siguiente:



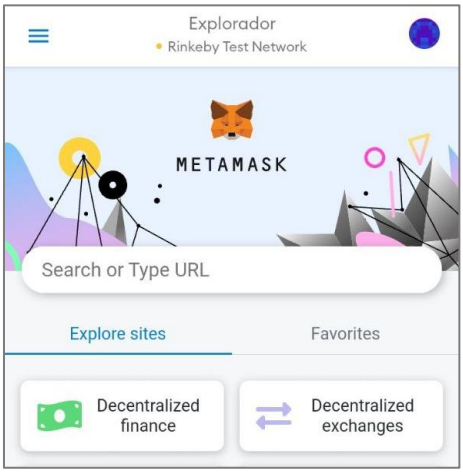
- Seleccionar 

y listo.

2. Pulsar sobre ☰ se le expande un menú y pulsar la opción “**Explorador**”:



3. En el buscador “Search or Type URL”, ingresar a la siguiente dirección web: https://certificados-unl.azurewebsites.net/



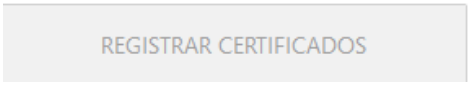
4. Al cargar la página se presentarán dos tipos de vistas según su cuenta, que le permitirá validar, registrar o ambos.

Si puede registrar y validar un certificado académico digital, la vista de la izquierda (Se observa dos opciones que le permiten navegar entre los formularios de validar y registrar. Si puede validar un certificado académico digital, la vista de la derecha:

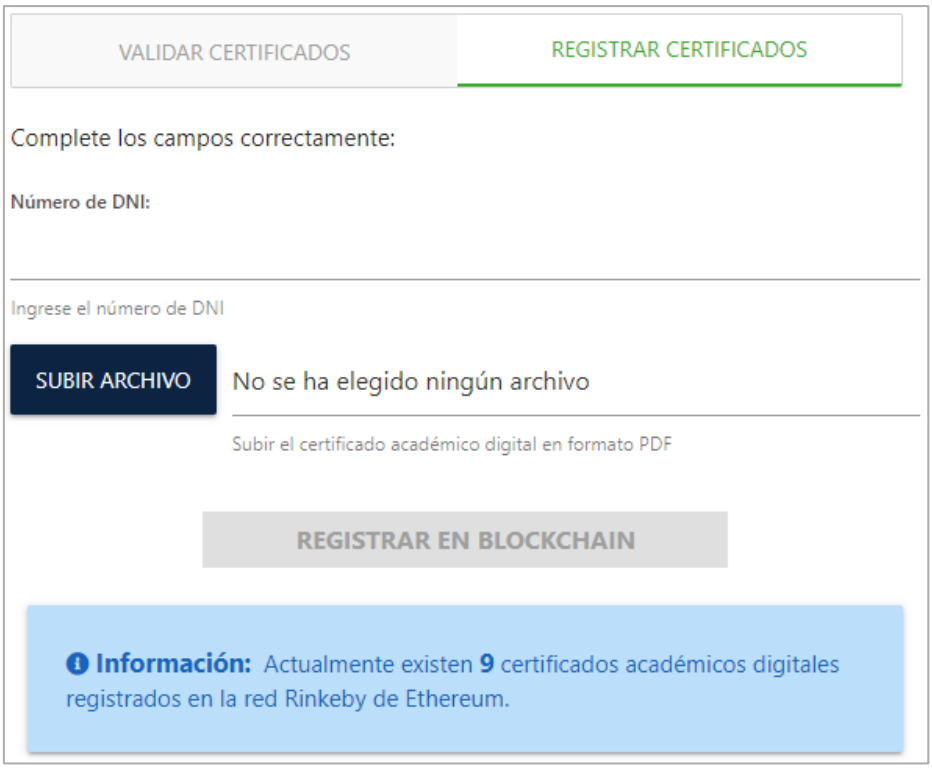
NOTA: En los siguientes puntos se indicará como registrar y validar un certificado académico digital, aunque el ejemplo esta realizado en un navegador web, a partir de aquí su funcionamiento es igual en la aplicación móvil.

4. Registrar un certificado académico digital

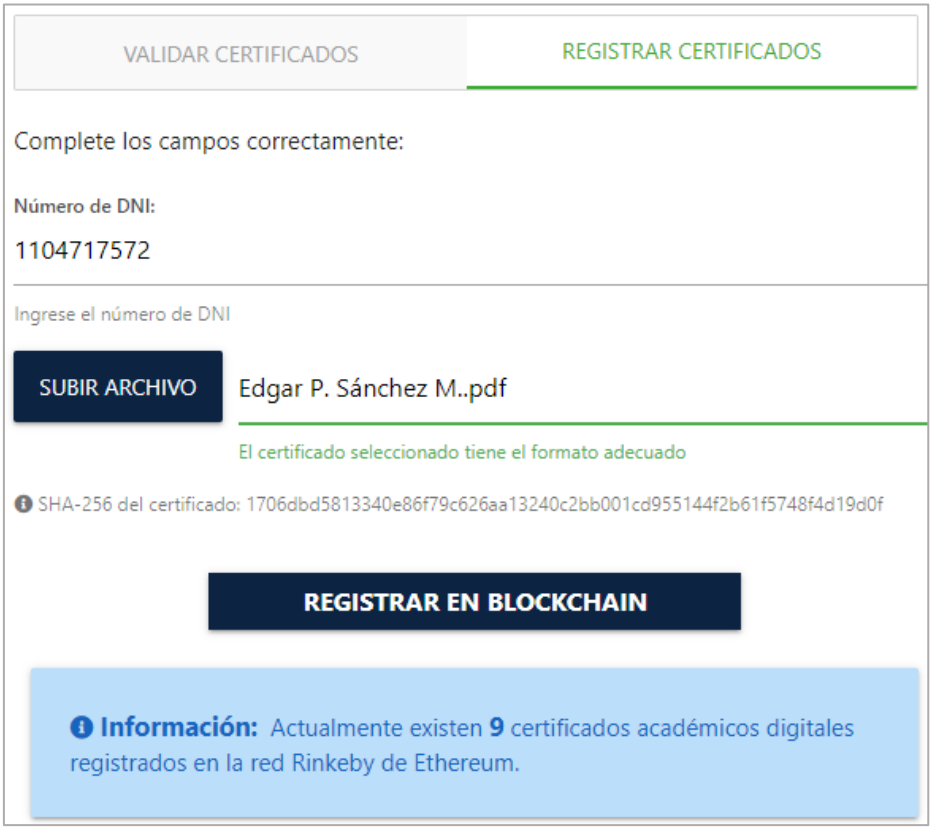
Esta funcionalidad nos permite registrar certificados académicos digitales en la red Blockchain de Ethereum de la testnet Rinkeby.

1. Hacer clic sobre 

y se observa el siguiente formulario:

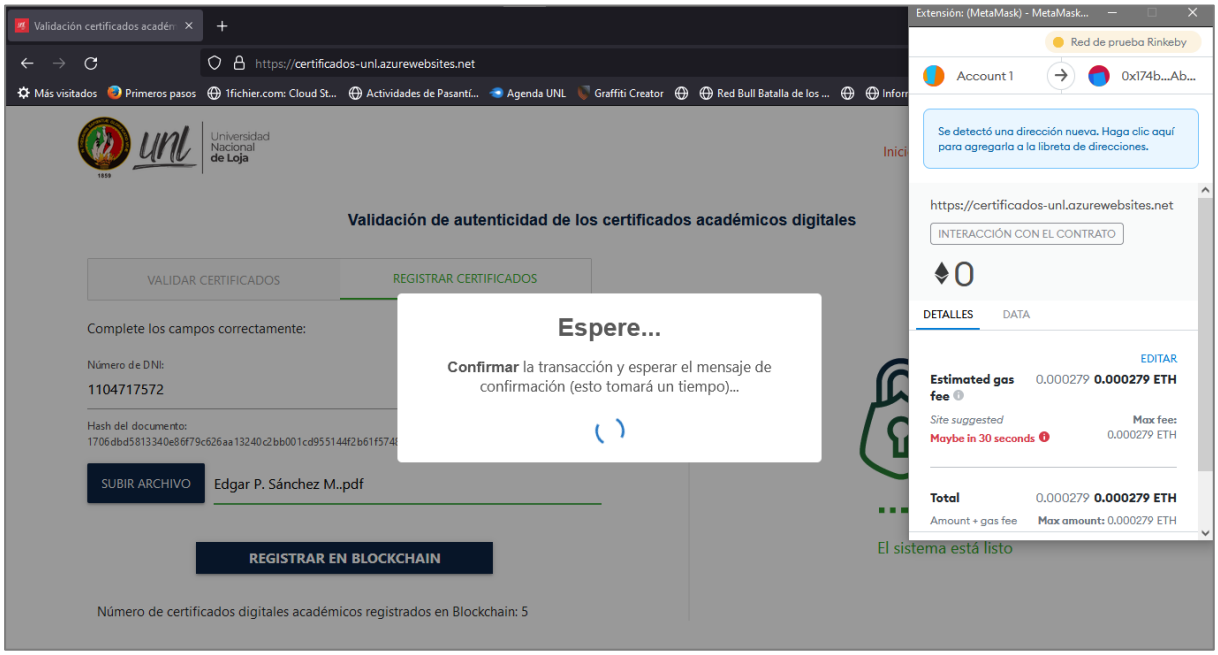


2. Completar los campos de Número de DNI y subir el archivo en formato PDF del certificado académico digital. Se observa el resultado así:



NOTA: El botón de **REGISTRAR EN BLOCKCHAIN** se activará cuando los campos están completos adecuadamente.

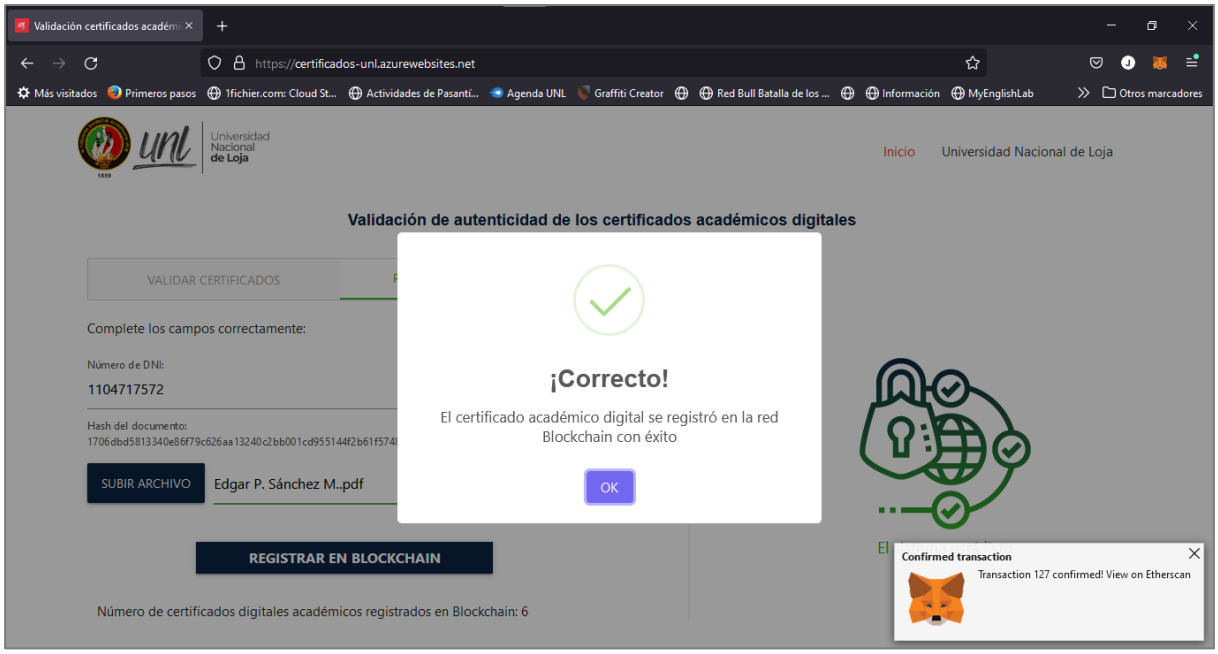
3. Hacer clic sobre **REGISTRAR EN BLOCKCHAIN** se muestra lo siguiente:



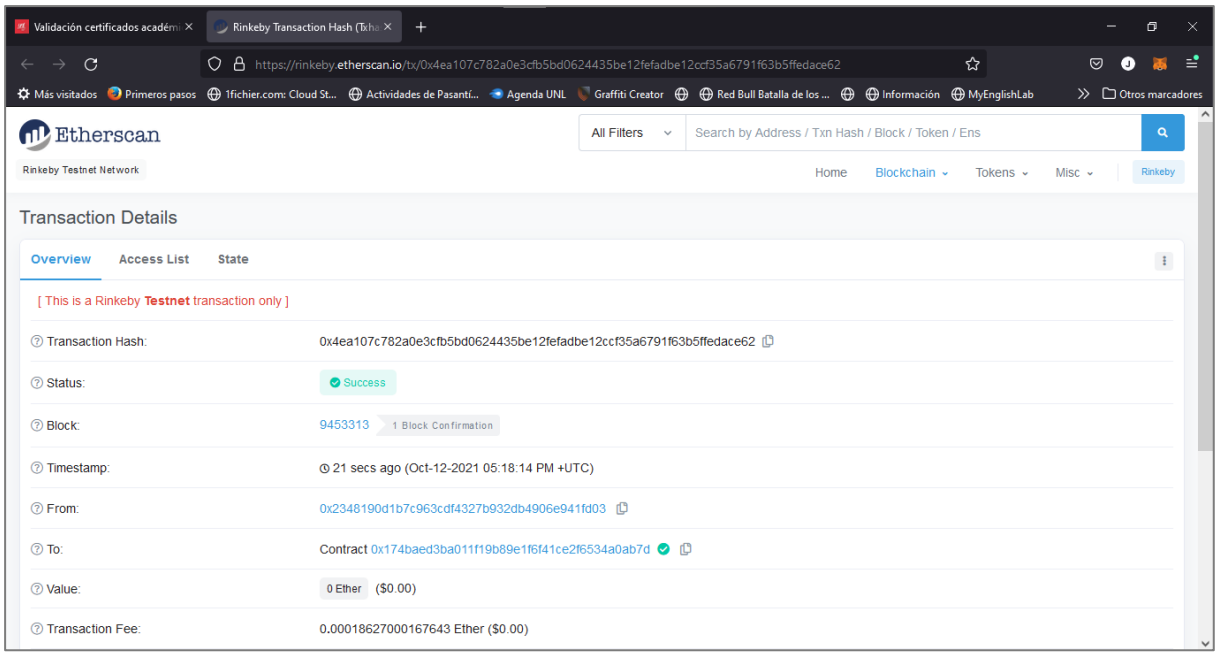
4. En la ventana de la extensión de MetaMask desplazarse hasta el final y hacer clic sobre

NOTA: Es Importante que la cuenta tenga fondos o sino el botón no se habilitará, para obtener fondos, se explica en el punto de **Requisitos** de este documento.

5. Se debe esperar hasta obtener el siguiente mensaje de confirmación:



Al hacer clic sobre  te redirige a una ventana de tu navegador donde se observa todos los detalles de la transacción:



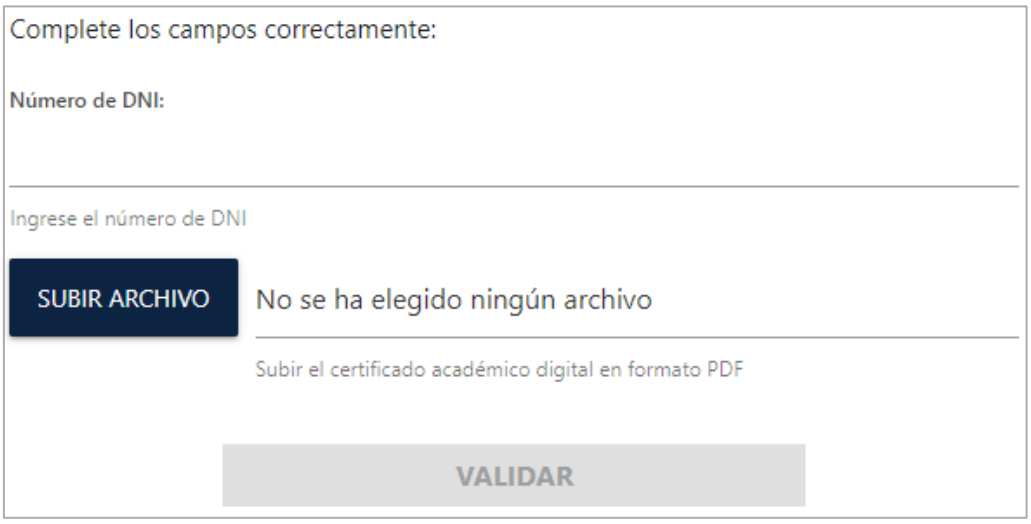
5. Validar un certificado académico digital

Esta funcionalidad nos permite validar la autenticidad de certificados académicos digitales en la red Blockchain de Ethereum de la testnet Rinkeby.

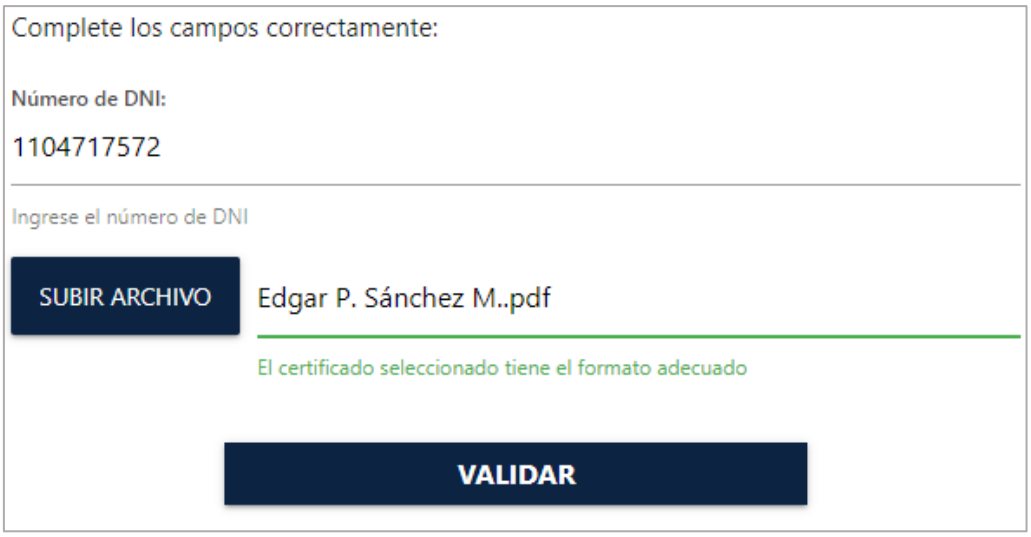
1. Si al ingresar al sistema se le muestra lo siguiente:

NOTA: En caso de no tener este componente, continuar al paso 2.

Asegurarse que este sobre la opción **VALIDAR CERTIFICADO**S y se le mostrará el siguiente formulario:



2. Llenar los campos de Número de DNI y cargar el certificado académico digital en formato PDF de la siguiente manera:

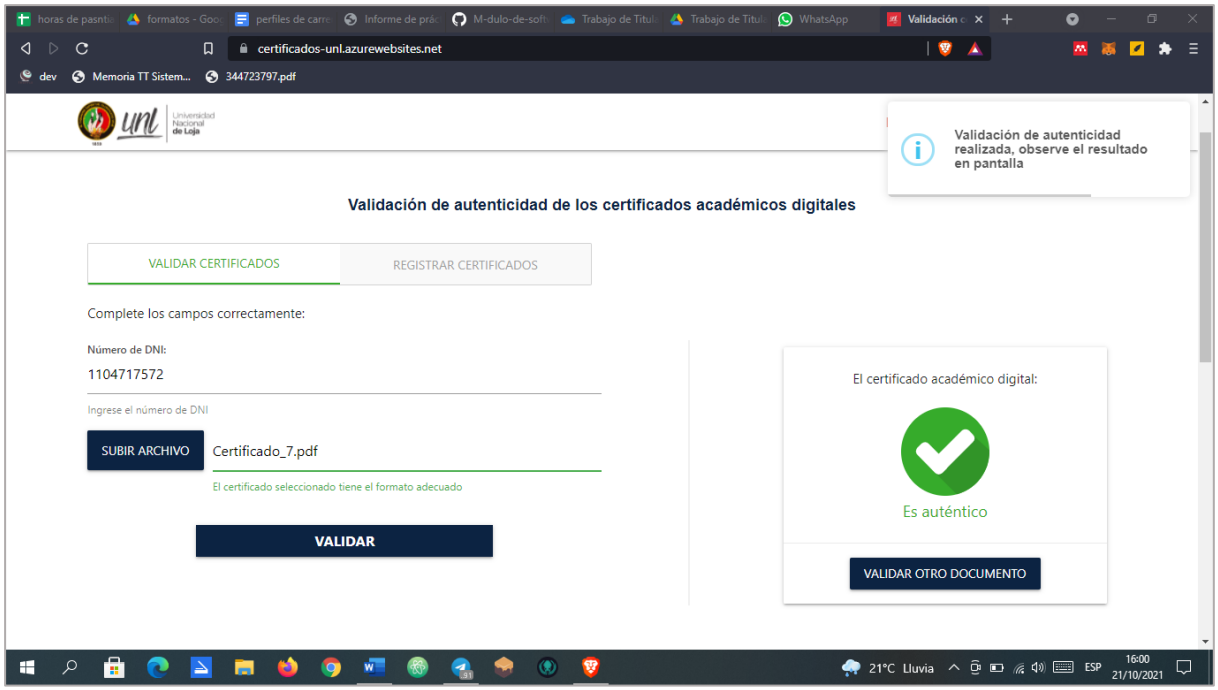


NOTA: el botón de **VALIDAR** se activará si los campos están completos adecuadamente.

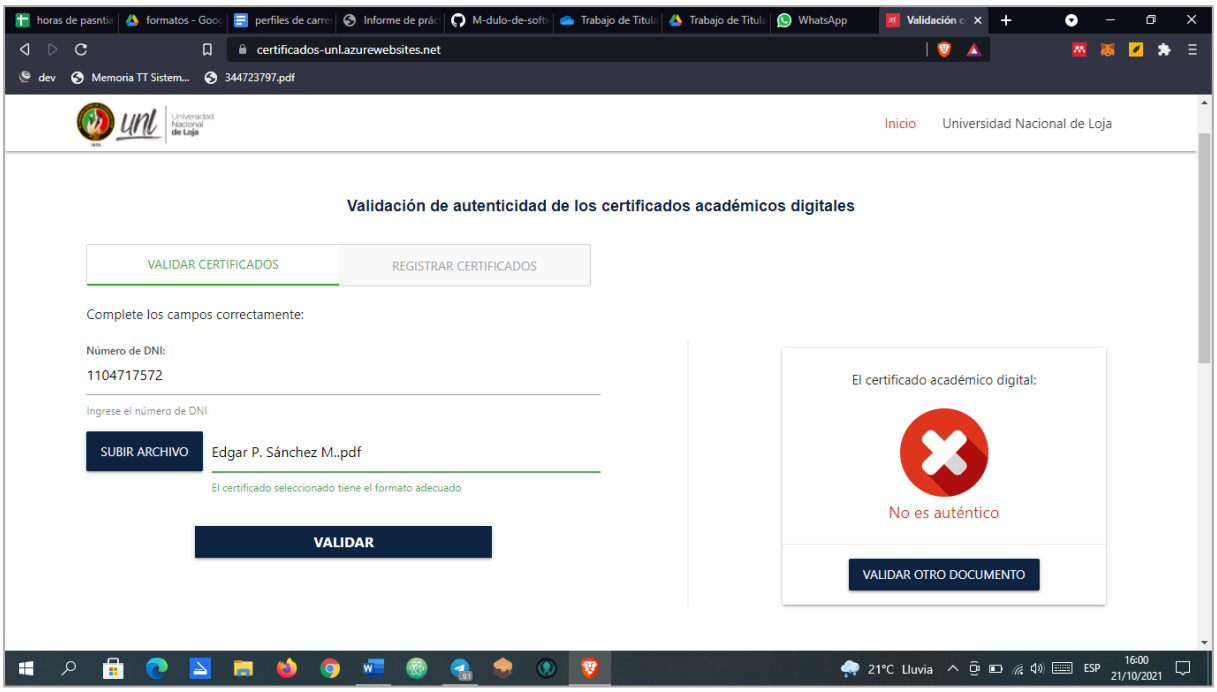
3. Hacer clic sobre 

y se le mostrará el resultado según los siguientes casos:

- Si es autentico la siguiente vista:



- Si NO es autentico la siguiente vista:


certificados-unl.azurewebsites.net
Inicio
Universidad Nacional de Loja
Validación de autenticidad de los certificados académicos digitales
VALIDAR CERTIFICADOS
REGISTRAR CERTIFICADOS
Complete los campos correctamente:
Número de DNI:
1104717572
Ingrese el número de DNI
SUBIR ARCHIVO
Edgar P. Sánchez M..pdf
El certificado seleccionado tiene el formato adecuado
VALIDAR
El certificado académico digital:
No es auténtico
VALIDAR OTRO DOCUMENTO

Anexo 13. Certificado de traducción

CERTIFICADO DE TRADUCCIÓN

El Sr. Manuel Alejandro Poma Tacuri, identificado con número de cédula 1105661100, Licenciado en Ciencias de la Educación mención Idioma Inglés.

CERTIFICA:

Que el texto traducido al idioma inglés que compone el **Resumen** del Trabajo de Titulación denominado: "**Implementación de la tecnología Blockchain para la validación de autenticidad de los certificados académicos digitales / Blockchain technology implementation for the authenticity validation of digital academic certificates**" correspondiente a la **Sr. Edgar Patricio Sánchez Malla**, con número de cédula **1104717572**, fue realizado y verificado bajo mi supervisión.

Eso es todo en cuanto puedo indicar en honor a la verdad, facultando al interesado hacer uso del presente documento para los fines que crea pertinentes.

Loja, 15 de diciembre de 2021

.....

Lic. Manuel Alejandro Poma Tacuri

C.I. 1105661100

Celular: 0939830288