“Desarrollo de un sistema de validación de documentos electrónicos basado en Blockchain

Caso de estudio: certificados educativos”

Rivadeneira Maldonado, Mauro Alejandro

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Sistemas e Informática

Trabajo de titulación, previo a la obtención del título de Ingeniero en Sistemas e Informática

Ing. Villacís Silva, César Javier. MsC.

14 de septiembre del 2021

Curiginal

Document Information

Analyzed document	TESIS RIVADENEIRA MALDONADO MAURO ALEJANDRO (2).pdf (D112593281)
Submitted	9/14/2021 11:04:00 PM
Submitted by	
Submitter email	biblioteca@espe.edu.ec
Similarity	4%
Analysis address	ilbbioteca.GDC@analysis.urkund.com

Firmado electrónicamente por:
CESAR JAVIER VILLACIS

.....

Ing. Villacís Silva, César Javier. MsC.

C.C: 1704892726

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

CERTIFICACIÓN

Certifico que el trabajo de titulación, "**Desarrollo de un sistema de validación de documentos electrónicos basado en Blockchain. Caso de estudio: certificados educativos**" fue realizado por el señor **Rivadeneira Maldonado, Mauro Alejandro** el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto, cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 14 de septiembre del 2021

Firmado electrónicamente por:
CESAR JAVIER VILLACIS

.....

Ing. Villacís Silva, César Javier. MsC.

C. C 1704892726

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

RESPONSABILIDAD DE AUTORÍA

Yo, **Rivadeneira Maldonado, Mauro Alejandro**, con cédula de ciudadanía n° 1723502231, declaro que el contenido, ideas y criterios del trabajo de titulación: **"Desarrollo de un sistema de validación de documentos electrónicos basado en Blockchain. Caso de estudio: certificados educativos"** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 9 de septiembre del 2021

.....

Rivadeneira Maldonado, Mauro Alejandro

C.C.: 1723502231

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

AUTORIZACIÓN DE PUBLICACIÓN

Yo **Rivadeneira Maldonado, Mauro Alejandro** con cédula de ciudadanía n° 1723502231, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **"Desarrollo de un sistema de validación de documentos electrónicos basado en Blockchain. Caso de estudio: certificados educativos"** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 9 de septiembre del 2021

.....

Rivadeneira Maldonado, Mauro Alejandro

C.C.: 1723502231

DEDICATORIA

El presente trabajo está dedicado a mis padres, quienes siempre me dieron su apoyo en las buenas y en las malas y supieron estar ahí desde el inicio de mi carrera universitaria.

A mi hermana, quien me inspiró para poder alcanzar objetivos que no creí posibles

A mis tíos, que no me dejaron solo en este proceso a pesar de estar pasando por momentos difíciles.

A mis abuelos, que siempre fueron un ejemplo de lucha y resistencia para toda la familia.

A mis amigos, quienes me han aportado su valioso conocimiento para convertirme en un mejor profesional cada día.

Mauro Alejandro Rivadeneira Maldonado.

AGRADECIMIENTO

A Dios por privilegiarme de una gran familia, de salud y de fuerza para seguir adelante día a día.

A la Universidad de las Fuerzas Armadas – ESPE, que ha sabido ser mi alma mater, acogiéndome durante años como mi segundo hogar y en donde pasé momentos duros que me fortalecieron, pero así también donde pasé algunos de los mejores días de mi vida.

A mi familia que me dio el impulso y la motivación para convertirme en una mejor persona.

Mauro Alejandro Rivadeneira Maldonado.

Índice de contenido

Índice de tablas

Índice de figuras

RESUMEN

El surgimiento en los últimos años de las nuevas tecnologías que trae la industria 4.0 trae consigo nuevos paradigmas sobre el funcionamiento de las tecnologías de la información, así como poderosas herramientas que deben comenzar a ser aprovechadas. En el presente trabajo se busca encaminar una de estas soluciones como es Blockchain, una red inmutable que no depende de intermediarios, hacia el sector de la educación. Más específicamente a la validación de certificados educativos, que cada año implican costos de almacenamiento, transporte e impresión a gran cantidad de instituciones educativas, además del peligro que conllevan cantidad de certificados falsos que circulan, algo que implica un obstáculo para los estudiantes que desean mostrar su conocimiento adquirido en búsqueda de empleos, becas de estudios, etc. Por ello se plantea una solución capaz de mantener la integridad de estos certificados educativos y validarlos de manera confiable. Información que además se encontrará disponible dentro de un sistema de archivos descentralizado.

Palabras clave:

- **BLOCKCHAIN**
- **CERTIFICADOS EDUCATIVOS**
- **ETHEREUM**
- **OPEN BADGES**
- **VALIDACIÓN DE DOCUMENTOS ELECTRÓNICOS**

ABSTRACT

The emergence in recent years of new technologies brought by Industry 4.0 brings with it new paradigms on the operation of information technologies, as well as powerful tools that must begin to be used. The present work seeks to direct one of these solutions such as Blockchain, an immutable network that does not depend on intermediaries, towards the education sector. More specifically to the validation of educational certificates, which each year entail storage, transport and printing costs for a large number of educational institutions, in addition to the danger posed by the amount of false certificates that circulate, something that implies an obstacle for students who wish to show your knowledge acquired in search of jobs, scholarships, etc. Therefore, a solution capable of maintaining the integrity of these educational certificates and validating them in a reliable way is proposed. Information that will also be available within a decentralized file system.

Keywords:

- **BLOCKCHAIN**
- **EDUCATIONAL CERTIFICATES**
- **ETHEREUM**
- **OPEN BADGES**
- **VALIDATION OF ELECTRONIC DOCUMENTS**

Capítulo I

Introducción

En el presente capítulo se describe los antecedentes, la problemática, los objetivos planteados y la hipótesis de este trabajo de titulación.

Antecedentes

En la última década las nuevas tendencias tecnológicas y en conjunto con la próxima llegada de la cuarta revolución industrial o industria 4.0, traen consigo la necesidad de adoptar nuevos paradigmas tecnológicos, así como nuevas herramientas para automatizar una gran cantidad de procesos (Chávez, 2018).

Sin embargo, a su vez también crece la exigencia de contar con sistemas y plataformas que brinden la seguridad necesaria para crear confianza en estas tecnologías (Almeida, 2018)

> La información es un activo valioso para casi todas las organizaciones, por lo que muchas invierten cantidad de recursos en que esta sea correctamente creada, administrada y mantenida (Voutssas M, 2010)

Por lo tanto, es esencial contar con mecanismos que aseguren la integridad de la información de manera que se eviten muchas de las problemáticas comunes, entre las que encontramos: perdida o destrucción de información, falsificación de documentos, ataques de ingeniería social, entre otros (Kushmaro, 2018).

Según Gonzáles Glenda, (2018) uno de los casos de uso dentro del ámbito de la educación es el de la certificación académica.

Además, estos certificados, evidencia del aprendizaje de un estudiante, tienen la finalidad de ser vistos y comprobados por terceras partes (Cheng, Lee, Chi & Cheng, 2018, p.27).

En la actualidad muchos de los registros académicos se emiten en documentos físicos impresos u otros formatos similares (Cheng, Lee, Chi & Cheng, 2018, p.30).

Estos formatos son propensos a deteriorarse y en ocasiones requieren de una comunicación directa con la institución para realizar su validación respectiva. Además, gran cantidad de estos han sido falsificados o alterados, dificultando la tarea de obtener la validez de los mismos, así como rastrear su procedencia como reporta “El Telégrafo” (2013) .

El Ecuador ha sido un país en el que la innovación tecnológica se encuentra por debajo de los niveles esperados, la adaptación tecnológica lleva casi 20 años de diferencia con respecto a otros países de la región, por tanto, sería comprensible que en la actualidad se continúe confiando en métodos de certificación manuales aún en gran cantidad de centros de estudios, institutos o universidades en el país (Pinasco, 2019).

Según las estadísticas otorgadas por la Secretaría Nacional de Educación Superior Ciencia Tecnología e Innovación (Rosales, 2018), aproximadamente existen más de 150 mil graduados cada año. Algunos de ellos irán a países, escuelas secundarias o instituciones terciarias ya sea para continuar sus estudios o integrarse al campo laboral (Cheng, Lee, Chi & Cheng, 2018, p.30)

Por lo tanto, es necesario que los estudiantes puedan validar su conocimiento adquirido ya sea en cursos, escuelas, capacitaciones, etc. El cual puede constar en diplomas, certificados, plataformas de e-learning, entre otros. De manera que sean una referencia importante para solicitar su admisión en nuevas escuelas o trabajos.

Como es común que estos documentos contengan básicamente información del estudiante, institución y educación recibida, impresa o quizá almacenada en una base de datos privada, sin muchas seguridades ocasionando que estos documentos sean vulnerables ante falsificaciones o adulteraciones, por lo que los interesados no tienen una manera veraz de legitimar los datos. Por lo tanto, es necesaria una solución que permita respaldar esta información, validarla y proteger su integridad.

Una de las tecnologías que lleva varios años como una de las principales tendencias disruptivas es Blockchain. Se trata de una tecnología revolucionaria para almacenar y compartir datos, la cual proporciona una base de datos descentralizada basada en un conjunto de redes peer to peer, cuyos nodos son capaces de llegar a un consenso para detectar posibles alteraciones en los registros, invalidando posibles datos falsificados o modificados, sin necesidad de intermediarios o autoridades de verificación (Gartner, 2019).

Desde su aparición a inicios del 2009 como la tecnología que permite el funcionamiento de la criptomoneda Bitcoin, se ha demostrado que tiene gran cantidad de usos aplicables a todo tipo de soluciones debido a que es sumamente adaptable y en la actualidad tanto organizaciones públicas, privadas y entidades gubernamentales se encuentran poniendo en marcha nuevas propuestas y proyectos relacionados a esta tecnología (REYES DELGADO, 2018)

La propiedad de Blockchain de inalterabilidad llega a ser una opción adecuada al momento de brindar una propiedad anti falsificación a los datos que se almacenen en la red y como apunta. (Cetina, 2020).

“Con ‘Blockchain’ se pueden resguardar documentos sensibles y tener información del histórico de la documentación, de manera que se puede realizar una trazabilidad para reconocer el registro, evitando la falsificación”(Tovar, 2020).

Por tanto, una solución basada en esta tecnología es adecuada para lograr un proceso de validación y verificación de certificados educativos de manera segura y transparente, a pesar de sus útiles características se considera una tecnología que está en sus primeros años de desarrollo.

Según Gartner (2019) a la fecha actual, la tecnología ha alcanzado la fase de la desilusión, una fase que disminuye las expectativas que se tenía en sus inicios y de la cual se espera salga en 2 a 5 años a medida que los casos de uso prácticos continúen desplegándose. Es debido a esto que no existen estándares o protocolos para la realización de soluciones fuera del sector financiero, convirtiendo a los nuevos casos de uso e implementaciones con Blockchain en una oportunidad para aportar con el desarrollo de esta tecnología y explorar sus capacidades.

Problemática

Definición del problema

Los estudiantes de las diferentes instituciones educativas, programas, cursos, conferencias, etc. Una vez culminados sus estudios, comúnmente dependen de certificados físicos para poder validar su conocimiento adquirido, los cuales no siempre son fáciles de autentificar como válidos, dificultando los procesos de selección en lugares de trabajo u otros centros de estudios. Otro inconveniente para el caso de poseer certificados educativos en plataformas en línea es que se debe confiar en la seguridad informática del emisor para mantener la integridad y disponibilidad del certificado de aprendizaje.

Contexto del problema

Considerando las estadísticas brindadas por la Secretaría de educación superior, ciencia, tecnología e innovación. En el Ecuador se registraron 116.766 títulos universitarios registrados en el Senescyt en el 2018 que corresponden a personas que han completado sus estudios de tercer nivel en universidades y otros establecimientos educativos (Rosales, 2018).

Sin embargo, a pesar de contar con el registro oficial de este título en la base de datos de la secretaria, no sucede de la misma forma con estudiantes que a más de contar con un título de tercer nivel, necesitan validar sus conocimientos en temáticas o tecnologías específicas, ya sea que vengan estos de cursos, talleres, capacitaciones, conferencias o incluso experiencia laboral. La importancia de la validación de estos saberes para los estudiantes radica en las oportunidades ya sea de estudio o empleo que evidentemente desearán obtener.

En la actualidad, a pesar de que las instituciones comienzan a utilizar registros electrónicos, muchas de estas siguen manteniendo procesos manuales para transferir registros académicos como transcripciones que pueden tardar días o incluso meses en procesarse, teniendo como resultado a veces el deterioro o perdida de estos registros (Cheng, Lee, Chi & Cheng, 2018, p.27).

Si se considera además a las plataformas de e-learning como Udemy, sería correcto afirmar que poseen sus propios sistemas de generación de certificados educativos, cuya seguridad en la preservación documental es responsabilidad de cada una de estas plataformas (Udemy, 2019).

La información digital también puede ser vulnerable de no existir la adecuada seguridad de la información, siendo esta propensa a inutilizarse o destruirse tal como sucede con los documentos en formato físico (Voutssas M, 2010).

Adicionalmente para instituciones que emplean registros en papel se deben considerar los costos de transporte, impresión y almacenamiento de los mismos. Las instituciones certificadoras ofrecen algunas formas de verificación, poniéndose en contacto con el emisor original o los servicios de terceros, pero a menudo estos trámites son engorrosos y no inmunes al fraude.

Como reporta La Hora, (2003) en el Ecuador esto podría ser una de las causas de la gran cantidad de documentos falsificados que circulan en el país. Muchos de estos correspondientes a instituciones, conferencias o cursos inexistentes, y por tanto imposibles de verificar su autenticidad, además de la existencia de mafias dedicadas a la falsificación de títulos y certificados académicos a cambio de dinero.

El estado ecuatoriano reconoce a estas acciones como un delito el cual se encuentra establecido en el Art. 328, del Código Orgánico Integral Penal (COIP), tipifica la falsificación y uso de documentos falsos y lo sanciona con penas de 3 a 4 años (Derecho Ecuador, 2018, p.92).

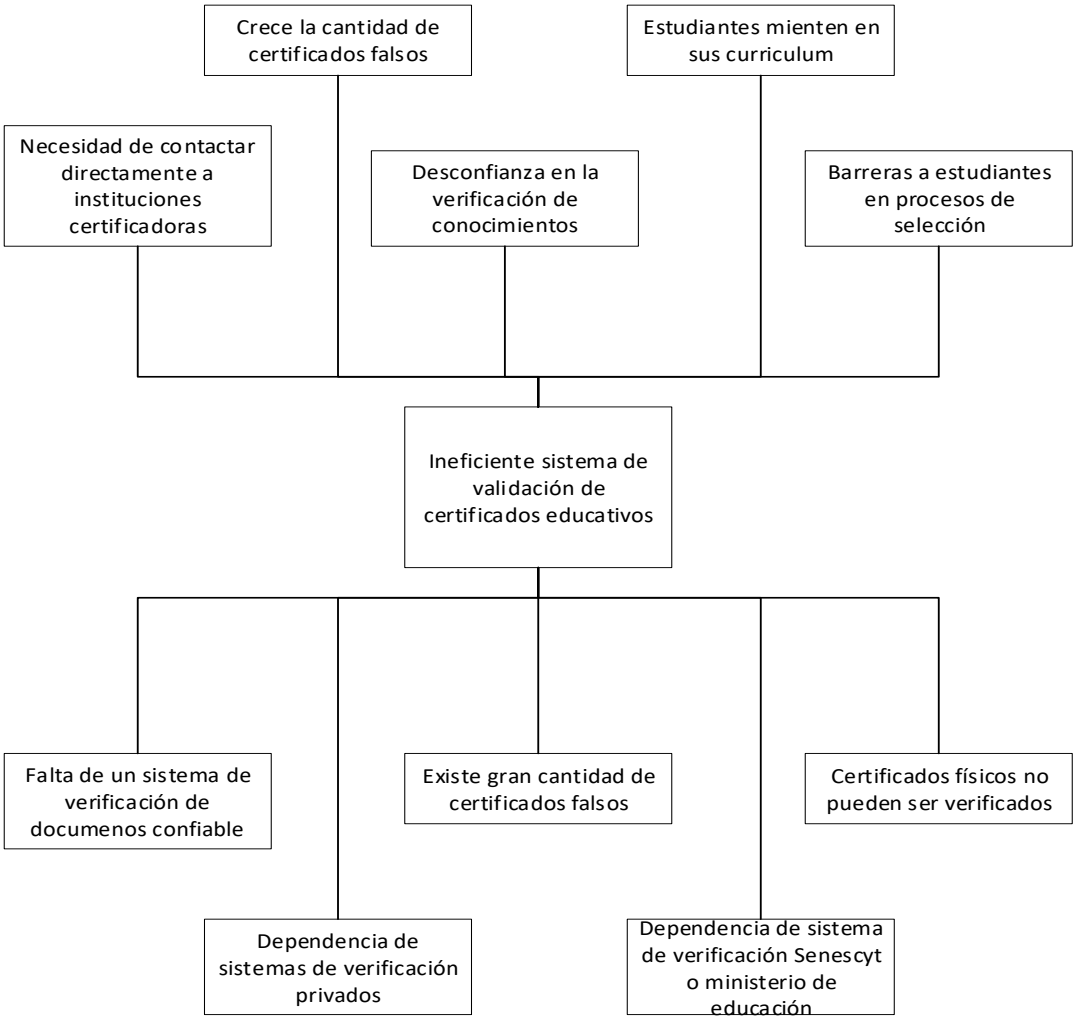
Para lograr una confianza digital se debe ir adoptando una cultura de colaboración entre los interesados, dando la confianza de usar las herramientas digitales, pero también compartiendo responsabilidades de manera que la gente no deserte del aprendizaje tecnológico por miedo a equivocarse (Cortés, 2019).

Árbol de problemas

La **Figura 1** muestra el problema central detectado, así como sus causas y consecuencias.

Figura 1.

Árbol de problemas



Justificación

Como se ha expresado en la actualidad Blockchain es una tecnología que necesita madurar, pero una vez que lo haga se prevé su impacto en industrias enteras mejorando la confianza en ecosistemas tecnológicos, razón por la cual ha llegado el momento de experimentar para aportar con el desarrollo de esta tecnología y explorar sus capacidades.

Con un mecanismo de validación seguro, y que cuente con una adecuada experiencia de usuario, se espera genere la confianza en los usuarios de abandonar procesos tradicionales de certificación manuales.

Al contar con un sistema de validación de certificados educativos basado en una tecnología descentralizada y segura, permitiría contar un único medio de almacenamiento y validación de estos documentos ya sea para instituciones educativas, de capacitación o plataformas de e-learning, de manera que cualquier certificado de educación que haya obtenido una persona se pueda validar en la misma plataforma indistintamente de su procedencia.

El impacto del proyecto a largo plazo se espera que mejore el proceso de selección para las empresas e instituciones que deseen reclutar talento humano con capacidades específicas, esto debido a que se generaría confianza en que los certificados educativos de los estudiantes no sean falsificados, lo que ayudaría a ganar competitividad en el campo laboral para el Ecuador.

Se espera además que, al tener un sistema de validación seguro de certificados de aprendizaje, este brinde utilidad como un modelo que pueda ser aplicado de manera general para cualquier tipo de certificación educativa de manera que el conocimiento adquirido por una persona sea reconocido en el momento que se necesite, siendo este

un incentivo para personas que desean continuar con sus estudios motivando su participación en completar cursos, conferencias, congresos, etc.

En cualquier modalidad que estén disponibles, disminuyendo de esa manera el porcentaje de personas que no completan sus estudios, como es el caso de los cursos en línea de las plataformas de e-learning, que según la Escuela Europea de Dirección de Empresa tiene una tasa de deserción alta, donde el 90% abandona sus cursos MOOC y el 35% sus másteres virtuales (Carrizosa, 2014)

Objetivos

Objetivo general

Diseñar y desarrollar un sistema basado en Blockchain que permita el almacenamiento y validación de certificados educativos de manera segura.

Objetivos específicos

- Elaborar el estado del arte que permita conocer el funcionamiento de la tecnología Blockchain, sistemas de validación documental y proponer un prototipo.
- Realizar el análisis, diseño y desarrollo del sistema aplicando la combinación de las Metodologías Agiles SCRUM y XP para conseguir un prototipo funcional de validación de certificados educativos.
- Evaluar y validar los resultados, rendimiento y seguridad del sistema.

Alcance

Para delimitar el alcance del proyecto a desarrollarse se ha considerado las siguientes fases:

1. Estudio sobre Blockchain y validación documental.

- Identificar qué mecanismos son los más adecuados al momento de realizar la verificación de la autenticidad de un certificado educativo.
- Identificar las tecnologías de Blockchain más apropiadas para el almacenamiento y validación de certificados educativos.

2. Elaboración del diseño de la arquitectura del sistema.

    - Identificar los requerimientos de la solución.
    - Identificar los componentes que participarán en el desarrollo del sistema.
    - Realizar el diseño del sistema.

3. Construcción del prototipo del sistema.

    - Desarrollar cada uno de los componentes de la solución descritos en la arquitectura.
    - Realizar la integración de los componentes.
    - Desplegar el sistema y sus componentes en una red de prueba.

4. Evaluación y validación de los resultados de la solución.

    - Ejecutar casos de prueba
    - Analizar resultados

Preguntas de investigación

- ¿Qué mecanismos de seguridad son necesarios para lograr un sistema de validación documental basado en Blockchain?
- ¿Qué Framework o herramienta de Blockchain es la más adecuada para el almacenamiento y validación de certificados educativos?
- ¿Cómo se debe realizar la gestión de información en Blockchain para lograr un eficiente sistema de validación de certificados educativos?

Hipótesis

El desarrollo de un sistema de validación electrónica basado en Blockchain provee de un mecanismo eficiente, seguro y confiable de mantener la integridad de documentos electrónicos y confirmar su autenticidad de manera sencilla.

Variables de investigación

Se definen las variables de investigación independiente y dependiente a partir de la hipótesis

- **Variable independiente:** Desarrollo de un sistema de validación electrónica basado en Blockchain
- **Variable dependiente:** Mecanismo eficiente, seguro y confiable de mantener la integridad de documentos electrónicos y confirmar su autenticidad.

Capítulo II

Metodología y Estado Del Arte

Metodología

Durante el desarrollo del presente trabajo, se utilizaron métodos y técnicas que permitan alcanzar el resultado esperado y que orienten el enfoque de la investigación. Para poder desarrollar software de manera exitosa, existen varias metodologías en las cuales basarse. Se detalla a continuación algunas de ellas.

XP o Extreme Programming

La metodología ¨Programación Extrema¨ o conocida por su abreviatura ¨XP¨ es una metodología ágil de desarrollo de software. La metodología ágil hace referencia a principios y valores de desarrollo que permitan la elaboración de software de manera rápida y respondiendo a los cambios que puedan surgir durante la etapa de desarrollo del proyecto (Torres et al., 2017).

Es una alternativa a los procesos de desarrollo de software tradicional, que se caracterizan por ser rígidos e inflexibles por la documentación generada en cada una de las actividades que se desarrolla.

La metodología XP se concentra en potenciar las relaciones interpersonales como la clave de éxito en el desarrollo del programa, utilizando siempre el factor humano como clave del desarrollo, fomentando el trabajo en equipo, preocupándose por el aprendizaje integral de los desarrolladores, y proporcionando un ambiente de trabajo idóneo para cada uno de los implicados en el desarrollo del proyecto (Torres et al., 2017).

Esta metodología está diseñada para entregar el software a los clientes que necesitan, en el momento en que lo necesitan; respondiendo a los requerimientos cambiantes de los mismos. Este modelo define cuatro variables que deben ser tomadas en cuenta al momento de realizar proyectos de software, estas son: costo, calidad, tiempo y alcance. (Navarro et al., 2013).

Según Torres (2017) mediante la metodología XP el ciclo de desarrollo consiste en los siguientes pasos:

1. El valor del negocio es seleccionado por el cliente
2. El esfuerzo para la implementación lo establece el desarrollador de software
3. El cliente selecciona que construir dependiendo de sus prioridades y restricciones de tiempo.
4. El programador elabora dicho valor de negocio.
5. Se repite el paso 1.

Durante el desarrollo del ciclo de trabajo, tanto el cliente como el programador aprenden; el programador no debe ser presionado para elaborar más trabajo que el estimado en el plan, pues la calidad y el tiempo de desarrollo se verán afectados(Torres et al., 2017).

Es por tanto que la metodología XP, cuenta con características que permiten un desarrollo adecuado de la programación. Permiten juego de planificación, es decir, un espacio frecuente de comunicación entre los programadores y el cliente, permitiendo el aporte mutuo de las partes para aumentar el valor de negocio, disminuyendo riesgos y mejorando la planificación del proceso en general. Propone también entregas pequeñas al cliente, permitiendo que el usuario pueda verificar el avance adecuado del sistema;

de esta manera la metodología no emplea una definición temprana de arquitectura, esta va evolucionando de acuerdo a las iteraciones que se realicen, y se define mediante metáforas compartidas entre el cliente y el equipo de desarrollo. Esto facilita el diseño, disminuye el número de pruebas necesarias para la correcta ejecución y reestructura el código con el objetivo de mejorar legibilidad y flexibilidad para procesos futuros.

Todos los aspectos anteriormente mencionados encajan con las necesidades del presente proyecto, y hacen que esta metodología pueda ser utilizada para el desarrollo del sistema de validación en Blockchain, mediante una programación flexible, arquitectura evolutiva, de manera eficiente, eficaz y segura.

Ad-Hoc

Ad-Hoc, o también llamadas pruebas aleatorias, es una metodología de comprobación de software en el cual se realizan ensayos y pruebas al programa sin planificación y documentación. Este tipo de pruebas se realizan de manera informal y al azar, esperando por errores en la programación (Sánchez, 2015).

El desarrollador improvisa, tanto en pasos como en ejecución de manera arbitraria, por lo tanto, los defectos encontrados mediante este método son en ocasiones difíciles de reproducir, describir y documentar. Durante la ejecución de este tipo de metodología de ensayos se encuentran defectos que no se esperaba y nuevos tipos de errores; con lo que este método es utilizado durante la prueba de aceptación del programa (Sánchez, 2015).

Durante el desarrollo de estas pruebas aleatorias es necesario tener creatividad y tenacidad al probar el programa, además de cierto porcentaje de suerte. Debido a esta naturaleza de las pruebas, se hace mención a una metodología de caja negra, es decir, que se basa únicamente en las entradas y salidas que presenta el programa y

dependiendo única y exclusivamente de los resultados que se tiene al aplicar ciertos parámetros de entrada, permitiendo al programador enfocarse en lo que el programa realiza y no en cómo lo realiza (Sánchez, 2015).

Este tipo de pruebas tienen ciertas características:

- Se realizan generalmente después de la finalización de las pruebas formales.
- Su objetivo es romper la solicitud del programa sin aplicar un proceso específico.
- El desarrollador de las pruebas debe tener un conocimiento profundo sobre el producto.
- Los errores encontrados durante la ejecución de esta metodología exponen lagunas en el proceso de pruebas formales.

Ventajas de utilizar la metodología de pruebas Ad-Hoc:

- La aplicación de pruebas aleatorias da libertad al programador para utilizar sus propias formas de probar el programa, facilitando de esta manera descubrir un mayor número de defectos en comparación con pruebas formales.
- Este tipo de pruebas se puede realizar en cualquier etapa del ciclo de vida del software de desarrollo (SDLC).
- Se pueden realizar a módulos de programación, permitiendo programar de manera más adecuada.
- Estas pruebas demuestran ser muy beneficiosas en plazos cortos de desarrollo.
- Pueden ejecutarse de manera simultánea con otro tipo de pruebas, permitiendo búsqueda de más errores en menor tiempo.
- No es necesario llevar documentación de las pruebas.

Desventajas de utilizar la metodología de pruebas Ad-Hoc:

- La recreación de errores se dificulta debido a que no se posee una planificación formal.
- Al no tener documentación de los escenarios de prueba, el desarrollador de las pruebas debe mantener en mente todas las características del ensayo.
- Este tipo de ensayos es dependiente de la pericia del desarrollador, pues es necesario un conocimiento profundo del programa.

La aplicación de este tipo de pruebas en la programación del presente proyecto permitirá encontrar errores de difícil detección mediante pruebas tradicionales, además de que se podrán realizar durante todas las fases del diseño, obteniendo resultados rápidos y permitiendo verificar cada paso realizado durante la ejecución del programa. Estas pruebas serán realizadas por el desarrollador del programa, por lo que se tendrá un amplio conocimiento del programa y las pruebas a realizar serán adecuadas (Sánchez, 2015).

Scrum

Esta metodología propone un marco de trabajo que mejora la productividad en los equipos de trabajo de manera ágil (Trigás et al., 2018). Se fundamenta en un control incremental del desarrollo de los requerimientos y mantiene una serie de roles para su empleo. Se basa en transparencia, inspección y adaptación (Navarro et al., 2013).

Los equipos Scrum se caracterizan por ser multifuncionales y trabajar bajo iteraciones. Esto hace que los equipos puedan elegir las mejores formas para hacer el trabajo; los miembros deben tener los conocimientos necesarios para poder llevar a cabo el trabajo, y la entrega del producto debe realizarse en iteraciones, y en cada una

de ellas se crea nuevas funcionalidades dependiendo de las necesidades del usuario (Trigás et al., 2018).

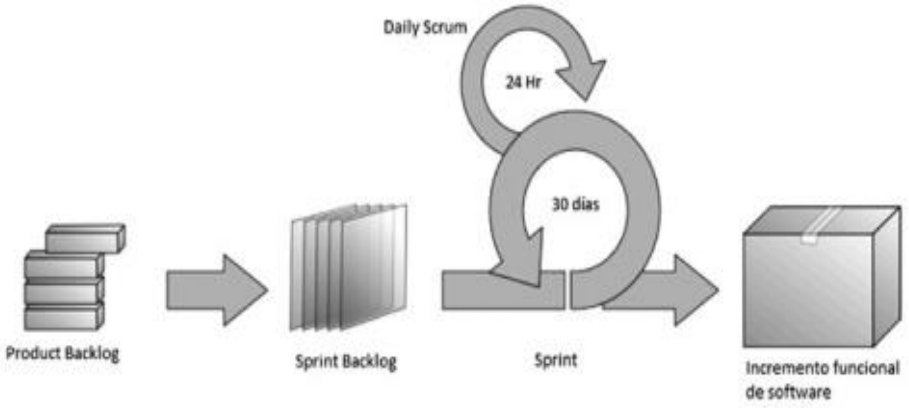
Los roles que define esta metodología consisten en el Product Owner quien representa los intereses del cliente y da valor al producto, el Scrum Master quien se lo considera un líder pero no un gestor de desarrollo, equipo de desarrollo encargado de la implementación de los componentes de la solución (Navarro et al., 2013)..

El periodo de trabajo en Scrum se encuentra definido por un Sprint, la **Figura 2** corresponde a una ventana de tiempo donde se crean versiones utilizables del producto (Navarro et al., 2013).

Cada Sprint posee artefactos, que son subproductos de las actividades que realiza el grupo, estos son: Product Backlog, Sprint Backlog, Monitoreo de Progreso e Incremento.

Figura 2.

Metodología Scrum: Fases de un Sprint.



Nota. En la figura se muestra la metodología Scrum, por (Navarro et al., 2013).

El Product Backlog son los requerimientos que el dueño da al programa; el Sprint Backlog son los requerimientos más importantes del Product Backlog, son las características principales para la realización del Sprint; el monitoreo del progreso es la suma de trabajo que falta por realizar para culminar el Sprint, y permite al dueño evaluar el progreso del desarrollo; finalmente, el incremento es la suma de todos los ítems terminados en el Sprint Backlog, es lo que se presenta al cliente y al grupo de Scrum para su análisis (Navarro et al., 2013).

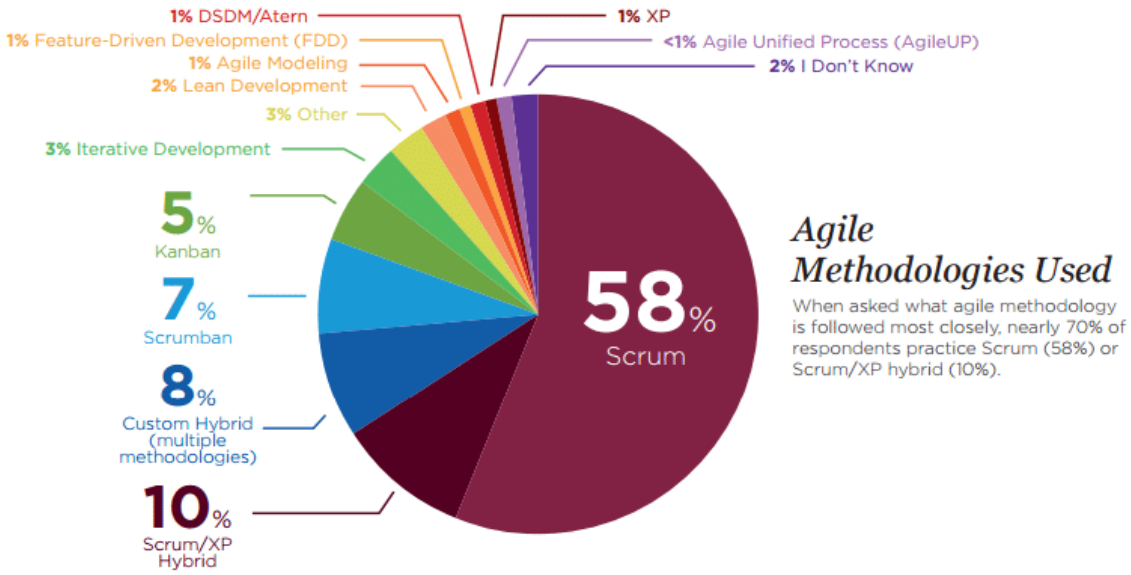
Cada uno de estos eventos o sprints es considerado un evento independiente, y tiene una duración máxima de un mes. Este Sprint contiene los siguientes componentes o actividades: reunión de planeación, Daily Scrum, trabajo de desarrollo, revisión y realimentación (Trigás et al., 2018).

- La reunión de planificación se realiza con el fin de establecer cuáles van a ser los entregables y el cómo lograrlos.
- El Daily Scrum es un evento del equipo de desarrollo, que se realiza a diario y dura 15 minutos. Básicamente es un control sobre lo que se ha realizado desde la anterior reunión, lo que se va a realizar sus posibles inconvenientes.
- El trabajo de desarrollo se realiza por el equipo Scrum, y en este se siguen todos los lineamientos anteriormente analizados, y se procede a la realización de las tareas del programa.
- La revisión del Sprint se realiza al final del mismo, en esta etapa el dueño del proyecto revisa lo que se ha realizado, identifica lo que no se ha realizado y discute sobre el Product Backlog. Durante este control se presentan los problemas encontrados y su resolución.
- Se fortalece la comunicación al tener un espacio para expresar los procesos realizados correctamente y los que no, en vista de encontrar mejoras.

Según el estudio, 10th Annual State of Agile Report, Scrum es utilizado en el 58% de los métodos y practicas ágiles de desarrollo de software como se puede ver en la **Figura 3**, siendo una de las metodologías líderes para el desarrollo de programas. Sus directrices son eficaces y permiten un desarrollo adecuado de software, entre el cliente, el líder de programación y el equipo de trabajo (Report, 2020).

Figura 3.

Metodologías ágiles utilizadas en grupos de desarrollo de software durante 2020.



Nota: La figura se observa el uso de las metodologías ágiles, por (Report, 2020).

Estado del arte

Para la elaboración del presente proyecto, es necesario realizar una investigación sobre estudios previamente realizados sobre la temática en cuestión. Esta es una base de información fundamental, que permitió entender el funcionamiento de la tecnología Blockchain, conocer las aplicaciones que se han realizado con esta tecnología, y de esta manera poder aplicar todos los conocimientos adquiridos a través

de esta investigación, para el desarrollo adecuado del sistema para la validación de certificados educativos.

Papers

La tecnología Blockchain proporciona una propiedad de trazabilidad en la que las transacciones procesadas se pueden rastrear, así como los activos que involucren a estas (Gupta, 2017).

Al aumentar la popularidad de esta tecnología en procesos de registro y proceso de activos, muchos artículos han surgido en donde se aplica esta tecnología para diversas aplicaciones. Uno de los artículos realizados en la Universidad de Piraeus, se realiza un estudio sistemático de las aplicaciones de la tecnología Blockchain, en donde se analiza el estado actual de esta, una clasificación y posibles problemas que se afronta al utilizarla.

El trabajo proporciona una breve explicación del funcionamiento de esta tecnología, su arquitectura y la distribución peer to peer en la que se basa. A partir de estos aspectos, se detalla múltiples aplicaciones en donde este tipo de tecnología basados en conferencias, artículos, libros de alta relevancia científica desde el 2014 hasta finales del 2019. Según este artículo Blockchain es aplicable en operaciones financieras, verificaciones de integridad de archivos, aplicaciones gubernamentales, municipios, sector público, en el IoT, gestión de la salud, seguridad y privacidad, aplicaciones industriales y sobre todo en la educación (Casino et al., 2019).

Se expresa que Blockchain puede resolver problemas de vulnerabilidad, seguridad y privacidad en el entorno del aprendizaje, se puede utilizar para almacenar registros educativos relacionados con recompensas al mérito. El artículo propone el uso de sistemas distribuidos basados en Blockchain para registro educativo y de diplomas.

Se muestra también referencias a universidades que han utilizado este sistema para agregar bloques de cadena que almacenan estos certificados, mejorando la seguridad de los datos almacenados (Casino et al., 2019).

Finalmente, este trabajo propone las limitaciones que presenta la tecnología, como una autonomía más eficiente en las cadenas de bloques, además que la tecnología en la que se basa son aplicaciones que se han implementado hace varios años, pero al aplicarlos en conjunto, hace ideales los usos que se le puede dar en determinados campos. Expresan que mientras las cadenas de bloques se vuelvan más maduras, escalables y eficientes, sus aplicaciones se adentraran más en la industria y dominios mucho más específicos (Casino et al., 2019).

De este trabajo se puede obtener información muy útil sobre todas las características necesarias para poder aplicar esta tecnología en el presente proyecto, además de entender las características que hacen de esta tecnología una solución para la verificación de documentos electrónicos.

Otra de las investigaciones es una realizada en la Universidad Deakin, en donde se realiza un sistema de micro-credenciales basados en la tecnología Blockchain. En este artículo se realizan verificaciones de credenciales e insignias digitales para los profesionales dentro de la universidad. El trabajo se centra en verificar todas las credenciales que provee la Universidad de Deakin, y el objetivo principal era el de mejorar el sistema de acreditación de estas credenciales, verificar la veracidad de estas y además mejorar los problemas de privacidad (Sai et al., 2019)

Este programa se desarrolló dentro de la Universidad, y se pueden evidenciar en el mismo, los resultados obtenidos al utilizar la tecnología Blockchain dentro del mundo educativo, y donde la universidad tiene la posibilidad de adoptar la tecnología. En el

caso específico de estudio, el trabajo realizado permitió proveer de autenticidad, confianza, seguridad y credibilidad a las credenciales que otorgaba la universidad a los profesionales (Sai et al., 2019)

Como resultado de este trabajo, el modelo creado permite a cualquier profesional que quiera obtener una de las credenciales proporcionadas por la Universidad Deakin, debe seguir el proceso de admisión mediante la aplicación. Una vez que haya realizado de manera exitosa todo este proceso, las credenciales serán adjudicadas a dicho profesional. El portafolio de cada persona se envía a la base de datos de la universidad para ser evaluado.

Según (Sai et al., 2019) el profesor a cargo calificara y dará las notas a cada portafolio y se adjudicara la certificación. El programa realizado llamado Credly emite las insignias digitales, se almacenan en Blockchain con acceso autenticado. Este programa emite y gestiona estas credenciales digitales, que pueden ser verificados por cualquier empresa particular en todo el mundo.

Las conclusiones del artículo especifican que el programa creado basado en Blockchain elimina las necesidades de tener una autoridad central para administrar las credenciales, y proporciona una forma de verificar cada una de las emisiones de manera segura, inmutable y transparente.

El registro es permanente y ningún tercero puede modificarlo sin ser rastreado, por tanto, es muy difícil que pueda ser hackeado. Sin embargo, se expresan problemas específicos en el modelo planteado, sobre todo en la capacidad de almacenamiento y en la escalabilidad (Sai et al., 2019). Proponen que, si las transacciones se vinculan a direcciones IP, se puede exponer la información personal de ubicación del usuario, y plantean mejoras a esos aspectos.

Se puede entonces notar las aplicaciones en el sector de la educación del sistema elaborado por los investigadores, obteniendo datos muy necesarios para la aplicación en el presente trabajo, utilizando los resultados obtenidos y trabajando en los problemas que tuvieron para tratar de eliminarlos al utilizar la tecnología Blockchain.

En cuanto a investigaciones y trabajos realizados mediante la técnica de Blockchain en Latinoamérica, dos artículos han destacado especial atención. El primero es el “Desarrollo de un sistema para recaudo y pago de tarifas de transporte público intermunicipal mediante tecnología Blockchain”. Este trabajo se desarrolló en la ciudad de Valledupar, en Colombia. El objetivo principal de este trabajo era el de desarrollar un aplicativo móvil que permita recaudo y pagos de tarifas de transporte, además de la visualización de vehículos y cupos de transporte (Carrillo & Roa, 2020).

Para lo cual se diseñó una red descentralizada para una empresa de transporte, con transacciones seguras y creación de contratos digitales. Se creó una base de datos online para el almacenamiento de todo tipo de la información, tanto de usuarios como de transportistas y se creó un sistema de calificaciones. En cuanto al aplicativo móvil fue desarrollado en Android, y se creó un manual de usuario orientado a la capacitación del personal sobre el sistema de pagos y los recaudos de las tarifas de transporte mediante Blockchain (Carrillo & Roa, 2020).

El trabajo además realiza una comparación exhaustiva en cuanto a los métodos de pago, métodos de almacenamiento de datos y la creación de las redes descentralizadas mediante Blockchain, lo que proporciona información muy útil para la aplicación en el presente trabajo.

El segundo estudio, es el realizado en la Universidad de Cundinamarca, Colombia. El estudio consiste en la creación de una “Plataforma tecnológica de voto

electrónico para elección de cuerpos colegiados de una universidad, que integra módulos Blockchain y ciberseguridad" (Chila et al., 2020).

El trabajo se centra en la creación de sistema para suplir la necesidad con respecto a los procesos de elección de representantes universitarios. En la investigación realizada se presenta el desarrollo de software mediante la utilización de tableros Kanban, se realizó planeación y control de tareas mediante sprints. En el artículo se presenta además diagramas que permiten observar las principales funcionalidades de módulos Blockchain para asegurar la seguridad del voto electrónico, la integración de todos los sistemas y la interacción entre cada uno (Chila et al., 2020)

Mediante la aplicación de esta tecnología, los autores lograron desarrollar el sistema seguro, confiable y rastreable de los votos realizados para la elección de los representantes universitarios. Finalmente, en cuanto a aplicaciones de Blockchain en el Ecuador, se ha estudiado dos trabajos realizados en base a los Smart Contracts y otro de registro de títulos académicos. El primero es un trabajo realizado en la Universidad Central del Ecuador, sobre "Los Smart Contracts como alternativa para la modernización de recaudación tributaria en el Ecuador" (Novoa Z. et al., 2020).

En este trabajo se propone la utilización de los contratos inteligentes para representar acuerdos legales programados. Se presenta una explicación muy amplia sobre la utilización de redes descentralizadas para la realización de contratos digitales, intercambios entre vendedores y compradores. Se presenta una codificación en una base de datos interconectados, creando de esta manera cadenas de entradas de información en bloques que pueden ser rastreadas, y permite obtener los datos ordenados de manera cronológica, inalterable y de manera permanente (Novoa Z. et al., 2020)

Se presenta entonces el aplicativo como una necesidad en tiempos de pandemia, mediante la recaudación de impuestos mediante la tecnología Blockchain, de manera ágil, segura e inalterable. Después de la realización del aplicativo, en el estudio se presenta las conclusiones, en las cuales se expresa que fue una gran alternativa para modernizar y hacer más eficaces los procesos de administración pública. Plantean este tipo de elementos como una forma de modernizar la organización tributaria en el Ecuador y gracias a la utilización de esta tecnología, proveen de seguridad a todos los procesos contractuales digitales (Novoa Z. et al., 2020)

El último trabajo a analizar, en el que se plantea un “Registro de títulos académicos mediante una aplicación basada en Blockchain y Smart Contracts”. Este trabajo plantea una similitud muy grande con el trabajo a realizar. Se presenta la implementación de la tecnología Blockchain para generar una base de datos de registros de títulos de manera segura y fiable. Esto con el objetivo de asignar títulos académicos a estudiantes, sin la necesidad de un ente central o terceras personas que administren la información, evitando fraudes y robo de datos (Morales Morales et al., 2020)

En este trabajo se desarrollaron dos contratos inteligentes complementarios entre sí, estos se ejecutaron en entorno virtualizado de Ethereum, con un conjunto de herramientas de Truffle. Se evaluaron cada uno de los contratos inteligentes mediante el ingreso de los datos de prueba y con tales registros almacenados se ejecutó el proceso de asignación de títulos académicos a los estudiantes. Para las validaciones de los procesos se realizaron consultas a la cadena de bloques, y se realizaron verificaciones a los registros de asignaciones (Morales Morales et al., 2020)

Al final de todas las pruebas realizadas, este trabajo dio como resultado un modelo de aplicación para el registro adecuado de los títulos académicos generados. Se verificó la factibilidad y seguridad de Blockchain, y se pudo validar que la metodología utilizada es factible para este tipo de aplicaciones. Por lo tanto, se puede extraer mucha información muy útil para la aplicación de la tecnología Blockchain en la verificación de documentos electrónicos.

Conclusión del estado del arte

Una vez analizados los diferentes trabajos previos realizados, relacionados con la tecnología Blockchain; se puede concluir que este tipo de redes descentralizadas, se pueden aplicar en gran cantidad de áreas multidisciplinarias. Permite brindar seguridad y fiabilidad a los procesos digitales, ya sean estos contratos, asignaciones y cualquier tipo de proceso de valor. Provee al usuario y a la compañía validez en los procesos que se realizan, bases de datos seguras, y verificación internacional de los procesos realizados.

Las aplicaciones de esta tecnología, evitan la necesidad de entes centrales que se encarguen de la verificación de los procesos, evita el manejo de la información de los clientes y disminuye el trabajo humano realizado. Esto conlleva a mejorar la seguridad de dichos procesos, permite el seguimiento detallado de cada uno de los procesos, por lo cual se vuelve muy difícil de alterar cualquiera de los datos que intervienen en el proceso, y por tanto los hackeos se minimizan. Además, los costos de administración se acortan, elimina mucho de los factores humanos que puedan conllevar errores, y se tiene un backup de la información si es que fuera necesario.

En cuanto a las aplicaciones en la educación, se puede concluir que la tecnología Blockchain es una herramienta eficaz, ya probada mediante una serie de

investigaciones en Europa y Latinoamérica. Se pudo verificar mediante la investigación que esta tecnología cumple con los requisitos para poder verificar, asegurar y monitorear cualquier tipo de documento electrónico que provenga de un ente educativo. Esto permite que empresas internacionales puedan verificar de manera rápida y sencilla cualquier documento generado mediante las entidades educativas.

Finalmente, a partir de toda la información obtenida de la investigación realizada se puede notar que la tecnología Blockchain es aplicable en el presente trabajo, teniendo una gran base informativa sobre la forma de utilización de la misma, datos sobre las problemáticas que se generan al aplicarla, y se tiene por tanto las de características necesarias a poner en práctica de manera adecuada, para que el sistema a realizar sea eficaz, eficiente y seguro.

Capítulo III

Marco Teórico

Modelo descentralizado en aplicaciones

Las redes descentralizadas consisten en la conectar un grupo de computadores de manera que puedan comunicarse e interactuar entre sí sin la necesidad de un servidor central (Hernández Juárez, 2020).

Los ordenadores interconectados se conocen como nodos y abren un canal por el que traspasan los datos hacia su destino como si fuera una malla. Esto ofrece unas propiedades interesantes de libertad para el usuario al momento de seleccionar cómo alojar su información (Hernández Juárez, 2020).

Los modelos descentralizados en aplicaciones brindan muchas ventajas en cuanto a los sistemas centralizados, desaparece el mainframe y permite que desaparezcan estructuras de coste de soporte, y permite una velocidad de avance de la información mucho mayor. Mediante la aplicación de los modelos descentralizados los usuarios pueden tener programas que responden mucha más rapidez, permitiendo llegar a requerimientos de manera mucho menos costosa (Preukschat, 2017).

Esta clase de modelos permiten que los niveles corporativos puedan estar mucho más cerca a los clientes, proporcionando a estos de la mayor cantidad de recursos, adoptando las decisiones sobre gastos y la gestión. La aplicación de esos sistemas implica que los rangos corporativos eliminen departamentos de sistemas de información central, haciendo la organización más simple, racionalizada y eficiente (Preukschat, 2017).

Técnicas criptográficas

La criptografía hace referencia a la transformación de un mensaje legible en uno ilegible. Este proceso es también llamado cifrado, y en la actualidad se cuenta con tres tipos principales de criptografía: Criptografía simétrica o convencional y criptografía asimétrica o de clave pública y Hashing (Preukschat, 2017).

Estos tres tipos de cifrado o criptografía se usan de una u otra forma en el mundo de las criptomonedas y criptodivisas de las Blockchain públicas y privadas.

Criptografía Simétrica. La criptografía simétrica utiliza una sola clave tanto para cifrar el mensaje como para descifrarlo. En los primeros pasos de esta disciplina, la seguridad de los mensajes cifrados se basa en el uso de una clave, y solo aquellos que la conocen pueden descifrar el mensaje. Este tipo de claves son muy extensas y se suelen generar de formas aleatorias, y para poder descifrar se crean dispositivos que permiten el almacenamiento de estas claves, y son conocidas como hardware wallets (Preukschat, 2017).

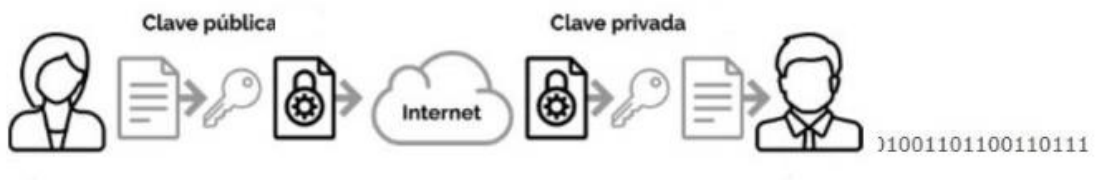
Para el cifrado con clave simétrica se usa el algoritmo AES (Advanced Encryption Standard) que utiliza claves de 256 bits. Esto es un dato crítico de la determinación de la seguridad del cifrado ante ataques de fuerza bruta. Es por tanto que este tipo de cifrados, es una forma muy segura para asegurar archivos y datos empresariales.

Criptografía Asimétrica. La criptografía asimétrica consiste en una solución mixta que usa una clave pública y privada. Estas dos claves son creadas y vinculadas entre sí mediante funciones especiales. Esas funciones especiales calculan la clave pública a partir de una clave privada, que se genera de forma aleatoria. Esta clave privada generatriz se guarda en secreto, mientras que la pública será de conocimiento popular, con esta clave pública cualquier podrá cifrar los mensajes secretos que envían y solo el que conozca la clave privada podrá descifrar dichos mensajes (Preukschat, 2017).

En la **Figura 4**, se puede notar que se envía un mensaje secreto al receptor, se utiliza la clave pública para cifrar el mensaje, se lo envía al receptor, y este mediante la clave privada puede descifrar dicho mensaje. Por lo tanto, si se conoce la clave privada se puede descifrar el mensaje, mientras que al revés no es posible, por lo tanto, es una función unidireccional o trampa de un solo sentido.

Figura 4.

Diagrama de explicación de la criptografía asimétrica.



Nota: En la figura se muestra el diagrama de explicación de la criptografía asimétrica, por (Preukschat, 2017).

La clave privada es un número aleatorio de tamaño extenso, por lo que hace que probabilísticamente resulte imposible generar otra igual, a partir de esta se calcula la clave pública usando un algoritmo, puede ser RSA o ECDSA (Preukschat, 2017).

Hash. Es un verbo en ingles que significa picar o moler. Esta corresponde a una expresión gráfica, pues consiste en picar los contenidos del mensaje, hasta obtener una secuencia de caracteres de longitud fija, creando una huella digital de un mensaje o documento. En la práctica hash consiste en la aplicación de una función matemática a los datos. Siempre que se aplique la misma función al mismo contenido, se obtendrá el mismo hash; esto implica que si el contenido del documento se modifica o se corrompe el hash cambiará por completo (Preukschat, 2017).

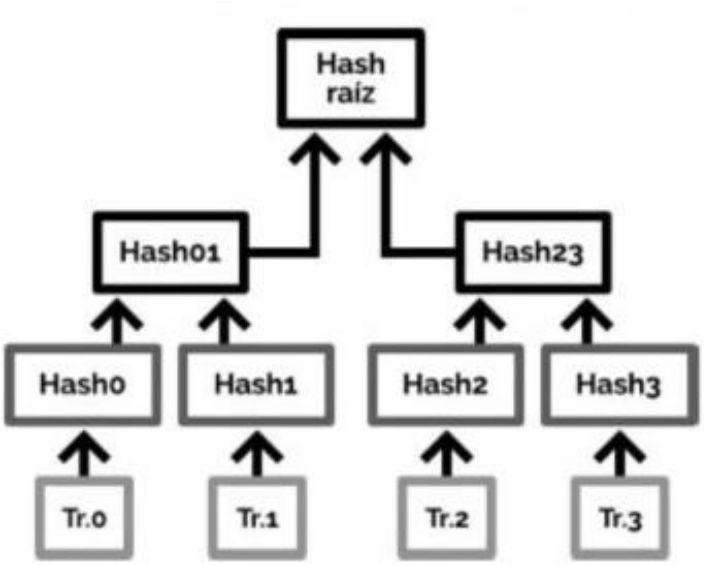
Esta característica hace que los métodos hash sean utilizados para aplicaciones en criptografía, en la generación de firmas digitales mediante algoritmos de autenticación, esto con el objetivo de verificar la integridad de los datos. Por lo tanto, el

propósito de un hash es el de permitir comprobar la integridad y seguridad de los datos y corroborar que no haya sido alterado (Preukschat, 2017).

Las funciones utilizadas para realizar hash, se distinguen por sus características fundamentales: eficiencia del cálculo, resistencia a la pre imagen y resistencia a segunda pre imagen y colisión. A partir de este concepto se han desarrollado tecnologías que permitan hacer del hash un método más seguro, y es donde surgen protocolos que permiten aplicar más de una función a un mensaje, permitiendo de esta manera aumentar la seguridad de los datos. Uno de estos métodos es el llamado Merkle Tree que se puede apreciar en la **Figura 5**, que es una formación piramidal de hashes que permiten aplicar funciones hasta llegar a un nodo raíz o Merkle Root. Estos árboles permiten verificar eficientemente la integridad de los datos, además que se pueden realizar a una gran cantidad de datos simultáneos.

Figura 5.

Merkle Tree o árbol de hash.



Nota: La figura muestra el Merkle Tree o árbol de hash, por (Preukschat, 2017).

Identificadores para redes descentralizadas (DID)

Para poder entender los identificadores dentro de las redes descentralizadas, es necesario partir del concepto de identificador. Los identificadores están compuestos por tres ejes principales: relaciones, atributos y agentes. Estos tres conceptos permiten establecer una arquitectura entre los datos, formando redes de tipo centralizado, federado y descentralizado (Fernández, 2018).

Las redes descentralizadas permiten conexiones punto a punto, evitando una organización central y una infraestructura que controlo y almacene todos los datos de la red. Los registros y datos se almacenan en cadenas de bloques y la definición de cada uno de estos es unívoca. Este conjunto de características que definen al objeto y a los datos se llaman identificadores descentralizados (DID) (Preukschat, 2017).

A diferencia de los métodos convencionales, los identificadores para redes descentralizados (DID) permiten que la generación y gestión de los datos no recaiga en entidades únicas, y permite que cada individuo ejerza cierto control en sus datos, aumentando la seguridad y la fidelidad de estos. La principal ventaja que proveen este tipo de identificadores es la seguridad que brindan a los participantes y a los activos que forman parte del ecosistema; el uso de claves privadas permite el registro criptográfico de los datos, permaneciendo al usuario proteger los datos y evitando robo de información (Fernández, 2018).

IPFS

InterPlanetary File System o Sistema de Archivos Interplanetario (IPFS) es un proyecto creado con el objetivo de crear una red de computadores al alcance mundial, que pudiera permitir el almacenamiento de la información de manera descentralizada, con elevada escalabilidad y con gran seguridad. Este sistema de archivos distribuidos

busca conectar todos los ordenadores con el mismo sistema de archivos, permitiendo de esta manera varios puntos de acceso a la información, desde cualquier parte (Barrios, 2017).

IPFS funciona mediante nodos; los nodos son identificados mediante hash con llaves públicas, cada nodo guarda su llave pública y privada. Cuando un nodo se quiere comunicar con otro, se realiza un intercambio de llaves públicas, y luego de comprobar la existencia de estas se realiza la comunicación. Los hashes utilizados en estos nodos utilizan el formato multihash, en el cual existe un header que especifica las funciones que se han utilizado y el largo de la misma, permitiendo tomar mejores decisiones en torno al compromiso entre performance y seguridad (Barrios, 2017).

El protocolo de red de IPFS incluye servicios para confiabilidad, conectividad, transporte, integridad y autenticidad. Posee además un sistema de enrutamiento que permite encontrar las direcciones de peers de la red y peers de objetos particulares. En cuanto al intercambio de datos, la distribución se realiza mediante intercambio de bloques entre peers, utilizando protocolos basados en BitTorrent. Esto permite que exista un sistema P2P masivo para albergar y distribuir bloques de manera rápida y robusta (Barrios, 2017).

CID

Cuando se intercambia datos con pares en las redes descentralizadas, se depende del direccionamiento de contenido. Aquí surgen los identificadores de contenido CID. Los CID son especificaciones que se originan en IPFS, son identificadores de contenido direccionado que se describen a sí mismos. Estos indicadores no indican donde se almacena el contenido, pero forma una especie de dirección que se basa en el contenido en sí. La cantidad de caracteres en un CID

depende del hash criptográfico del contenido subyacente, más que del tamaño de contenido (ProtoSchool, 2019)

Como estos se basan en las IPFS, la mayoría de los CID tienen tamaño sha2-265 (256 bits), haciendo de estos fáciles de administrar, sobre todo cuando se trata de varios contenidos.

El primer paso para crear un CID es transformar los datos de entrada, utilizando un algoritmo criptográfico que mapea la entrada de tamaño arbitrario a la salida de un tamaño fijo. Este tipo de transformación se conoce como resumen de hash criptográfico o solo hash. El algoritmo criptográfico utilizado debe generar hashes que tengan las siguientes características (Hernández Juárez, 2020):

- Determinista: la misma entrada siempre debe producir el mismo hash.
- Sin correlación: un pequeño cambio en la entrada debería generar un hash completamente diferente.
- Unidireccional: no debería ser factible reconstruir los datos a partir del hash.
- Único: solo un archivo puede producir un hash específico.

Blockchain

Definición. Blockchain consiste en una estructura capaz de almacenar información de manera distribuida haciendo uso de herramientas criptográficas para brindar seguridad. Es una base de datos descentralizada que no puede ser alterada, y permite la existencia de una serie de factores compartidos entre partes que no confían plenamente unas de las otras. Se realiza un consenso entre los nodos de la red de manera que sean los participantes quienes designan una información como válida, teniendo la opinión de la mayoría como el valor verdadero (Preukschat, 2017).

Evolución. La aparición de la Blockchain transaccional Bitcoin marca el punto histórico en el que este tipo de redes fueron puestas en producción en el año 2009. Esta nueva red trae consigo un paradigma con una serie de conceptos nuevos en el mundo de la informática como sucede con la minería de datos, una atracción de la minería en el mundo real, para crear un sistema de recompensas por mantener la red. A partir de este primer uso, se ha ido popularizando la arquitectura descentralizada y por tanto la aplicación de Blockchain para el tratamiento de los datos mediante bloques.

A partir de este concepto, el término se ha extendido para referirse a todo un ecosistema que incluye nuevos protocolos, tecnologías contables distribuidas, proyectos en individuos, en consorcios y empresas. En 2012 se desarrolla un sistema de pago directo llamado Ripple, basado en DLT alternativa y una comunidad corporativa, que actualmente es usado por varios bancos y redes de pagos, basados en la tecnología Blockchain. En 2013 se lanza NXT que fue el primer sistema Blockchain en implementar consenso de tipo Proff-of-Stake, que era un distribuidor de sus propias ciber monedas (Trujillo, 2017)

En 2015 la empresa tecnológica R3 lanza su plataforma de Blockchain privada llamada Corda, que comienza a liderar un consorcio inicial de entidades financieras. A partir del lanzamiento en 2015 de Ethereum e Hyperledger los sistemas Blockchain extienden su capa de aplicaciones y buscan un sinnúmero de aplicaciones en todo tipo de sectores. A día de hoy, la tecnología Blockchain es un sistema en construcción, en donde cada vez más las corporaciones e industrias empiezan a utilizar este tipo de redes descentralizadas, permitiendo una transformación digital (Trujillo, 2017)

Propiedades y características. Las propiedades principales de una Blockchain robusta son la disponibilidad y la persistencia. La disponibilidad asegura que una

transacción honesta que haya sido emitida termine siendo añadida a una cadena de bloques, evitando de esta manera que se produzca una denegación del servicio por parte de nodos corruptos. La persistencia hace referencia a cuando un nodo da una transacción como estable, el resto de nodos validarán esta transacción como estable y la harán inmutable (Dolores, 2017).

Para poder asegurar y cumplir con la propiedad de disponibilidad, la Blockchain implementa una red de nodos interconectados que interactúan como iguales. La red P2P (peer-to-peer) puede permitir que todos los nodos interactúen o también tener una White list en donde solo los nodos listados pueden participar (Dolores, 2017).

Se considera que una red Blockchain cumple con la propiedad de persistencia para cierto bloque, cuando existen seis bloques minados con posterioridad al mismo. Mediante esta regla se asegura que una transacción es inmutable con un riesgo inferior al 0.1%, suponiendo que un atacante posea más del 10% de la capacidad total del hash de la red (Dolores, 2017).

En cuanto a las características principales de Blockchain son (Dolores, 2017):

- Seguridad: Las cadenas de bloques almacenadas utilizan criptografía con lo cual los hackeos y robos de información se vuelven mucho más complicados.
- Trazabilidad: Permite recorrer la cadena de bloques y trazar todas las operaciones que se han realizado sobre una dirección determinada.
- Privacidad: Característica propia de Blockchain públicos en donde las direcciones no están ligadas a las identidades de las personas.
- Transparencia: Publicación de las reglas con las que se define el funcionamiento del Blockchain.

- Confianza: Característica que permite que dos personas que no confían entre si puedan realizar transacciones en Blockchain.

Algoritmos de consenso. Se trata del mecanismo mediante el cual una red Blockchain considera cierta información como verdadera, siendo básicamente la opinión de la mayoría de los participantes tomada como autentica. Razón por la cual una red es más segura mientras mayor número de nodos esta tenga. Los algoritmos de consenso en las redes Blockchain han ido evolucionando, buscando ser más seguros y se han ido adaptando a las necesidades del servicio de protocolo y de red (Amores, 2020).

Existen varios algoritmos de consenso, cada uno con sus determinadas características:

- Proof of Work (PoW)

Se trata del mecanismo más utilizado por las redes Blockchain, debido a que nació junto con Bitcoin, consiste en cuantificar el esfuerzo que los nodos realizan para resolver una determinada transacción, para ello se realizan complejos algoritmos matemáticos que implican carga de trabajo que debe ser recompensada con un costo representado por la criptomoneda de la red, siendo así un sistema de recompensas por mantener la red (Campaña, 2020).

- Proof of Stake (PoS)

Tiene un funcionamiento similar a PoW pero a diferencia evita generar una competencia entre nodos, en cambio aleatoriamente se selecciona un nodo para que sea el encargado de resolver el siguiente bloque. Este nodo se conoce como falsificador y se elige de manera determinista, de acuerdo a las participaciones que ha tenido en la red. Mientras más monedas, mayor poder tiene el minero. Este método permite ahorrar

energía al aprovechar un incentivo monetario en lugar de consumir muchos recursos computacionales (Campaña, 2020).

- Delegated Proof of Stake (DPoS)

En este método, los nodos testigos son los responsables de crear nuevos bloques y son recompensados, mientras que los nodos delegados son los encargados de mantener la red y sugerir cambios tales como el tamaño de los bloques, tarifas de transacciones o monto de recompensa. En cada ronda se eligen varios testigos, con votos más altos. Esto mejora el rendimiento y la latencia en comparación a PoS, convirtiendo a este protocolo en uno de consenso de bajo costo y con nivel de seguridad más bajo (Campaña, 2020).

- Proof of Activity (PoA)

Este método de consenso está basado en Prueba de Trabajo y Prueba de Participación, es robusto contra ataques que podrían difundir gran cantidad de bloques inválidos en la red. Los mineros intentan solucionar una función hash en una carrera para encontrar el siguiente bloque, sin embargo, el bloque resuelto únicamente contendrá un encabezado y la dirección del minero sin ninguna transacción. Después se agregan las transacciones al bloque y conforme con el encabezado del bloque resuelto, se selecciona un grupo de validadores para firmar el nuevo bloque con el fin de llegar a un acuerdo (Campaña, 2020).

- Proof of Burn (PoB)

Este método se basa en la quema de monedas, que significa el envío de monedas a una dirección irrecuperable. Al realizar este proceso a las monedas los mineros pueden mostrar su interés en la red, obteniendo de esta forma el poder de

minar y verificar las transacciones. Los usuarios tienen prioridad para resolver el siguiente bloque acorde con la cantidad de monedas que han quemado (Campaña, 2020).

PoB no usa ningún medio que no sean las monedas quemadas, o destruidas, garantizando así que la red permanezca dinámica y flexible. La desventaja que posee este método es la dependencia de la existencia de un marco monetario y la quema de monedas. Slimcoin es una criptomoneda que usa este protocolo de minado (Campaña, 2020).

- Proof of Capacity (PoC)

Es un protocolo básico de la criptomoneda Burst. En lugar de depender de una potencia en hardware de los mineros, se basa únicamente en la capacidad del disco duro, lo cual lo convierte en un método de consenso mucho más eficiente (Amores, 2020). Entre las monedas que emplean este método de consenso están SpaceCoin o actualmente conocida como SpaceMint y, PermaCoin (Campaña, 2020).

- Proof of Elapsed Time (PoET)

Nace en Intel y funciona de forma similar a PoW pero con consumo energético mucho menor. Los mineros deben resolver un hash al igual que en el método Prueba de Trabajo, el minero ganador o líder es elegido de manera aleatoria en función de un tiempo de espera obligatorio basado en el método. Una vez que termina el tiempo, el usuario ganador puede demostrar que se ha ejecutado el trabajo para extender la cadena de bloques (Campaña, 2020).

DApps

Las DApps son aplicaciones descentralizadas que se ejecutan en una Blockchain. Se ejecutan en servidores P2P distribuidos a través del mundo. El objetivo principal de estas DApps es el de generar aplicaciones cuyo almacenamiento y comunicación sea descentralizado y por tanto evitar que las aplicaciones desaparezcan cuando el que inicia el proyecto lo abandona. Se habla de DApps cuando cada usuario posee su propio servidor y este se puede comunicar y conectar con servidores de otros usuarios (Moreno, 2020).

Las DApps pueden tener infinito número de participantes, y sus beneficios se pueden aplicar en cualquier campo. Para poder ser considerada como una DApp se debe cumplir con ciertos factores y características (Moreno, 2020).

- Código Abierto: Debe ser 100% de código abierto
- Descentralizado: No debe poseer una entidad central que la maneje. Los cambios que respondan a las necesidades del mercado se realizan mediante solicitudes y debe ser aprobado por el consenso de sus usuarios. Las mejores propuestas serán aquellas que se implementen.
- Inmutable: Para evitar vulnerabilidades los datos deben ser registrados en forma de cadena de bloques.
- Incentivo: Se debe generar nuevos tokens mediante un algoritmo criptográfico para recompensar a usuarios de la red que colaboran con potencia de cálculo.

Casos de éxito de Blockchain

En la actualidad, el uso de Blockchain es algo cotidiano, permitiendo de esta manera incursionar en una gran cantidad de campos de aplicación, por lo tanto, se detalla ciertas aplicaciones de éxito en ciertos campos de interés:

Aplicaciones en educación. Blockchain puede resolver problemas de vulnerabilidad, seguridad y privacidad en el caso de entornos de aprendizaje y se puede utilizar para almacenar registros educativos relacionados con recompensas. El programa Sharples propone el uso de un sistema distribuido basado en Blockchain para el registro educativo y la reputación profesional. En este tipo de programas los profesores agregan bloques a la cadena de bloques que almacenan los logros de aprendizaje de estudiantes (Casino et al., 2019)

La gestión de certificados educativos también se puede mejorar mediante Blockchain, lo que mejora la seguridad y la confianza de los datos en infraestructuras digitales, y para la gestión del crédito (por ejemplo, relevante para la transferencia de acreditación europea y Sistema de acumulación). Además, las aplicaciones basadas en Blockchain podrían mejorar la acreditación digital del aprendizaje personal y académico (Casino et al., 2019)

También se podrían establecer centros de información escolar habilitados para Blockchain para recopilar, informar y analizar datos sobre los sistemas escolares para apoyar la toma de decisiones. Finalmente, en el caso de la publicación académica, Blockchain se puede utilizar para manejar mejor los envíos de manuscritos y para realizar revisiones adecuadas de manera oportuna o para la verificación de manuscritos (Casino et al., 2019)

Aplicaciones gubernamentales. A los gobiernos a lo largo de los años se les ha confiado la gestión y el mantenimiento de registros oficiales tanto de ciudadanos como de empresas. Las aplicaciones habilitadas para Blockchain pueden cambiar la forma en que los gobiernos a nivel local o estatal operan al no ser intermediario en transacciones y mantenimiento de registros. La responsabilidad, la automatización y la

seguridad que Blockchain daría a los registros públicos podrían eventualmente obstruir la corrupción y hacer que los servicios gubernamentales sean más eficientes (Casino et al., 2019)

En particular, Blockchain podría servir como una plataforma de comunicación segura para la integración física, social y empresarial. Al crear infraestructuras en un contexto de ciudad inteligente la utilización de Blockchain tiene como objetivo proporcionar los mismos servicios que son ofrecidos por el estado y sus autoridades públicas de manera descentralizada y eficiente, manteniendo la misma validez (Casino et al., 2019).

Ejemplos de tales servicios incluyen registro o documentos legales, atestación, identificación, contratos matrimoniales, impuestos y votaciones. El proyecto World Citizen es un ejemplo de un servicio de pasaportes descentralizado para identificar a los ciudadanos de todo el mundo. Las cadenas de bloques también se pueden utilizar para otros servicios públicos, como el registro de matrimonios, la gestión de patentes, y sistemas de impuestos sobre la renta (Casino et al., 2019).

Del mismo modo, Holacracy es programa personalizable de práctica de autogestión para organizaciones, donde la autoridad y la toma de decisiones se distribuyen a través de la auto organización de equipos en lugar de depender de un entorno típico de organización jerárquica (Casino et al., 2019).

Aplicaciones en la industria. Blockchain tiene el potencial de convertirse en una fuente significativa de innovaciones en los negocios y la gestión mediante la mejora, la optimización y la automatización de los procesos comerciales. Muchos modelos de comercio electrónico están surgiendo basados en IoT y Blockchain. Se puede encontrar un ejemplo en la creación de un modelo electrónico de negocio basado

en protocolo bitcoin, donde los autores proponen una empresa modelo en el que las transacciones entre dispositivos se realizan utilizando SC en una base de datos distribuida basada en Blockchain (Casino et al., 2019).

Otro ejemplo es la creación un sistema de preservación de la privacidad que utiliza una red IoT y Blockchain para probar la procedencia de fabricación de materiales sin la autenticación de terceros. Las aplicaciones de Blockchain parecen ofrecer una mejora considerable del rendimiento y oportunidades de comercialización, mejorando la credibilidad en el comercio electrónico y permitiendo a las empresas de IoT optimizar sus operaciones, ahorrando tiempo y dinero (Casino et al., 2019).

Las aplicaciones basadas en Blockchain podrían servir como negocio descentralizado y presentarse como un sistema de gestión de procesos para varias empresas. En tales casos, cada instancia de proceso empresarial puede mantenerse en la cadena de bloques, y los SC podrían realizar el enrutamiento del flujo de trabajo, lo que agiliza y automatiza procesos intra organizacionales y reducción de costos (Casino et al., 2019).

Aplicaciones para la seguridad de la información. Las organizaciones centralizadas, tanto públicas como privadas, acumulan grandes cantidades de información personal y confidencial. Aunque existen entes que tiene como objetivo regular el procesamiento de estos datos, todavía hay un gran vacío por cubrir. Blockchain presenta una oportunidad para mejorar los aspectos de seguridad de los macrodatos y su escalabilidad combinado con otros sistemas de almacenamiento eficientes que implementan métodos de minería de datos. Por tanto, la privacidad y las aplicaciones orientadas a la seguridad que se basan en la tecnología Blockchain son muchas (Casino et al., 2019).

Una de ellas es Namecoin es una tecnología de cadena de bloques de código abierto que implementa una versión descentralizada de DNS. El principal beneficio de un enfoque de DNS descentralizado son la seguridad, la resistencia a la censura, la eficiencia y la privacidad. Otro de los proyectos es Alejandría, un proyecto de código abierto basado en Blockchain que proporciona una biblioteca segura y descentralizada de cualquier tipo de medio permitiendo la libertad de expresión. Ambos sistemas utilizan servicios de identidad digital que pueden confirmar las identidades de un individuo (por ejemplo, utilizando seudónimos), lo que permite la seguridad y el anonimato en una verificación estandarizada modelo (Casino et al., 2019).

Smart Contracts

Se tratan de un conjunto de instrucciones que se pueden ejecutar en un entorno de Blockchain, son autoejecutables, es decir, su ejecución es automatizada. Los contratos inteligentes se pueden utilizar como acuerdos entre dos o más partes y definen condiciones previamente establecidas (Ramirez Valencia, 2019)

Los contratos inteligentes están escritos en códigos de programación, que ejecutan acciones de manera autónoma y automática, en el cual se definen reglas y consecuencias, de la misma manera que un contrato tradicional, con la diferencia de que no existe una intervención humana ni centralizada, evitando problemas de inseguridad, robo de información, coimas y falacias (Ramirez Valencia, 2019)

Los Smart Contracts tienen una versatilidad importante para adaptarse a las necesidades requerida, a parte de las ya conocidas aplicaciones financieras. Entre los casos de uso más conocidos se encuentran las de automatizar procesos legales, teniendo de esta manera un mecanismo más eficiente de realizar las actividades.

Debido a su popularidad son cada día más las plataformas o proyectos que son compatibles con contratos inteligentes, ente estos podemos encontrar: Ethereum, Hyperledger, Counterparty, Rootsock o Corda son alternativas capaces de desplegar y ejecutar contratos inteligentes en su entorno. Se debe tener en cuenta que hasta la actualidad las aplicaciones de los Smart contracts son exploratorias, por lo que a medida que aparezcan soluciones y casos de uso para los mismos se irá descubriendo su verdadera utilidad (BBVA, 2020).

Herramientas basadas en Blockchain

Ethereum. Ethereum fue creada por Vitalik Buterin, programador y escritor ruso, y también cofundador de Bitcoin Magazine1, que a finales de 2014 comenzó con el desarrollo de Ethereum. El propósito de Ethereum es crear una plataforma que pueda facilitar y permitir a otros programadores la creación de aplicaciones descentralizadas con Smart Contracts como base de estas (Miranda Palacios, 2018).

Al mismo tiempo Ethereum sirve como plataforma mundial donde se ejecutan estas aplicaciones. Tiene su propia criptomoneda llamada Ether (ETH), el medio con el cual se impulsa la plataforma. Esta criptomoneda es la monedan utilizada por los clientes de la Blockchain de Ethereum. Además de ser una moneda similar a Bitcoin para poder realizar pagos a personas o entidades, también es usado en el desarrollo de los Smart contracts, es decir, Ether es el incentivo que asegura que los programadores escriban aplicaciones de calidad y puedan recibir la recompensa a los mineros que van incluyendo los bloques en la cadena (Miranda Palacios, 2018).

En Ethereum se manejan además ciertos conceptos importantes para su aplicación, una de ellas es Gas. Los bloques tienen un tamaño limitado y cada bloque tiene un límite de gas que es establecido colectivamente por los mineros y la red para

evitar que un tamaño de bloque arbitrariamente grande suponga una carga menor para el nodo completo en términos de espacio en disco y requisitos de velocidad. Una forma de conceptualizar el límite de gas de bloque es pensar en él como el suministro de espacio de bloque disponible en el que realizar transacciones por lotes. Luego está el gas real que se usa diariamente para pagar la computación realizada en la cadena Ethereum (es decir, enviar una transacción, llamar a un contrato inteligente, acuñar un NFT) (Ethereum, 2020).

El desarrollo de Ethereum se da gracias a la evolución de la web hasta llegar a la más reciente web 3.0, o también conocida como web semántica, en donde en conjunto con otras tecnologías, paradigmas y conceptos recientes como Big Data, Algoritmos de inteligencia artificial, Web Scrapping, etc. Logran no solo encontrar información almacenada, sino entenderla y transformarla (Ethereum, 2020).

Dentro de las características principales de la Web 3.0 están: Registro de historial de usuario, análisis de datos, personalización de la web dependiendo del usuario, interoperabilidad, geolocalización, búsqueda inteligente, entre otros (Miranda Palacios, 2018).

Ethereum utiliza las características y la tecnología de la Blockchain para crear su infraestructura y para poder desarrollar y evolucionar las aplicaciones y servicios centralizados a un mundo descentralizado. Ethereum también utiliza el mecanismo de Proof-of-Work para la creación de nuevos bloques en la Blockchain. Pero desde el año 2017 se plantea que Ethereum cambie su mecanismo de minado al conocido como Proof-of-Stake o prueba de participación (Miranda Palacios, 2018).

A diferencia del Proof-of-Work, donde se recompensa a los mineros que resuelven problemas matemáticos validando las transacciones y creando nuevos

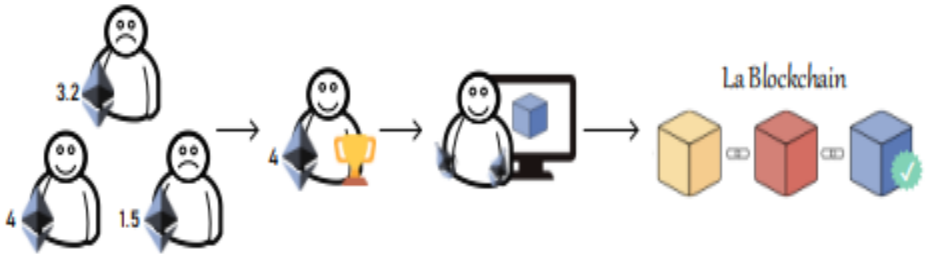
bloques, Proof-of-Stake como se puede ver en la **Figura 6**, permite que el creador de un nuevo bloque sea elegido de manera aleatoria. Proof-of-Stake se basa en un sorteo. Para participar en él, se debe bloquear tanta cantidad de Ether como se desee. La cantidad de Ether bloqueada se usará para escoger el minero del bloque; cada Ether tiene un identificador a sus y cuanto más Ether es bloqueado, más posibilidades hay de ser elegido. Para escoger el ganador del sorteo, aleatoriamente se seleccionará usando alguno de los valores aleatorios de un bloque y no será hasta bloques más adelante que el usuario escogido mine el bloque (Miranda Palacios, 2018).

En cuanto al Ethereum Virtual Machine (EVM) no hace referencia a una nube o una ola del océano, esto significa que EVM existe como una sola entidad mantenida por miles de computadoras conectadas que ejecutan un cliente Ethereum. El funcionamiento de esta máquina virtual debe ser continuo e ininterrumpid, pues mantiene en operación tanto las cuentas como Smart Contracts desplegados. En cualquier bloque de la cadena, Ethereum tiene uno y solo un estado 'canónico', y el EVM es lo que define las reglas para calcular un nuevo estado válido de un bloque a otro (Ethereum, 2020).

El ejecutar la Ethereum Virtual Machine implica correr una máquina de estado que tiene memoria transitoria. Los contratos, sin embargo, contienen una triada de almacenamiento de Merkle como se puede ver en la **Figura 6**, que hace referencia a una matriz de palabras direccionables, asociado con la cuenta en cuestión y parte del estado global. El código de bytes de contrato inteligente compilado se ejecuta como una serie de códigos de operación de EVM, que realizan operaciones de pila estándar como XOR, AND, ADD, SUB, etc. El EVM también implementa una serie de operaciones de pila específicas de Blockchain, como ADDRESS, BALANCE, KECCAK256, BLOCKHASH (Ethereum, 2020).

Figura 6.

Mecanismo Proof-of-Stake



Nota: En la figura se muestra el mecanismo Proof-of-stake, por (Miranda Palacios, 2018).

Hyperledger. Hyperledger es un proyecto que se concibe con la idea de desarrollar la tecnología Blockchain en el ámbito empresarial, para lo cual mantiene una propuesta de garantizar la seguridad, confidencialidad de la información, confianza entre empresas, fortaleciendo la colaboración y el comercio. (REYES DELGADO, 2018).

Hyperledger se mantiene gracias al apoyo de grandes empresas, algunas de ellas muy conocidas en el mundo del desarrollo tecnológico, y como resultado se han creado variedad de proyectos entre los que podemos encontrar frameworks y herramientas cada una centrada en resolver una temática recionada al desarrollo de Blockchains, entre las cuales se encuentra Hyperledger Fabric, que brinda un stack bastante completo para el despliegue de redes privadas, o Hyperledger Burrow, que es capaz de desplegar contratos inteligentes de Ethereum, además de otras utilidades enfocadas al rendimiento, seguridad o diseño (REYES DELGADO, 2018).

Multi Chain. Multichain es una plataforma que permite a los usuarios diseñar, implementar y operar los registros distribuidos de Blockchain de manera sencilla, ágil y

rápida. El objetivo es el de crear Blockchain privadas, decidiendo de esta manera quien puede conectarse a la plataforma para enviar y recibir transacciones, crear bloques y activos. Otra de las funciones es la de decidir la apertura de la Blockchain y hacerla pública. Esta plataforma permite mantener el registro de las transacciones realizadas en la red del usuario (Casas, 2019).

Este tipo de plataformas permite a entidades bancarias, empresas y usuarios desarrollar su propio sistema basado en Blockchain, son sus propios permisos y la eficiencia que sea necesaria, mediante compatibilidad con Bitcoin Core, y compartiendo características técnicas con los mismos. Esto hace que muchas de las creaciones sean más eficientes, seguras y rápidas dependiendo de la aplicación de las mismas, facilitando los procesos específicos que pueda tener una entidad financiera o empresa (Casas, 2019).

Tabla comparativa. A continuación en la **Tabla 1**, se presenta una comparativa entre las herramientas basadas en Blockchain (Sandner, 2017)

Tabla 1.

Comparativa de herramientas Blockchain.

Característica	Ethereum	Hyperledger	MultiChain
Descripción de Plataforma	Plataforma de blockchain genérica	Plataforma de blockchain modular	Plataforma especializada distribuida.
Gobernancia	Desarrolladores de Ethereum	Basado en Linux	Basado en Linux
Modo de Operación	Sin permisos, públicos o privados	Privada, con permisos	Sin permisos, privadas
Consensos	Minado basado en proof of work (PoW)	-Niveles de transacción -Consensos específicos	-Niveles de transacción -Consensos específicos
Contratos Inteligentes	-Contratos inteligentes vía código	-Contratos inteligentes vía código	-Contratos inteligentes vía código -Contratos inteligentes legalizados
Moneda o Token	-Ether -Tokens vía contratos inteligentes	-Tokens vía chaincode	-Bitcoin -Ether -Entre otras.

Nota. Esta tabla muestra la comparativa de herramientas Blockchain con sus diferentes características basadas en (Sandner, 2017).

Estándares Blockchain

Los estándares son especificaciones sobre cómo se debe desarrollar una plataforma que use Blockchain, basados en acuerdos de varias entidades mundiales que permiten gestionar de manera adecuada las tecnologías basadas en Blockchain.

ERC. Los ERCs (Ethereum Request for Comments) son propuestas generadas por desarrolladores o la comunidad centrada en Ethereum, que se crean con el fin de impulsar esta plataforma de Blockchain, generando de esta manera interoperabilidad en el ecosistema y facilitando el intercambio de información entre aplicaciones que hacen

uso de tokens, mediante el uso de estándares. Para poder denominarlo ERC es necesario que pase por fases de consulta y revisión para su posterior aprobación (Rodriguez, 2018).

Token ERC-20. Token ERC-20 es uno de los estándares más utilizados y con una mayor relevancia debido a su gran interoperabilidad en el entorno Ethereum. Este estándar proporciona los métodos de transferencia de tokens, aprobación y autorización de uso de tokens a otras direcciones, transferencia de tokens desde otras direcciones Ethereum, consulta de balances actuales y consulta de cantidades de tokens posibles de utilizar. Se plantea la existencia de dos tipos de eventos que se activan al realizar una transferencia o aprobación, el transfer y approval (Miranda Palacios, 2018).

Para poder caracterizar este token, es necesario asignar valor a una serie de atributos en formas de variable. El parámetro obligatorio es el número total de tokens disponibles, y los demás como nombre, símbolo y el número de decimales son opcionales, pero ayudan a dar mayores detalles al token. Cada día se despliegan nuevos tokens ERC20 (Rodriguez, 2018).

Token ERC-721. Este es el estándar de token no fungible, denominado NFT (Non Fungible Token). Esto quiere decir que es único y no puede ser reemplazado por otro. Otra característica es su indivisibilidad, pues se mantiene como una sola unidad, y esto es lo que le diferencia con el token ERC20. Este tipo de token no es consumible, pero si intercambiable. Este tipo de token puede representar cualquier cosa, desde una propiedad de una obra hasta préstamos o multas (Rodriguez, 2018).

Cada token ERC-721 está identificado con un ID que es único y no modificable y también por una dirección de Ethereum. Este tipo de tokens además tiene atributos y métodos que permiten tener compatibilidad entre aplicaciones, aunque no haga el

mismo uso de los tokens. Este estándar permite transferencia segura de token, propiedad del token, y propiedad del receptor del token (Rodriguez, 2018).

ERC 165. ERC-165 es un Estándar de detección de interfaz, este estándar es capaz de detectar la interfaz del token y adaptar el comportamiento dependiendo de este interfaz. Esto resulta útil para tener una naturaleza multitoken, y simplifica el diseño de aplicaciones.

Este estándar permite:

- Identificar interfaces
- Permite la publicación de Smart contract mediante implementación de interfaces.
- Detectar implementaciones ERC-165 en Smart contracts.
- Detectar si la implementación de un Smart contract da una interface.

Para este estándar una interfaz es un conjunto de selectores para funciones según lo definido por Ethereum ABI. Este es un subconjunto de un concepto de interfaces de Solidity y permite la definición de tipos de retorno, mutabilidad y eventos. Este selector de interfaz permite la selección de la función en la matriz. A partir de estas interfaces se puede realizar la publicación de Smart contracts y ayuda a su implementación

Otros estándares. Existen otros tipos de estándares como (Rodriguez, 2018):

- ERC-777: es un nuevo estándar de token avanzado que implementa nuevas funciones de envío, permite controlar y rechazar tokens enviados y recibidos mediante funciones gancho.

- ERC-823: una mejora al ERC-20 que incorpora un servicio de intercambio, permitiendo pagos cruzados de tokens.
- ERC-918: es un estándar de token mineable que usa algoritmo Proof of Work, distribuyendo los tokens mediante modelo de Initial Mining Offering.
- ERC-998: es una mejora del estándar ERC-721 que permite a un dueño de token formar árboles de propiedad entre ellos, en base a la relación entre cada token, como si se tratara de un árbol genealógico.
- ERC-1080: es una extensión del ERC-20 que soporta una función de devolución, prevención de robo y recuperación de tokens.

Activos no fungibles

NFT (Non Fungible Tokens) conocidos como tokens no fungibles, son activos que cumplen la característica de una unidad individual que no puede ser granulado o subdividido en pequeñas partes. Esto quiere decir que el estándar le permite al token representar activos físicos o contratos que representen como bienes inmuebles, arte, casas y activos virtuales. Este estándar de token permite a los desarrolladores tokenizar la propiedad de cualquier tipo de dato (Guarín Cardona, 2019).

En el mundo de las Blockchain, estos tokens son considerados como activos únicos e irrepetibles, que solo pueden tener un poseedor, siendo muchas veces utilizados como objetos de colección, formando así toda una economía basada en la compra venta de estos artículos e incentivando a cada vez más coleccionistas, desarrolladores o artistas a participar en este movimiento (Guarín Cardona, 2019).

E wallets

Los E-Wallets, también conocidos como billeteras electrónicas o carteras digitales son una nueva forma de pago que ha tenido un gran crecimiento en los últimos años. Bancos y empresas vinculadas al mundo financiero tienen sus propias billeteras electrónicas, esto da la posibilidad a que los E-Wallets disponibles sean considerables (MyChoice2Pay, 2020).

Las E-Wallets permiten a los usuarios poder administrar su dinero de forma virtual, desde sus móviles y realizar pagos de formas simples en cualquier momento. Estas carteras digitales permiten agilidad en pagos, servicios y presentan comodidad y facilidad a los usuarios al momento de pagar. Esto hace que actualmente los comercios integren, cada vez más, pagos mediante internet integrando la tecnología E-Wallet (MyChoice2Pay, 2020).

El principal objetivo de la integración de E-Wallets al comercio es de poner a disposición del usuario aplicaciones sencillas que le permitan pagar de forma adecuada, rápida y segura desde sus casas; esto se logra mediante la vinculación de las cuentas bancarias y tarjetas activas a las billeteras electrónicas (MyChoice2Pay, 2020).

En el mundo de Ethereum y criptomonedas es importante tener una cartera para recibir y enviar Ether, Bitcoin, etc. Una de estas carteras es Metamask. Metamask posee extensiones para navegadores como Chrome, Firefox y Brave que ofrece una cartera en Ethereum y además en cada página ingresada adjunta la librería web3 permitiendo que cada aplicación descentralizada (DApp) pueda integrar Metamask, y de esta manera el usuario pueda usar la aplicación de una manera fácil e intuitiva (Yogaterol, 2017).

Al instalar la extensión de Metamask en nuestro navegador, se genera una cartera cifrada con una contraseña. La extensión genera un código mnemotécnico que posee la información de la cartera del usuario, se debe recordar o guardar muy bien dicho código para poder recuperar la cartera virtual en caso de que se cambie de navegador o por alguna razón pierdas el wallet del navegador, incluso si se pierde la contraseña puedes usar el código mnemotécnico para recuperarla (Yogaterol, 2017).

Nodos y clientes Ethereum

Ethereum al ser una red distribuida de computadoras (conocido como nodos) ejecuta software que puede verificar bloques y datos de transacciones. Al ser una red de tecnología Blockhain, Ethereum se comunica a través de sus nodos (Ethereum, 2020).

Para lograr la comunicación con un nodo de Ethereum, es posible usar un cliente alguns de los cuales se sugieren en la **Tabla 2**, cada uno de estos puede interactuar con las redes públicas de Ethereum y se encuentran desarrllados en diversos lenguajes de programación como Go, Rust, JavaScript, Python, C # .NET y Java. Estas implementaciones mantienen una especificación que explica el funcionamiento de la red Ethereum y Blockchain. Además de poder configurar un nodo a necesidad de los 3 tipos existentes como son: (Ethereum, 2020).

1. Completo: almacena datos completos, participa activamente en la red validando bloques, se recomienda para sistemas de altos requerimientos técnicos capaces de almacenar gigabytes de información.
2. Ligero: usa cabeceras para referenciar la información, almacenarla y validarla, se recomienda para equipos de bajas especificaciones técnicas.

3. Almacenamiento: se dedica solo al almacenamiento de grandes cantidades de información y mantiene datos históricos.

Tabla 2.

Nodos y clientes Ethereum

Cliente	Lenguaje	Sistema Operativo	Redes	Estrategia de sincronización
Geth	Go	Linux, Windows, macOS	Mainnet, Görli, Rinkeby, Ropsten	Rápida, Completa
OpenEthereum	Rus	Linux, Windows, macOS	Mainnet, Görli, Rinkeby, Ropsten	Deformada, Completa
Nethermind	C#, .NET	Linux, Windows, macOS	Mainnet, Görli, Rinkeby, Ropsten	Rápida, Completa
Besu	Java	Linux, Windows, macOS	Mainnet, Görli, Rinkeby, Ropsten	Rápida, Completa
Erigon	Go	Linux, Windows, macOS	Mainnet, Görli, Rinkeby, Ropsten	Rápida, Completa

Nota. Se detalla a continuación una tabla con algunos clientes (Ethereum, 2020).

Redes Ethereum de prueba

Ethereum como tal no es una red, sino una plataforma que establece un protocolo a través del cual puede funcionar una red, de manera que es posible desplegar redes independientes de la red principal, cada una de estas no puede interactuar entre sí, pero una cuenta que siga el protocolo Ethereum si puede funcionar

en múltiples redes, considerando que en cada una de estas mantendrá un estado diferente (Ethereum, 2020).

Las redes creadas a partir del protocolo de Ethereum pueden ser tanto públicas como privadas, de manera que en el caso de las públicas sean accesibles desde cualquier parte del mundo, sin una restricción para la participación en estas a los usuarios (Ethereum, 2020).

Además de la red principal o Mainet, existen redes públicas de prueba, que se han construido con el objetivo de probar previamente las aplicaciones y contratos inteligentes desarrollados en el entorno de Ethereum, comportándose de manera similar que la red principal pero siendo mucho más accesibles en cuanto al costo por transacciones y la manera de obtener balance de la moneda Ether (Ethereum, 2020).

Actualmente muchas de estas redes trabajan con el algoritmo de consenso de prueba de autoridad, en donde se delimita la cantidad de nodos que pueden validar transacciones a los más confiables. Dentro de estas redes se encuentra: Görli, Kovan, OpenEthereum, Rinkeby, Ropsten, Testnet (Ethereum, 2020).

En cuanto a las redes privadas, se consideran aquellas cuyos nodos tienen interacción entre sí pero no forman parte de las mencionadas redes públicas, teniendo los participantes sus propias reglas para participar en estas (Ethereum, 2020).

Al iniciar el desarrollo de una DApp, es recomendable realizarlo en un entorno controlado, que no implique dificultad en la obtención de los recursos necesarios para realizar pruebas, por lo que una red privada sería una buena opción utilizar una red de prueba de manera local (Ethereum, 2020).

Providers

Un proveedor (provider) es una abstracción de una conexión a la red Ethereum, que proporciona una interfaz concisa y coherente para la funcionalidad estándar del nodo Ethereum, permitiendo al usuario entender y realizar la programación de manera fácil y rápida. Existen varias bibliotecas que permiten la aplicación de dichos proveedores (Ethers, 2019).

La biblioteca ethers.js proporciona varias opciones que deberían cubrir la gran mayoría de los casos de uso, pero también incluye las funciones y clases necesarias para la subclasificación si es necesaria una configuración más personalizada. Existen varios tipos de proveedores. El proveedor predeterminado es la forma más segura y fácil de comenzar a desarrollar en Ethereum, y también es lo suficientemente robusto para su uso en producción. Este proveedor crea un FallbackProvider conectado a tantos servicios de backend como sea posible. Cuando se realiza una solicitud, se envía a varios backends simultáneamente (Ethers, 2019).

A medida que se van teniendo las respuestas de cada backend, se comprueba que estén de acuerdo. Una vez que se ha alcanzado una cantidad suficiente de backends se proporciona la respuesta a su solicitud. Esto garantiza que, si un backend se desincroniza o si se ha visto comprometido, sus respuestas se descartan en favor de las respuestas que coinciden con la mayoría.

Existen también tipos de proveedores de API, que consisten en ofrecer una API web para acceder a Ethereum Blockchain. Estos proveedores permiten conectarse a ellos, lo que simplifica el desarrollo, ya que no necesita ejecutar su propia instancia de nodos Ethereum. Sin embargo, esta dependencia de los servicios de terceros puede reducir la resistencia, la seguridad y aumentar la cantidad de confianza necesaria y para

mitigar estos problemas, es recomendable que se utilice un proveedor predeterminado (Ethers, 2019).

Otro de los proveedores es el FallbackProvider, es el proveedor más avanzado disponible en el entorno Ethereum. Utiliza un quórum y se conecta a varios proveedores como backends, cada uno configurado con una prioridad y un peso. Cuando se realiza una solicitud, la solicitud se envía a múltiples backends, elegidos al azar y los resultados de cada uno se comparan con los demás (Ethers, 2019).

Solo una vez que se haya alcanzado el quórum, ese resultado se aceptará y se devolverá a la persona que llama. De forma predeterminada, el quórum requiere el 50% (redondeado hacia arriba) de los backends para estar de acuerdo. El peso se puede utilizar para dar más influencia a un proveedor de backend (Ethers, 2019).

Validación de documentos electrónicos

Documentos electrónicos. Un documento electrónico se conoce como un elemento producido por una persona natural o jurídica que contiene información generada, enviada, recibida y almacenada por medios electrónicos, y que se mantiene en este medio durante todo su ciclo de vida. Muchos de los documentos no nacen ser electrónicos, pero durante el proceso de gestión, se transforman a electrónicos (Gonzáles, 2021).

Los documentos electrónicos tienen componentes que son únicos para cada uno, y son útiles para identificar su contenido, estructura, contexto. El primero es Metadatos que son los datos que describen el contenido y características de un documento, algunos embebidos en el documento y otros son capturados de manera separada dependiendo del proceso. El segundo se trata de la lista de control de acceso,

que contiene la especificación de quienes pueden acceder al documento y qué actividades pueden realizar sobre este (Gonzáles, 2021).

Identidades digitales. Las identidades digitales se definen como una construcción completa, personal y social. Esta construcción es ligada al desarrollo de habilidades digitales y a las actividades en la red. Las identidades digitales poseen ciertas características, como ser: social, subjetiva, valiosa, indirecta, compuesta, real y dinámica (Voutssas M, 2010).

Dentro de la identidad digital se encuentran atributos de la persona, que se clasifican como los datos personales. En la práctica las identidades digitales se manejan mediante firmas digitales (claves públicas y privadas). Es entonces que se plantea que uno de estos programas, sea en realidad una representación de nuestra identidad en Blockchain, que nos permita interactuar con otros Smart contracts dentro de la red o con otras identidades(Voutssas M, 2010).

Se pueden crear programas donde el usuario da permiso a alguien para que escriba y firme información en su programa o llamado de otra manera identidad. Por ejemplo, tras un proceso en un banco, les podríamos pedir que certifiquen el resultado en nuestra identidad en Blockchain. Nosotros, pese a ser los propietarios de dicha identidad, no podemos modificar nada, pues la información está firmada. Si se desea dar el alta a otro banco, se podría comprobar lo que el primer banco ha escrito sobre nosotros sin comprometer la privacidad, y en base a ello permitirnos onboarding automático (Voutssas M, 2010).

Por lo tanto, para que la identidad en Blockchain sea efectiva, es necesario un punto de conexión con el mundo físico. Por lo tanto, es necesario que alguien de fe de la identidad que se representa en Blockchain y que coincida con la identidad real.

Además, no sólo identidades personales pueden ser certificadas en Blockchain, también se pueden representar empresa, propiedades, entre otras.

Herramientas de validación de documentos electrónicos

En la actualidad existen una gran cantidad de documentos electrónicos, por lo cual se han desarrollado un sinnúmero de documentos fraudulentos, mediante programas que pueden generar copias exactas de identidades, credenciales y documentos en general, por lo tanto, surge la necesidad de programas y herramientas que permitan validar la fidelidad de los documentos electrónicos.

Credly. Es una plataforma de uso sencillo que tiene como finalidad verificar, dar crédito y certificar mediante la creación de badges y su posterior asignación a los usuarios. Este tipo de herramientas permite realizar otorgar verificaciones a títulos, archivos, certificados para utilizarlos a nivel mundial. Las insignias emitidas a través de Credly es una representación digital de un resultado de aprendizaje, de una competencia o experiencia. Estas son verificadas y compartidas de manera ágil y segura (Credly, 2021).

Estas insignias están vinculadas a metadatos que proporcionan el contexto y la verificación de las mismas, pudiendo compartirse mediante internet para poder obtener la máxima visibilidad y reconocimiento. Para esto se puede publicar mediante redes profesionales como LinkedIn.

Blockcerts. Consiste en un estándar creado para la verificación de registros de cualquier tipo, a través de la tecnología Blockchain. Es de código abierto y permite el almacenamiento de documentos de cualquier proveedor (Blockerts, 2018).

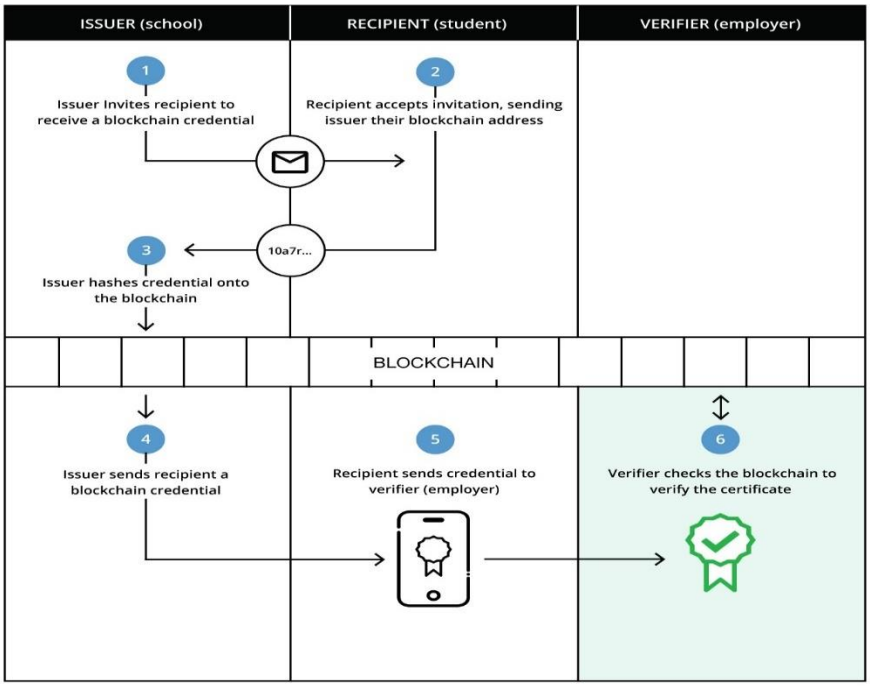
Blockcerts consta de bibliotecas, herramientas y aplicaciones móviles de código abierto que permiten un ecosistema en base a redes descentralizadas, basado en

estándares y centrado en el usuario, lo que puede llevar a una verificación sin confianza a través de tecnologías Blockchain. Blockcerts está asegura la privacidad de la identidad soberana de todos los participantes y permite al destinatario el control de sus reclamos a través de herramientas fáciles de usar como la billetera de certificados que es una aplicación móvil (Blockerts, 2018).

El flujo de Blockcerts para la emisión de documentos se basa en que el estudiante y emisor establezcan una comunicación para poder recibir certificados de éste, una vez se realiza esta operación el verificador puede validar la información, como se puede apreciar en la **Figura 7** (Blockerts, 2018).

Figura 7.

Diagrama Explicativo de cómo funciona Blockerts.



Nota: La figura muestra el diagrama explicativo de cómo funciona los Blockerts, por (Blockerts, 2018).

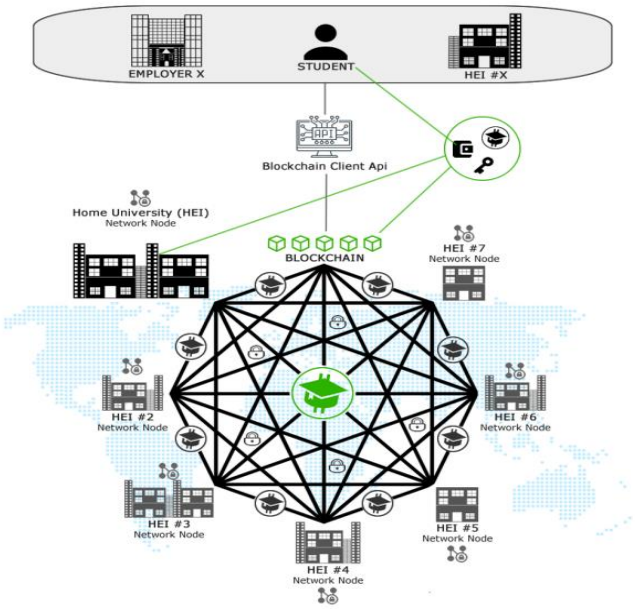
EduCTX. El sistema Edu CTX aprovecha los beneficios del Blockchain, como una arquitectura descentralizada, que ofrece seguridad, anonimato, longevidad, integridad, transparencia, inmutabilidad y simplificación del ecosistema global, con el fin de crear un sistema de calificación y crédito de educación superior confiable. La plataforma se basa en Ark Blockchain de código abierto (Turkanović et al., 2018).

EduCTX proporciona un modelo de arquitectura interoperable para el crédito de educación superior sistema que aborda un punto de vista unificado globalmente para estudiantes e instituciones. Los empleadores potenciales pueden beneficiarse del sistema propuesto pues las bases de datos están estructuradas para ser accedidas exclusivamente por un personal de la institución y en sistemas en línea dedicados, por lo tanto, con poca o ninguna interoperabilidad (Turkanović et al., 2018).

El sistema EduCTX se basa en una red distribuida peer-to-peer (P2P) que se puede ver en la **Figura 8**. Estos sistemas son flexible, seguro y resistente debido a su capacidad de almacenamiento e intercambio de recursos a escala mundial. El EduCTX transfiere el crédito y la calificación de la educación superior del mundo analógico y físico a una versión globalmente eficiente y simplificada basada en la tecnología de la cadena de bloques. La plataforma visualiza un nivel superior unificado, simplificado que permita un sistema educativo y de calificación global (Turkanović et al., 2018).

Figura 8.

Estructura de plataforma EduCTX.



Nota: En la siguiente figura se observa la estructura de plataforma EduCTX, por (Turkanović et al., 2018).

0Xcerts. 0xcert es un protocolo para certificar y validar emisores de tokens no fungibles. Hay muchos protocolos, aplicaciones y empresas diferentes que intentan llevar activos y certificaciones del mundo real a la cadena de bloques. Sin embargo, no existe un consenso claro sobre cómo validar la legitimidad de los tokens acuñados. Por ejemplo, no hay forma de garantizar que el emisor tenga derecho a acuñar tokens de identidad en Ethereum. Como las cadenas de bloques públicas no tienen permiso, cualquiera puede crear dichos tokens, pero solo los tokens creados por un proveedor de identidad legítimo serán valiosos (0xcert, 2020).

0xcert crea un estándar para tokens certificados no fungibles, Xcerts. Los desarrolladores pueden interactuar con este estándar a través de un SDK y API de alto

nivel. Además, 0xcert creará un registro descentralizado de emisores auténticos (0xcert, 2020).

Xertify. Es una empresa establecida en Colombia, que brinda servicios tecnológicos aplicando nuevas tendencias como Blockchain, y contando con una solución de certificación digital (Xertify, 2021).

La solución brindada por Xertify apunta a aumentar la productividad de las empresas ahorrando tiempo y recursos en el proceso de emisión de documentos digitales, además de brindar facilidades en la verificación a personas externas a las instituciones (Xertify, 2021).

Metodologías y técnicas para seguridad de documentos electrónico.

Existen una serie de técnicas y metodologías para asegurar los documentos electrónicos, para lo cual es indispensable entender los diversos conceptos que rigen la seguridad informática. Estas técnicas deben asegurar todo el proceso de generación del documento, es decir los recursos informáticos, el impacto, la vulnerabilidad, el riesgo y saber las posibles amenazas, garantizando de esta manera la continuidad de los servicios y reduciendo los riesgos para los recursos informáticos (Voutssas M, 2010).

En primer lugar, se recomienda como primera etapa, diseñar una estrategia de seguridad informática, estudiando tres fuentes principales, los procedimientos de la organización, los requisitos legales y la valoración de riesgos de la información. Este análisis permite estar en la capacidad de identificar los riesgos, establecer probabilidades de fallo, determinar medidas de seguridad y tomar decisiones preventivas (Voutssas M, 2010).

En segundo lugar, se debe establecer una estrategia para la construcción de la seguridad informática en la organización, por lo que se plantea utilizar estándares de

seguridad mundial, que se acoplen con la necesidad de la organización (Voutssas M, 2010).

Estándares de validación de credenciales electrónicas

- Open Badges

Se trata de un estándar que especifica una serie de entidades para identificar un determinado logro de aprendizaje, habilidad obtenida, entre otras competencias, describiendo una gran cantidad de información sobre la habilidad o conjunto de habilidades obtenidas y que puede representarse a través de una imagen en complemento con metadata (Open Badges, 2016).

Se pueden utilizar como credenciales reales, por lo tanto, se puede reconocer logros de todo tipo en estudiantes y profesionales, exportando en CV en todo el mundo. Los badges pueden ser otorgados por escuelas y universidades, cursos online, profesores y preparadores, empleadores, bibliotecas entre otros (Open Badges, 2016).

- Verifiable Credentials Data Model

Una credencial puede consistir en información relacionada con la identificación del sujeto, por ejemplo: una foto, nombre o número de identificación, también puede relacionarse con la autoridad emisora como el gobierno de una ciudad, una agencia nacional o un organismo de certificación o información relacionada con el tipo de credencial que es ya sea un pasaporte, una licencia de conducir o una tarjeta de seguro médico (Chadwick & Longley, 2021).

Una credencial verificable (Verifable Credential) puede representar toda la misma información que representa una credencial física. La adición de tecnologías, como las firmas digitales, hace que las credenciales verificables sean más evidentes y

más confiables que sus contrapartes físicas. Los titulares de credenciales verificables pueden generar presentaciones verificables y luego compartir estas presentaciones verificables con los verificadores para demostrar que poseen credenciales verificables con ciertas características (Chadwick & Longley, 2021).

Las credenciales verificables se pueden utilizar de manera más rápida, permitiendo la transmisión inmediata. Si bien esta especificación intenta mejorar la facilidad para la creación de credenciales digitales, también se trata de equiparar con una serie de objetivos para asegurar la privacidad. La persistencia de la información digital y la facilidad con la que se pueden recopilar y correlacionar fuentes de datos digitales constituyen un problema de privacidad, este problema puede ser mitigado mediante el uso de credenciales verificables (Chadwick & Longley, 2021).

Capítulo IV

Análisis, Diseño y Desarrollo del sistema

Análisis

El siguiente apartado describe los participantes que interactúan con el sistema, así como sus roles, características y responsabilidades; así mismo se describen los requerimientos del sistema y sus limitaciones.

Actores

Partiendo de la arquitectura de algunos de los proyectos mencionados anteriormente, tales como Blockcerts o EduCTX, los cuales son soluciones similares a la propuesta actual, probadas y alineadas a estándares como Open Badges, se logran identificar los siguientes actores fundamentales para un sistema de validación de certificados educativos.

- Estudiante. Se trata del actor interesado en adquirir los certificados educativos correspondientes a la finalización de un proceso de aprendizaje.
- Emisor. Consiste en el actor que se encarga de emitir certificados educativos a los estudiantes que hayan sido respectivamente validados por éste como aptos para recibir el documento mencionado. Además de revocar los certificados si éste lo considera necesario.
- Gestor de red. Se incluye este actor con la finalidad de tener un control sobre los emisores que se registren en la red, de manera que sea éste quien pueda gestionarlos.
- Revisor. Se trata del actor que puede acceder al sistema para verificar la validez de un certificado educativo y no requiere un registro previo.

Características de los usuarios

Los actores previamente descritos corresponden a un respectivo usuario del sistema, teniendo en consideración la definición de estos, se detallan sus características en la **Tabla 3.**

Tabla 3.

Características de los usuarios del sistema

Usuario	Características
Estudiante	Realiza su registro en la aplicación para poder recibir certificados educativos y visualizarlos en el sistema.
Emisor	Debe solicitar a un gestor de red su registro para poder interactuar con el sistema. Podrá seleccionar las opciones de visualizar y crear plantillas de certificados, emitir certificados, revocar certificados.
Gestor de red	Puede ingresar al sistema para visualizar y añadir emisores, gestores de red, listado de estudiantes.
Revisor	Podrá acceder al sistema para validar información sobre los certificados de los estudiantes, visualizar el estado y el archivo de evidencia del certificado.

Historias de usuario

Siguiendo con la metodología de desarrollo planteada Scrum, se desarrollan las historias de usuario como herramienta para conocer los requerimientos del sistema, así como las entradas y salidas del mismo para cada uno de los usuarios descritos en el apartado anterior, formalizando de esta manera las interacciones de estos en el sistema. Al ser el presente trabajo una propuesta de solución, los requerimientos han sido abstraídos de las lecturas anteriores y descritos por el autor.

- Registrar estudiante

La funcionalidad de registro de estudiante como historia de usuario se encuentra especificada en la **Tabla 4.**

Tabla 4.

Requerimiento registrar estudiante

Código	001
Título	Registrar estudiante
Descripción	Yo como estudiante quiero poder registrarme en el sistema para poder recibir certificados y visualizarlos.
Usuario	Estudiante
Datos de entrada	• Identificador del estudiante • Nombre del estudiante • Correo electrónico del estudiante • Número de contacto • País • Ciudad • Provincia o Estado
Datos de salida	• Mensaje de respuesta de registro
Observaciones	El identificador del estudiante dentro del sistema se convierte en una dirección personal de blockchain

- Visualizar certificados de estudiante

La funcionalidad de visualizar certificados de estudiante como historia de usuario se encuentra especificada en la **Tabla 5.**

Tabla 5.

Requerimiento visualizar certificados de estudiante

Código	002
Titulo	Visualizar certificados de estudiante
Descripción	Yo como estudiante quiero poder visualizar en el sistema los certificados que han sido creados por los emisores para mi persona.
Usuario	Estudiante
Datos de entrada	• Identificador del estudiante
Datos de salida	• Lista de certificados emitidos al estudiante
Observaciones	El identificador del estudiante dentro del sistema se convierte en una dirección personal de blockchain

- Confirmar propiedad del certificado

La funcionalidad de confirmar propiedad del certificado como historia de usuario se encuentra especificada en la **Tabla 6.**

Tabla 6.

Requerimiento confirmar propiedad del certificado

Código	003
Título	Confirmar propiedad del certificado
Descripción	Yo como estudiante quiero poder verificar que soy el poseedor del certificado.
Usuario	Estudiante
Datos de entrada	• Identificador del estudiante
Datos de salida	• Confirmación de la posesión del certificado
Observaciones	El identificador del estudiante dentro del sistema se convierte en una dirección personal de blockchain

- Registro de emisor

La funcionalidad de registro de emisor como historia de usuario se encuentra especificada en la **Tabla 7**.

Tabla 7.

Requerimiento Registro de emisor

Código	004
Título	Registro de emisor
Descripción	Yo como gestor de red quiero poder registrar los datos de un emisor, para que éste pueda interactuar con la emisión, visualización y revocación de certificados.
Usuario	Gestor de red
Datos de entrada	• Identificador del emisor • Nombre del emisor • Enlace web del emisor • Correo electrónico del estudiante • Número de contacto • Llave pública • País • Ciudad • Provincia o Estado
Datos de salida	• Mensaje de respuesta de registro
Observaciones	El identificador del emisor dentro del sistema se convierte en una dirección personal de blockchain

- Emitir certificados educativos

La funcionalidad de emitir certificados educativos como historia de usuario se encuentra especificada en la **Tabla 8.**

Tabla 8.

Requerimiento emitir certificados educativos

Código	005
Título	Emitir certificados educativos
Descripción	Yo como emisor quiero poder crear un certificado y que éste sea transferido al estudiante correspondiente
Usuario	Emisor
Datos de entrada	• Identificador del estudiante • Datos de la plantilla seleccionada • Fecha de expiración del certificado • Archivo de evidencia
Datos de salida	• Mensaje de respuesta de registro
Observaciones	El identificador del estudiante dentro del sistema se convierte en una dirección personal de blockchain

- Visualizar certificados emitidos

La funcionalidad de emitir visualizar certificados emitidos como historia de usuario se encuentra especificada en la **Tabla 9.**

Tabla 9.

Requerimiento visualizar certificados emitidos

Código	006
Titulo	Visualizar certificados emitidos
Descripción	Yo como emisor quiero visualizar los certificados que he emitido a los estudiantes
Usuario	Emisor
Datos de entrada	• Identificador del emisor
Datos de salida	• Lista de certificados emitidos al estudiante
Observaciones	El identificador del emisor dentro del sistema se convierte en una dirección personal de blockchain

- Crear plantillas de certificado

La funcionalidad de crear plantillas de certificado como historia de usuario se encuentra especificada en la **Tabla 10.**

Tabla 10.

Requerimiento crear plantillas de certificado

Código	007
Titulo	Crear plantillas de certificado
Descripción	Yo como emisor quiero poder crear una plantilla que sirva como base para la generación de un certificado.
Usuario	Emisor
Datos de entrada	• Identificador del estudiante • Datos de la plantilla seleccionada • Fecha de expiración del certificado • Archivo de evidencia
Datos de salida	• Mensaje de respuesta de registro
Observaciones	El identificador del estudiante dentro del sistema se convierte en una dirección personal de blockchain

- Visualizar plantillas de emisor

La funcionalidad de crear visualizar plantillas de emisor como historia de usuario se encuentra especificada en la **Tabla 11.**

Tabla 11.

Requerimiento visualizar plantillas de emisor

Código	008
Título	Visualizar plantillas de emisor
Descripción	Yo como emisor quiero visualizar las plantillas de certificados que he creado
Usuario	Emisor
Datos de entrada	• Identificador del emisor
Datos de salida	• Lista de plantillas creadas por el emisor
Observaciones	El identificador del estudiante dentro del sistema se convierte en una dirección personal de blockchain

- Revocar certificado

La funcionalidad de revocar certificado como historia de usuario se encuentra especificada en la **Tabla 12.**

Tabla 12.

Requerimiento revocar certificado

Código	009
Título	Revocar certificado
Descripción	Yo como emisor quiero poder revocar un certificado emitido previamente de manera que no conste como válido.
Usuario	Emisor
Datos de entrada	• Identificador del certificado • Fecha de revocación • Razón o motivo de la revocación del certificado
Datos de salida	• Mensaje de respuesta de registro
Observaciones	N/A

- Visualizar certificados revocados

La funcionalidad de visualizar certificados revocados como historia de usuario se encuentra especificada en la **Tabla 13.**

Tabla 13.

Requerimiento visualizar certificados revocados

Código	010
Título	Visualizar certificados revocados
Descripción	Yo como emisor visualizar cuales son los certificados que han sido revocados
Usuario	Emisor
Datos de entrada	• Identificador del emisor
Datos de salida	• Lista de certificados revocados
Observaciones	El identificador del emisor dentro del sistema se convierte en una dirección personal de blockchain

- Registrar gestor de red

La funcionalidad de registrar gestor de red como historia de usuario se encuentra especificada en la **Tabla 14.**

Tabla 14.

Requerimiento registrar gestor de red

Código	011
Titulo	Emitir certificados educativos
Descripción	Yo como gestor de red quiero poder registrar nuevos gestores de red que puedan acceder a las mismas funcionalidades
Usuario	Emisor
Datos de entrada	• Identificador del gestor de red
Datos de salida	• Mensaje de respuesta de registro
Observaciones	El identificador del gestor de red dentro del sistema se convierte en una dirección personal de blockchain

- Visualizar gestores de red

La funcionalidad de visualizar gestores de red como historia de usuario se encuentra especificada en la **Tabla 15.**

Tabla 15.

Requerimiento visualizar gestores de red

Código	012
Título	Visualizar gestores de red
Descripción	Yo como emisor quiero visualizar los gestores de red registrados en el sistema
Usuario	Emisor
Datos de entrada	• Identificador del gestor de red
Datos de salida	• Lista de gestores de red registrados
Observaciones	El identificador del gestor de red dentro del sistema se convierte en una dirección personal de blockchain

- Validar certificado

La funcionalidad de validar certificado como historia de usuario se encuentra especificada en la **Tabla 16.**

Se debe tener en cuenta que el usuario revisor no necesitará de autenticarse de alguna manera en el sistema para realizar sus actividades.

Tabla 16.

Requerimiento validar certificado

Código	013
Título	Validar certificado
Descripción	Yo como revisor quiero poder verificar la validez de un certificado correspondiente a un estudiante
Usuario	Emisor
Datos de entrada	• Identificador del certificado
Datos de salida	• Información sobre estado del certificado, el archivo y la data respectiva
Observaciones	N/A

- Visualizar archivo de evidencia de certificado

La funcionalidad de visualizar archivo de evidencia de certificado como historia de usuario se encuentra especificada en la **Tabla 17.**

Tabla 17.

Requerimiento visualizar archivo de evidencia de certificado

Código	014
Título	Visualizar archivo de evidencia de certificado
Descripción	Yo como revisor quiero poder ver el documento electrónico de evidencia correspondiente a un certificado educativo
Usuario	Emisor
Datos de entrada	• Identificador de contenido del certificado
Datos de salida	• Enlace de descarga hacia la el documento electrónico almacenado
Observaciones	N/A

Alcance del prototipo

El prototipo se desarrolló en un entorno de pruebas, en donde se prioriza el proceso de validación de los certificados educativos como eje central, buscando crear una herramienta útil para su creación a través de plantillas y ofrecer una interfaz de rápido uso para el usuario revisor. Se considera el uso de librerías, frameworks y estándares basados en la tecnología Blockchain y orientados al propósito de verificación de certificados educativos. Se desconsidera a los usuarios descritos anteriormente para crear un sistema al que se pueda tener acceso a través del registro del usuario estudiante directamente en la interfaz del sistema, y los usuarios emisor y gestor de red a través de este último. El usuario revisor no necesita registro previo para realizar sus interacciones con el sistema.

Al ser un entorno que interactúa con protocolos web se procedió a desarrollar una aplicación accesible a través del navegador.

Limitaciones

No se considera dentro del prototipo el proceso de obtención de los datos de los emisores por parte del usuario gestor de red, esto debido a que para el actual trabajo es un flujo que no se considera dentro del alcance.

La emisión de los certificados educativos a los estudiantes se realizó de acuerdo a la información disponible en el sistema, no considerando para las pruebas la relación en el mundo real entre las entidades emisoras y los estudiantes.

La actualización de datos en la interfaz se llevó a cabo después de los procesos de almacenamiento de información del sistema por parte del usuario, sin embargo, no se considera validación en tiempo real para las acciones realizadas por otros usuarios.

Definiciones

Certificado educativo: Dentro del sistema se define entonces, como el conjunto de un documento electrónico de evidencia, los metadatos del mismo y los estados de éste.

NFT CERTS: Consiste en un prototipo de sistema de validación de certificados educativos basado en Blockchain.

Dirección de Ethereum: Consiste en un conjunto de caracteres alfanuméricos generados a través de un mnemónico, que sirven para identificar de manera única una cuenta de Ethereum asociada a una persona o entidad, y se usa como identificador para los usuarios del sistema.

CID: Conjunto de caracteres alfanuméricos generados al momento de ingresar un archivo a la red IPFS a partir del hash, y que identifica de manera única, un documento disponible en esta.

Diseño del sistema

Selección de herramientas de Blockchain para la solución

La información considerada en el prototipo sobre certificados, estudiantes y emisores se considera de carácter público, pues el objetivo es validar esta información la cual debe estar disponible para los revisores que no requieren un acceso o registro para acceder a ella, sino únicamente que los datos de entrada les sean facilitados previamente. Por lo que no se prevén problemas de confidencialidad de la información, además de que, para el actual prototipo inicial, no se consideran restricciones en la participación de la red que deba ser administrada de manera privada; por el contrario,

se proyecta que el presente trabajo escale a largo plazo de manera que pueda resultar útil para diferentes instituciones a nivel nacional e internacional.

Si bien en el alcance no se considera que los usuarios emisores sean representados por una entidad en particular, estos pueden corresponder a cualquier institución educativa, plataforma de cursos, entidad certificadora, entre otros. De manera que la mejor opción partiendo desde estos puntos es optar por una red pública, que permita a cualquiera participar en la red, pero siendo libre de elegir el grado de participación que tendrá en esta.

De esta manera aprovechando los beneficios de las redes públicas en los aspectos de seguridad, autonomía y trazabilidad, siendo parte de una red robusta, mantenida por gran cantidad de nodos, siendo este un gran punto a favor al usar Ethereum, pero teniendo en cuenta los posibles inconvenientes de escalabilidad y costes de transacciones que esto puede conllevar. Pues en las redes públicas estos costos pueden llegar a incrementar de gran manera debido a la cantidad de nodos y transacciones.

Como se estipuló anteriormente, Ethereum se crea con el objetivo de tener una plataforma exclusiva para la creación de aplicaciones descentralizadas basadas en el uso de Smart Contracts, los cuales pueden llegar a tener innumerables casos de uso, siendo una gran alternativa para desarrollar la lógica de los requerimientos con el uso de estos.

De esta manera se prefiere una infraestructura construida con Ethereum, que cumple con las características buscadas para el prototipo deseado y considerando el crecimiento del sistema a futuro.

IPFS

Se decide trabajar con este sistema de archivos con el objetivo de mejorar el procesamiento y almacenamiento de los documentos electrónicos de manera que en Blockchain se mantenga una referencia a estos, pues están disponibles a través de internet, y se evite la excesiva carga de datos en los contratos inteligentes que puede llegar a ocasionar altos costes de procesamiento y por ende mayores tarifas de gas que debe ser pagado con Ether. Además, se mantiene el paradigma de descentralización del sistema al ser una red peer to peer similar a Blockchain.

Otra ventaja dentro del esquema del presente trabajo es la generación del identificador CID, el cual se crea a partir del hash del archivo regresando éste como un identificador único y que sirve como hash de validación y a su vez enlace al documento en IPFS, por tanto, se adopta una estrategia de almacenamiento de estos CID dentro de Blockchain, así como la utilización de estos códigos para poder validar los certificados educativos que dentro del sistema fueron identificados con éste.

Cuentas de Ethereum

Las cuentas de Ethereum son de fácil obtención, existiendo la posibilidad de crearlas dentro de una billetera electrónica de Ethereum como Metamask, tienen gran utilidad al momento de realizar el intercambio de activos digitales como tokens, y con las recientes aplicaciones de los Ethereum request for comments, han ganado terreno debido a las características de interoperabilidad entre los sistemas, por lo que su utilidad es cada vez más evidente, y gracias a las billeteras electrónicas estas se pueden conectas con las aplicaciones e interactuar con ellas conectándose a la red elegida.

Teniendo lo anterior en cuenta, se utilizaron las cuentas de Ethereum como identificadores de los usuarios en el sistema, esta estrategia nos brinda las mismas utilidades de cualquier identificador que podría usarse en el registro para identificar a los participantes como nombres de usuarios o documentos nacionales de identidad, sin la necesidad de almacenar una contraseña dentro del sistema, pues cada cuenta debe encontrarse segura en la billetera del usuario a la cual solo éste debería tener acceso.

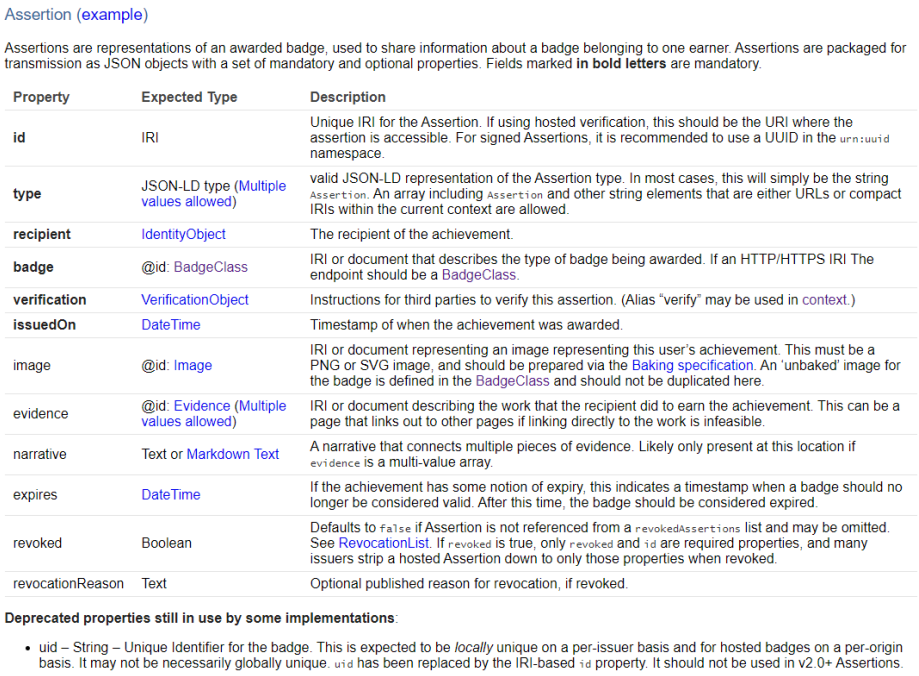
Open Badges

El estándar Open Badges cumple con muchos de los requerimientos de portabilidad, además de proveer una gran cantidad de información sobre las credenciales, logros conseguidos, emisores, habilidades, etc. Consiguiendo un buen entendimiento del aprendizaje del estudiante beneficiario por parte de los interesados en validar su conocimiento.

La última versión de Open Badges, disponible en su página oficial, describe un conjunto de entidades para almacenar información sobre los certificados emitidos, que pueden variar según los requerimientos de la solución, pero que mantiene ciertos datos fundamentales como obligatorios como se puede apreciar en el ejemplo de la **Figura 9**, los cuales fueron considerados en la construcción del prototipo en vista de alinear la solución al estándar. Así como las entidades fundamentales Assertion, BadgeClass, Profile, Issuer.

Figura 9.

Open Badges



Assertion (example)

Assertions are representations of an awarded badge, used to share information about a badge belonging to one earner. Assertions are packaged for transmission as JSON objects with a set of mandatory and optional properties. Fields marked **in bold letters** are mandatory.

Property	Expected Type	Description
id	IRI	Unique IRI for the Assertion. If using hosted verification, this should be the URI where the assertion is accessible. For signed Assertions, it is recommended to use a UUID in the `urn:uuid` namespace.
type	JSON-LD type (Multiple values allowed)	valid JSON-LD representation of the Assertion type. In most cases, this will simply be the string `Assertion`. An array including `Assertion` and other string elements that are either URLs or compact IRIs within the current context are allowed.
recipient	IdentityObject	The recipient of the achievement.
badge	@id: BadgeClass	IRI or document that describes the type of badge being awarded. If an HTTP/HTTPS IRI The endpoint should be a BadgeClass.
verification	VerificationObject	Instructions for third parties to verify this assertion. (Alias "verify" may be used in context.)
issuedOn	DateTime	Timestamp of when the achievement was awarded.
image	@id: Image	IRI or document representing an image representing this user's achievement. This must be a PNG or SVG image, and should be prepared via the Baking specification. An 'unbaked' image for the badge is defined in the BadgeClass and should not be duplicated here.
evidence	@id: Evidence (Multiple values allowed)	IRI or document describing the work that the recipient did to earn the achievement. This can be a page that links out to other pages if linking directly to the work is infeasible.
narrative	Text or Markdown Text	A narrative that connects multiple pieces of evidence. Likely only present at this location if `evidence` is a multi-value array.
expires	DateTime	If the achievement has some notion of expiry, this indicates a timestamp when a badge should no longer be considered valid. After this time, the badge should be considered expired.
revoked	Boolean	Defaults to `false` if Assertion is not referenced from a `revokedAssertions` list and may be omitted. See RevocationList. If `revoked` is true, only `revoked` and `id` are required properties, and many issuers strip a hosted Assertion down to only those properties when revoked.
revocationReason	Text	Optional published reason for revocation, if revoked.

Deprecated properties still in use by some implementations:

- uid – String – Unique Identifier for the badge. This is expected to be *locally* unique on a per-issuer basis and for hosted badges on a per-origin basis. It may not be necessarily globally unique. `uid` has been replaced by the IRI-based `id` property. It should not be used in v2.0+ Assertions.

Nota. Se visualiza un ejemplo de Open Badges, por (IMS, 2018)

ERC 721

El Ethereum request for comment 721 describe una forma de implementar tokens no fungibles en un entorno de Ethereum basado en contratos inteligentes, los cuales tienen como objetivo la representación de activos únicos, los cuales son identificados por un tokenId, teniendo la facultad de dar al propietario de la cuenta que recibe el token, una propiedad de éste, así como el control de la transferencia del mismo hacia otras cuentas. Estas características son bastante útiles al momento de representar certificados educativos, pues agregar la característica de pertinencia al sistema, representado por la variable owner y la función ownerOf dentro del estándar, que permiten verificar quién es el poseedor de dicho certificado.

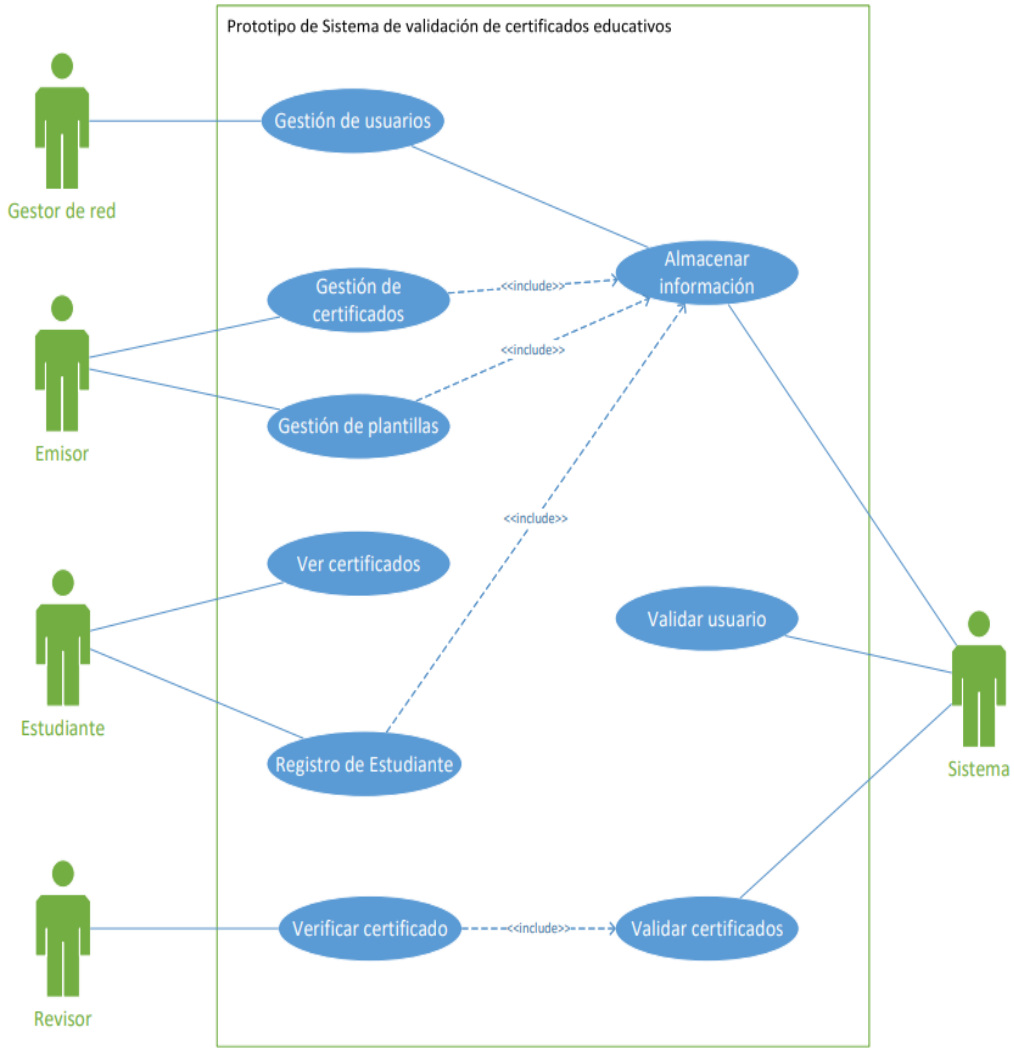
Diagramas UML

Diagrama de casos de uso

Una vez conocidos los requerimientos del sistema, se abstrae de manera general cuáles son las interacciones con el mismo, teniendo así las interacciones con el sistema descritas en la **Figura 10.**

Figura 10.

Casos de uso general



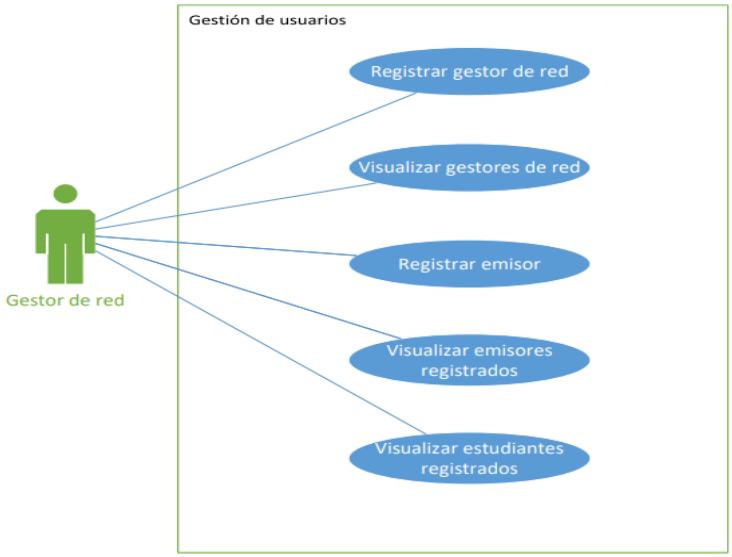
- Caso de uso Gestión de usuarios

Este caso de uso, como se puede apreciar en la **Figura 11** comprende las actividades del gestor de red relacionadas al registro y visualización de otros usuarios gestores de red, para el registro de estos se consideró la identificación del usuario como la dirección de una cuenta de Ethereum, descrita anteriormente, siendo este el único parámetro necesario para registrar un gestor de red. Para el caso del registro de emisor sus datos de entrada se encuentran descritos en la Tabla 6.

Después de cada registro se actualizarán automáticamente los valores más recientes en las tablas de visualización.

Figura 11.

Gestión de usuarios



- Caso de uso Emisor

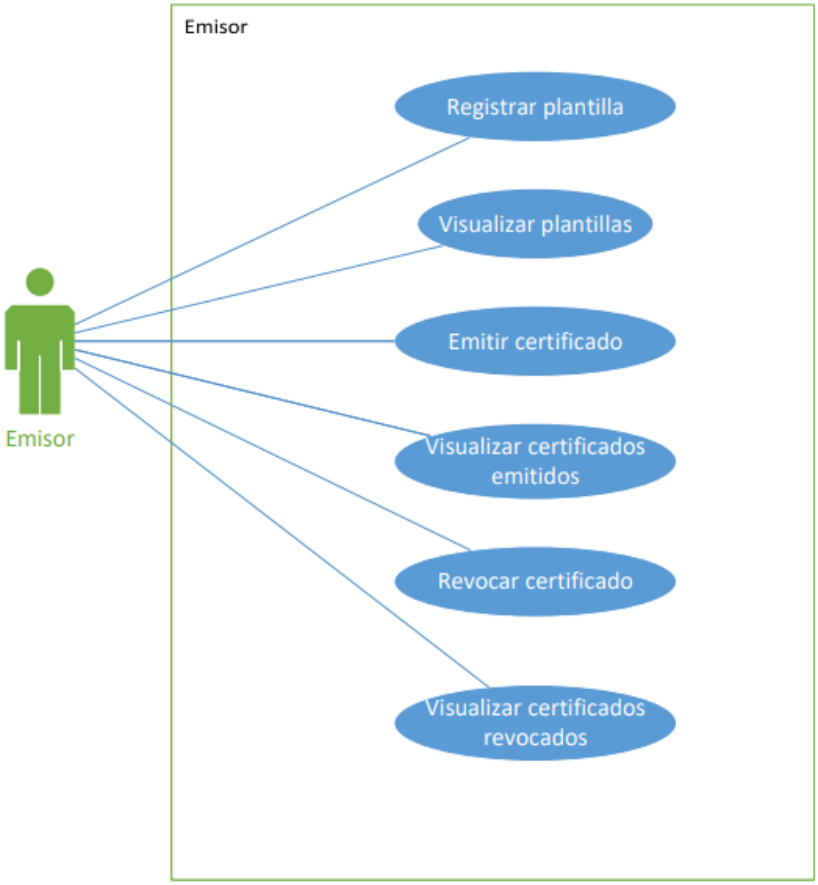
Describe las interacciones del usuario emisor con el sistema como se puede ver en la **Figura 12**. El registro de plantillas no requiere de interacciones previas, se debe

especificar los tipos de dato para cada uno de los parámetros de la plantilla, que serán completados al momento de emitir un certificado.

La emisión de certificados debe realizarse una vez se tengan registradas una o más plantillas del usuario emisor dentro del sistema. Se valida la información y los tipos de datos descritos en la plantilla, así como la carga de un archivo de evidencia. Este proceso genera una salida hacia el usuario Estudiante cuya dirección de Ethereum coincida con la entrada del formulario de creación del certificado, además de la generación CIDs correspondientes a un archivo Json y al archivo de evidencia, disponibles a través de IPFS.

Figura 12.

Emisor



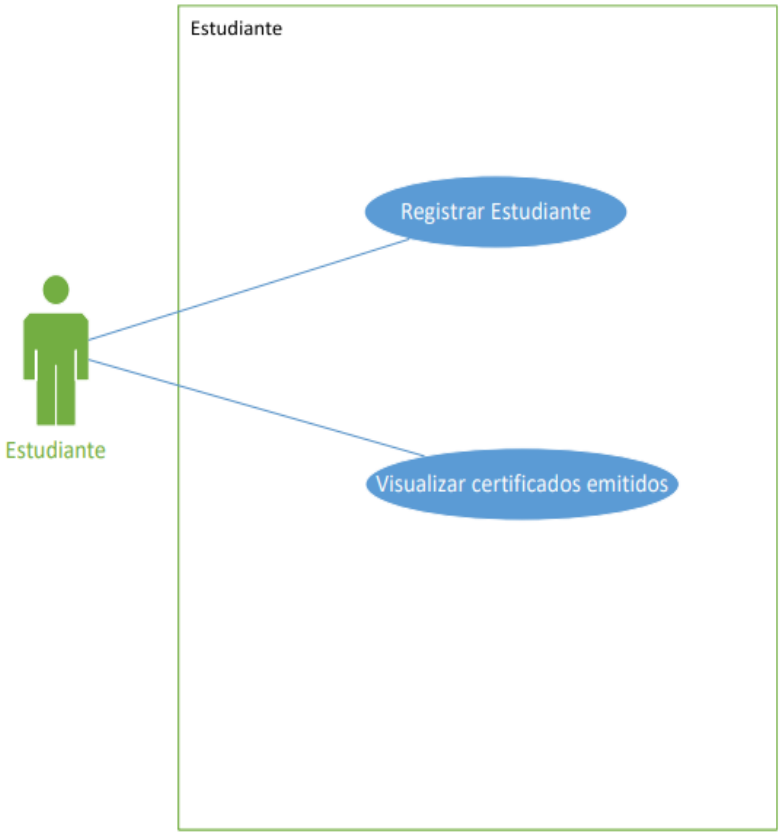
- Caso de uso Estudiante

Describe las interacciones del usuario Estudiante con el sistema, en dónde se puede apreciar que éste es responsable por su registro como en la **Figura 13**, para acceder a esta funcionalidad, debe encontrarse conectado a una cuenta de Ethereum a través del navegador, la cual se convertirá en su identificador de entrada hacia el sistema, así como para que el emisor pueda emitir registros hacia esta dirección.

Una vez el usuario se encuentre conectado a la cuenta de Ethereum utilizada en el registro, podrá acceder a la visualización de certificados emitidos a éste.

Figura 13.

Caso de uso Estudiante



- Caso de uso verificar certificados

El usuario revisor, no requiere de registro para realizar sus interacciones con el sistema, para lo cual se le facilitó una interfaz gráfica en la que pueda especificar los datos de entrada, en este caso pueden ser un CID correspondiente al archivo disponible a través de IPFS que fue generado al momento de emitir el certificado y debe ser facilitado externamente por el estudiante. También existe la opción de subir el archivo para que el sistema realice la validación del mismo. Una vez especificada la variable de entrada el sistema procederá a validar la información y presentarla en pantalla al revisor. Esto se aprecia en la **Figura 14.**

Figura 14.

Verificar certificados

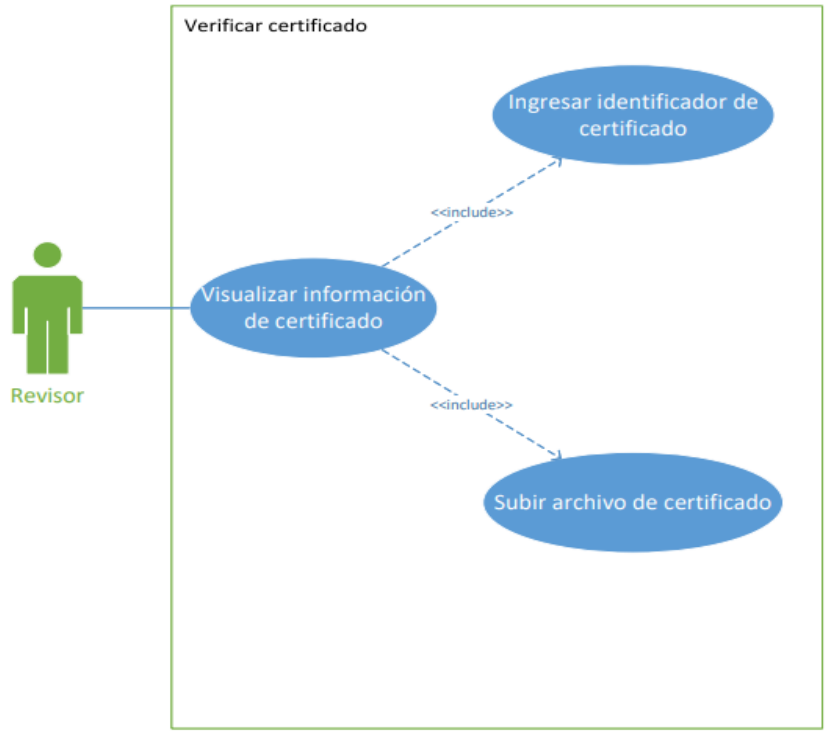
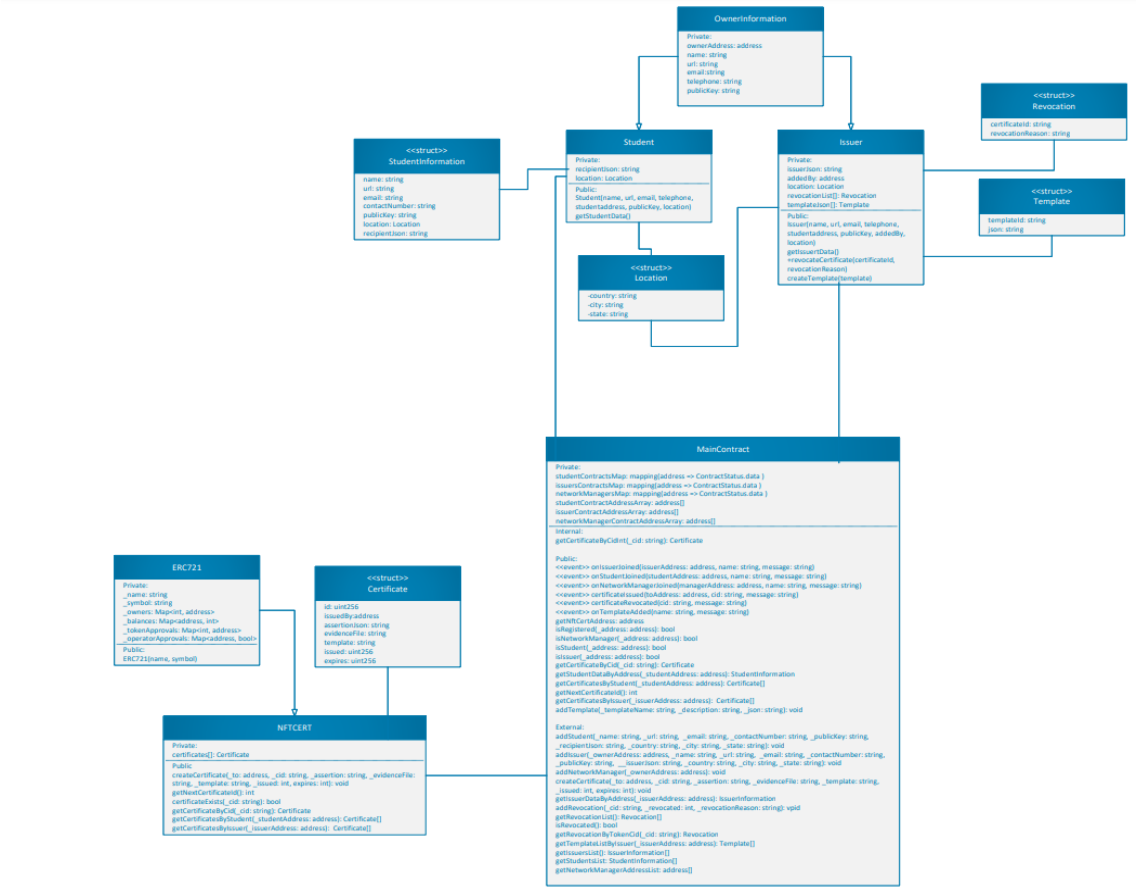


Diagrama de clases

Los contratos inteligentes se han seleccionado previamente como recurso para el almacenamiento, validación y procesamiento de la información del sistema, y siendo estos escritos en Solidity, un lenguaje de programación orientado a objetos, es posible realizar un diagrama de clases, en el que cada clase representa a un contrato inteligente y sus estructuras de objetos. Se realiza teniendo en cuenta los 4 niveles de privacidad propios de Solidity, así como propiedades como la herencia. Este diagrama se describe en la **Figura 15**.

Figura 15.

Diagrama de clases



- Contrato principal Main

Este contrato como se ve en la **Figura 16,** se encarga de interactuar con el exterior mediante los atributos y métodos externos y públicos descrito en la clase MainContract, que contiene las direcciones de los contratos de usuarios y certificados desplegados de manera que es a través de ésta que los usuarios interactúan con el resto de Smart contracts en el sistema.

Figura 16.

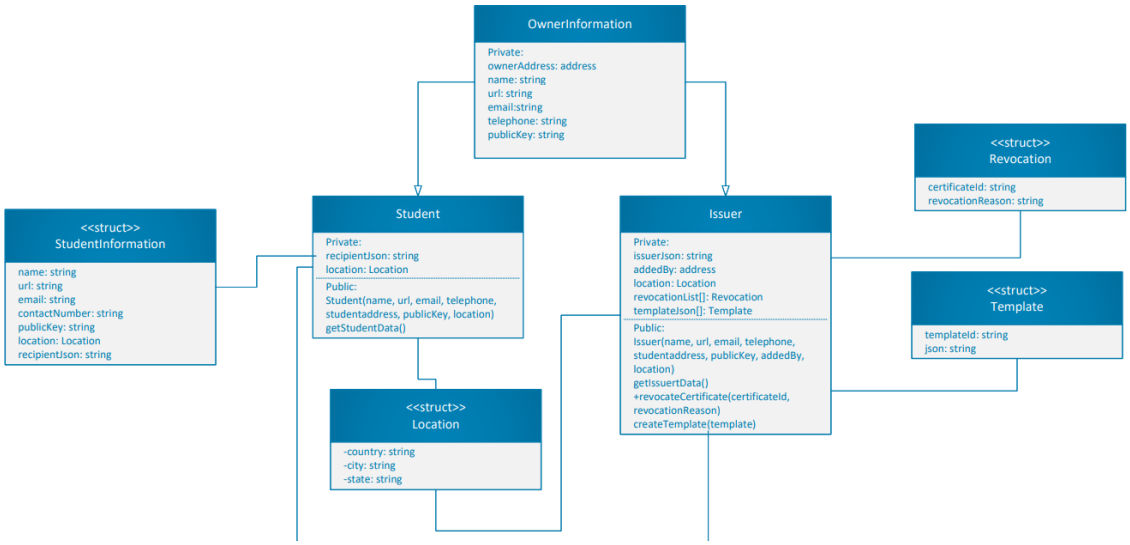
Contrato principal Main

- Contratos de usuarios

Los contratos descritos en las clases de la **Figura 17**, tienen como finalidad funcional desplegarse para corresponder a cada uno de los usuarios del sistema, manteniendo la información de estos en los atributos descritos para lo cual se establecen structs, que forman la definición de la información al almacenar en Blockchain.

Figura 17.

Contratos de usuarios



- Contratos de certificado y NFT

Los contratos descritos en la **Figura 18**, establecen la forma en la que se almacena la información de certificados, así como los estándares compatibles con NFT que aseguran las características de unicidad de cada certificado registrado.

Figura 18.

Contratos de certificados y NFT

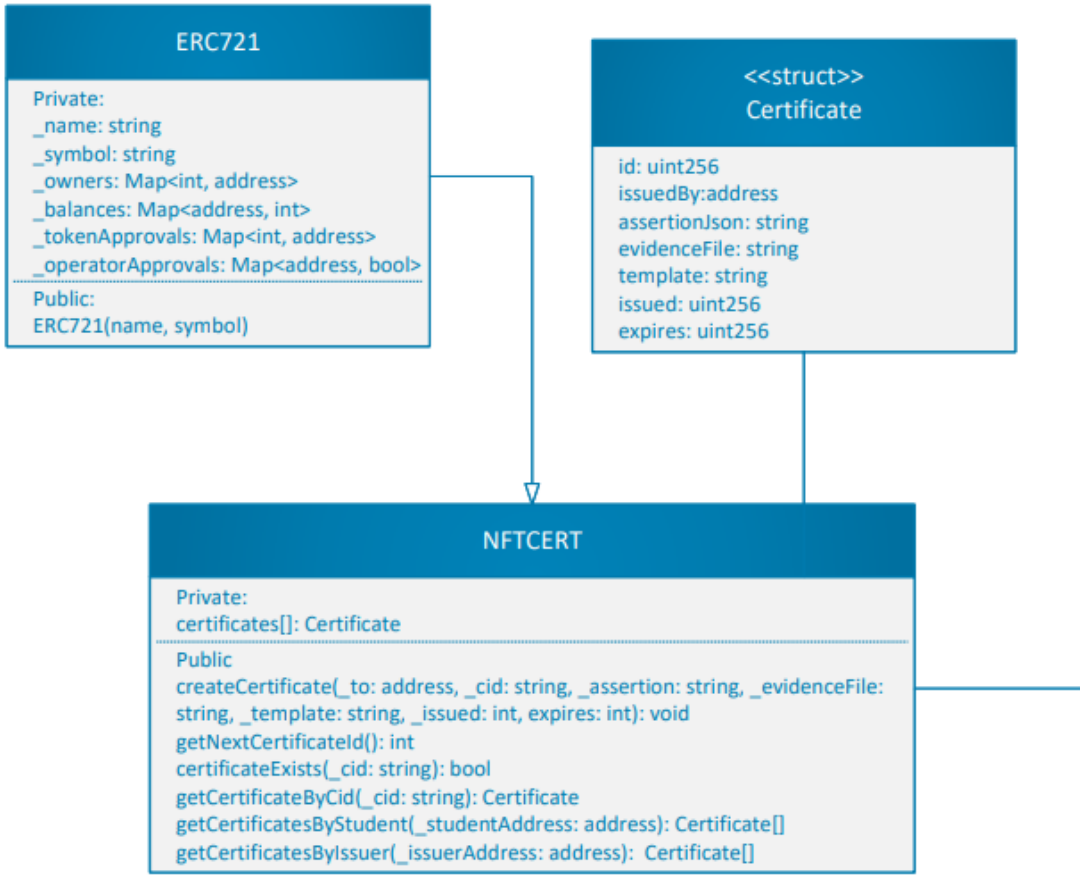
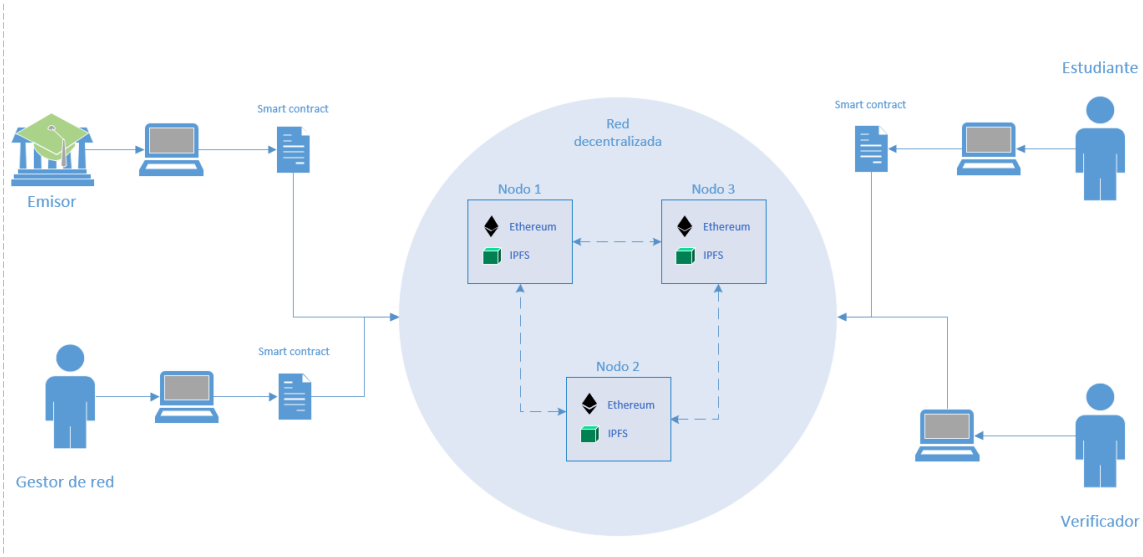


Diagrama de arquitectura

Se plantea la arquitectura de los componentes de la solución a nivel general como se puede ver en la **Figura 19.**

Figura 19.

Diagrama de arquitectura



Desarrollo del sistema de validación de certificados educativos

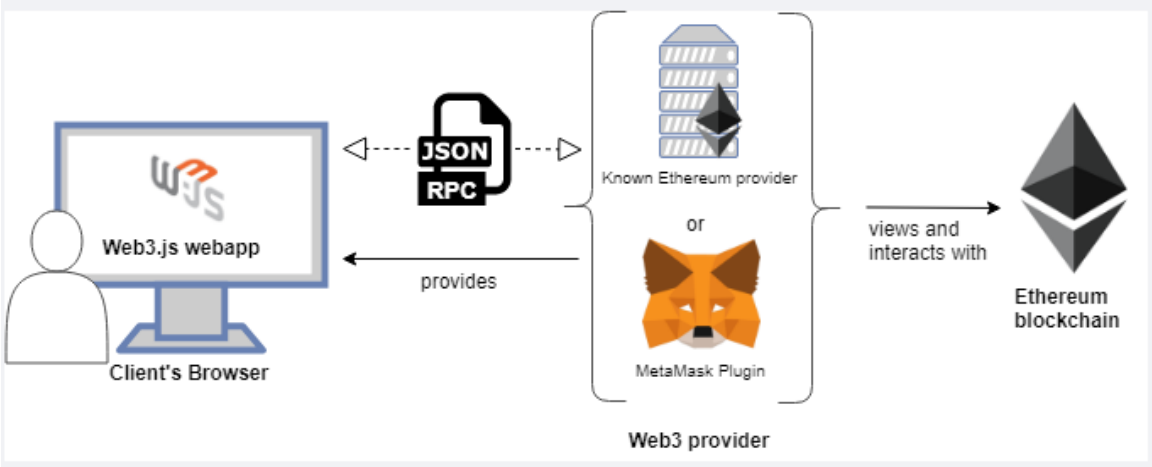
Librerías, herramientas y Frameworks

Metamask. Se utiliza la billetera electrónica Metamask como medio para interactuar con el sistema a través de cuentas de Ethereum, asegurando de esta manera la compatibilidad del prototipo con una de las billeteras electrónicas más usadas.

Web 3. Consiste en una librería compatible con Javascript que ayuda a interactuar con los nodos o proveedores de Ethereum a través de protocolo Http, IPC o WebSockets, la necesidad de usar esta librería se debe a que facilita la comunicación entre el cliente y el proveedor que se realiza a través de llamadas a procedimientos remotos en formato JSON-RPC, esta interacción se aprecia en la **Figura 20** (web3 js, 2021).

Figura 20.

Interacción de web 3 con el cliente



Nota. Se visualiza la interacción de web 3 con el cliente, por (Lasa, 2019).

Solidity. Es un lenguaje desarrollado para correr sobre la Ethereum Virtual Machine (EVM), es orientado a objetos y se usa para describir contratos inteligentes (Solidity, 2021).

Truffle. Se trata de un framework que asiste en la compilación, despliegue y pruebas de contratos inteligentes escritos en lenguaje Solidity, contiene además herramientas de interacción con las redes Blockchain, teniendo la capacidad de interactuar con contratos desplegados desde la terminal con lenguaje Javascript, gracias a que implementa el módulo de web3 (TruffleSuite, 2021b).

Ganache. Es parte de la suite de Truffle y consiste en una red Ethereum de prueba que funciona de manera local, es configurable y permite realizar pruebas controladas sin la preocupación por asumir excesivos costos en las transacciones (TruffleSuite, 2021)

Contratos Open Zeppelin. Se trata de una librería enfocada en proveer las implementaciones de las especificaciones ERC de manera segura. De ésta se puede tomar como base implementaciones de contratos inteligentes como el ERC 721 que pueden ser incluidas libremente en el sistema (OpenZeppelin, 2021).

React. Se usa la librería React para la construcción de interfaces gráficas de usuario, que debido a que es compatible con el entorno Javascript de la solución y permite un desarrollo ordenado, basado en componentes que al ser reutilizables ayudan a acortar los tiempos de desarrollo, y mejoran el rendimiento de la aplicación (React, 2021).

Typescrypt. Para ayudar en el manejo de datos que deben ser transferidos desde los contratos inteligentes, cuyas especificaciones y tipos se encuentran descritas en los mismos, se utiliza Typescrypt con la finalidad de tener un mejor entendimiento de los datos procesados por la aplicación cliente. De esta manera el código Javascript tiene la característica de ser tipado especificando tipos de datos previamente descritos en interfaces (Typescriptlang.org, 2021).

Desarrollo de los contratos inteligentes

Los contratos inteligentes describen la funcionalidad de la lógica del sistema, así como el almacenamiento de datos de los usuarios, perfiles.

Ownable.sol. Este contrato inteligente establece un propietario representado por una dirección de Ethereum hacia el propio contrato, es una clase heredable que se instancia para que otras entidades hijas puedan acceder a esta propiedad. El método isOwner realiza la validación con respecto a la dirección que intenta interactuar con el contrato, de esta manera permitiendo su uso o negándolo a través de un require como se puede ver en la **Figura 21.**

Figura 21.

Ownable.sol

```
pragma solidity ^0.8.0;

contract Ownable {
    address internal ownerAddress;

    constructor(address _ownerAddress){
        ownerAddress = _ownerAddress;
    }

    modifier isOwner(){
        require(ownerAddress == msg.sender, 'you are not the owner');
        _;
    }
}
```

OwnerInformation.sol. Este contrato describe las propiedades básicas de un usuario en el sistema, se definen los atributos en común que tendrán todos los usuarios, así como la estructura Location, que describe una zona geográfica estas propiedades se pueden apreciar en la **Figura 22.**

Figura 22.

OwnerInformation.sol

```
contract OwnerInformation is Ownable {

    struct Location {
        string country;
        string city;
        string state;
    }

    string internal name;
    string internal url;
    string internal email;
    string internal contactNumber;
    string internal publicKey;
    Location internal location;
```

Student.sol. El contrato describe a un Estudiante dentro del sistema, la estructura de la información está dada por StudentInformation, que define la variable de retorno de datos para las consultas de estudiantes como se puede ver en la **Figura 23.**

Figura 23.

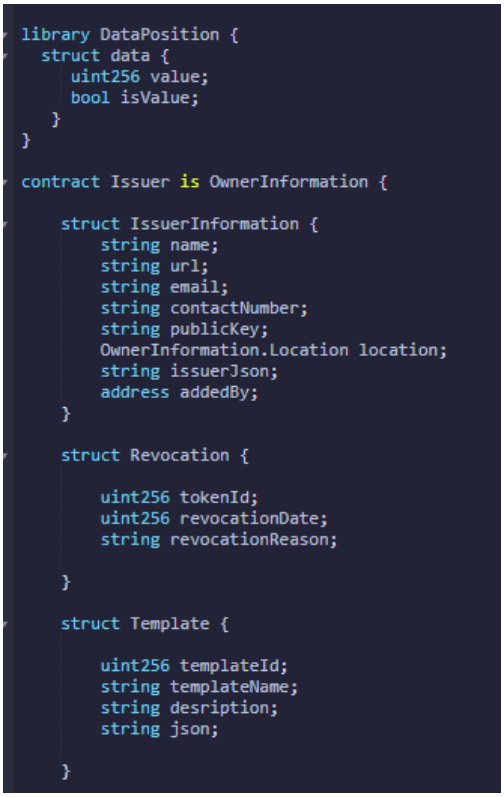
Student.sol

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

import "./OwnerInformation.sol";

contract Student is OwnerInformation {

    struct StudentInformation {
        string name;
        string url;
        string email;
        string contactNumber;
        string publicKey;
        OwnerInformation.Location location;
        string recipientJson;
    }

    string private recipientJson;

    /**
     * @dev Initialize the Student contract
     *
     * @param _ownerAddress representing the address of the student
     * @param _name representing the name of the student
     * @param _url representing a url which links to the student data
     * @param _email representing the email of the student
     * @param _contactNumber representing the telephone number of the student
     * @param _publicKey representing the student`s public key
     * @param _recipientJson representing the json with the student data as a Open Badges recipient
     * @param _country representing the country of the student
     * @param _city representing the city of the student
     * @param _state representing the specific state in the country of the student
     */
    constructor (address _ownerAddress, string memory _name, string memory _url, string memory _email,
    string memory _contactNumber, string memory _publicKey, string memory _recipientJson, string memory _country,
    string memory _city, string memory _state)
    OwnerInformation(_ownerAddress, _name, _url, _email, _contactNumber, _publicKey, _country, _city, _state){
        recipientJson = _recipientJson;
    }

    /**
    * @dev search for the student data
    *
    * @return StudentInformation the data of the student described in StudentInformation struct
    */
    function getStudentData() public view
    returns (StudentInformation memory){
        StudentInformation memory studentInformation = StudentInformation(name, url, email, contactNumber, publicKey, location, recipientJson);
        return studentInformation;
    }
}
```

Issuer.sol. Este contrato describe a los emisores dentro del sistema y la información propia de estos, como es el caso de las plantillas, descritas en la estructura Template, revocaciones de certificados, descrita en Revocation, y la propia variable de retorno de información de emisor IssuerInformatio como se puede ver en la **Figura 24.**

Figura 24.

Issuer.sol estructuras



```
library DataPosition {
  struct data {
     uint256 value;
     bool isValue;
   }
}

contract Issuer is OwnerInformation {

    struct IssuerInformation {
        string name;
        string url;
        string email;
        string contactNumber;
        string publicKey;
        OwnerInformation.Location location;
        string issuerJson;
        address addedBy;
    }

    struct Revocation {

        uint256 tokenId;
        uint256 revocationDate;
        string revocationReason;

    }

    struct Template {

        uint256 templateId;
        string templateName;
        string desription;
        string json;

    }
```

Los atributos y métodos que describe esta clase están orientados a la creación de plantillas, revocaciones y la propia data del emisor como se puede ver en la **Figura 25.**

Figura 25.

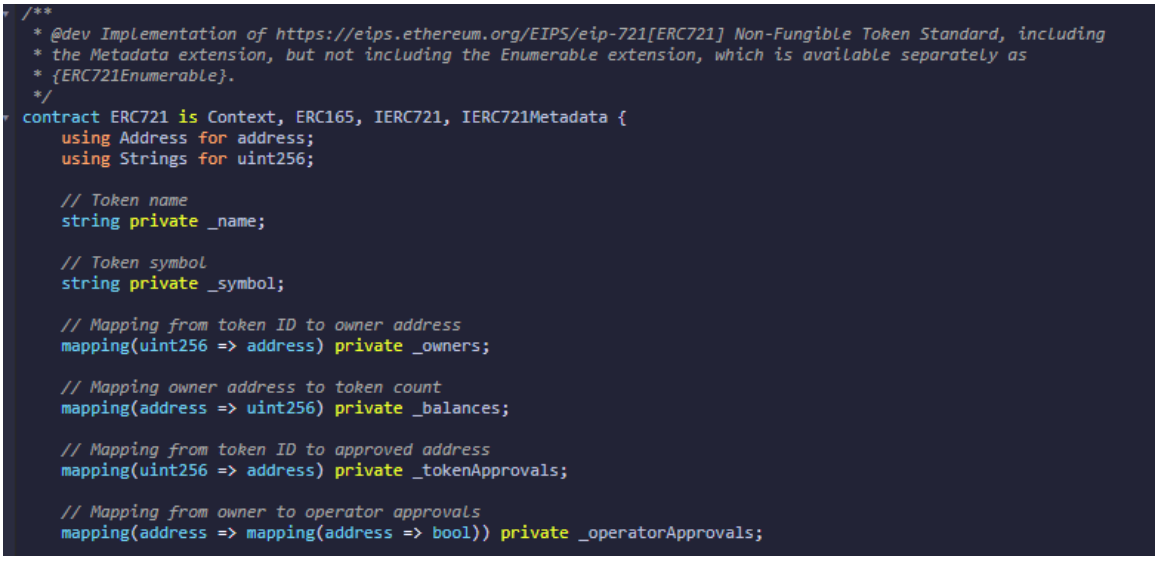
Issuer.sol atributos y métodos

```
string private issuerJson;

address private addedBy;

// Mapping from Token id to revocationList index
mapping(uint256 => DataPosition.data) private revocationMap;

Revocation[] private revocationList;

Template[] private templateList;

/*...*/
constructor (address _ownerAddress, string memory _name, string memory _url, string memory _email, string memory _contactNumber,
string memory _publicKey, string memory _issuerJson, string memory _country, string memory _city, string memory _state)
OwnerInformation(_ownerAddress, _name, _url, _email, _contactNumber, _publicKey, _country, _city, _state){...}

/*...*/
function getIssuerData() public view
returns (IssuerInformation memory){...}

/*...*/
function getRevocationList() public view
returns (Revocation[] memory){...}

/*...*/
function isRevocated(uint256 _tokenId) public view returns(bool) {...}

/*...*/
function getRevocationByTokenId(uint256 _tokenId) public view
returns (Revocation memory){...}

/*...*/
function addRevocation(uint256 _tokenId, uint256 _revocationDate, string memory _revocationReason) public{...}

/*...*/
function getTemplateList() public view returns (Template[] memory){...}

/*...*/
function addTemplate(string memory _templateName, string memory _description, string memory _json) public{...}
```

Erc721.sol. Este contrato describe la funcionalidad del Non fungible token, el cual entre sus características más importantes destacan; el identificador único, una dirección que representa al propietario del mismo, balances que describen cuantos tokens tiene cada dirección, el tokenApprovals que describe qué direcciones tienen permitida la gestión del token, operatorApprovals que describe las direcciones que tienen permisos para gestionar todos los tokens. Cada una de estas propiedades y permisos se encuentran definidos por la estructura mapping, como se puede observar en la **Figura 26.**

Figura 26.

ERC721.sol

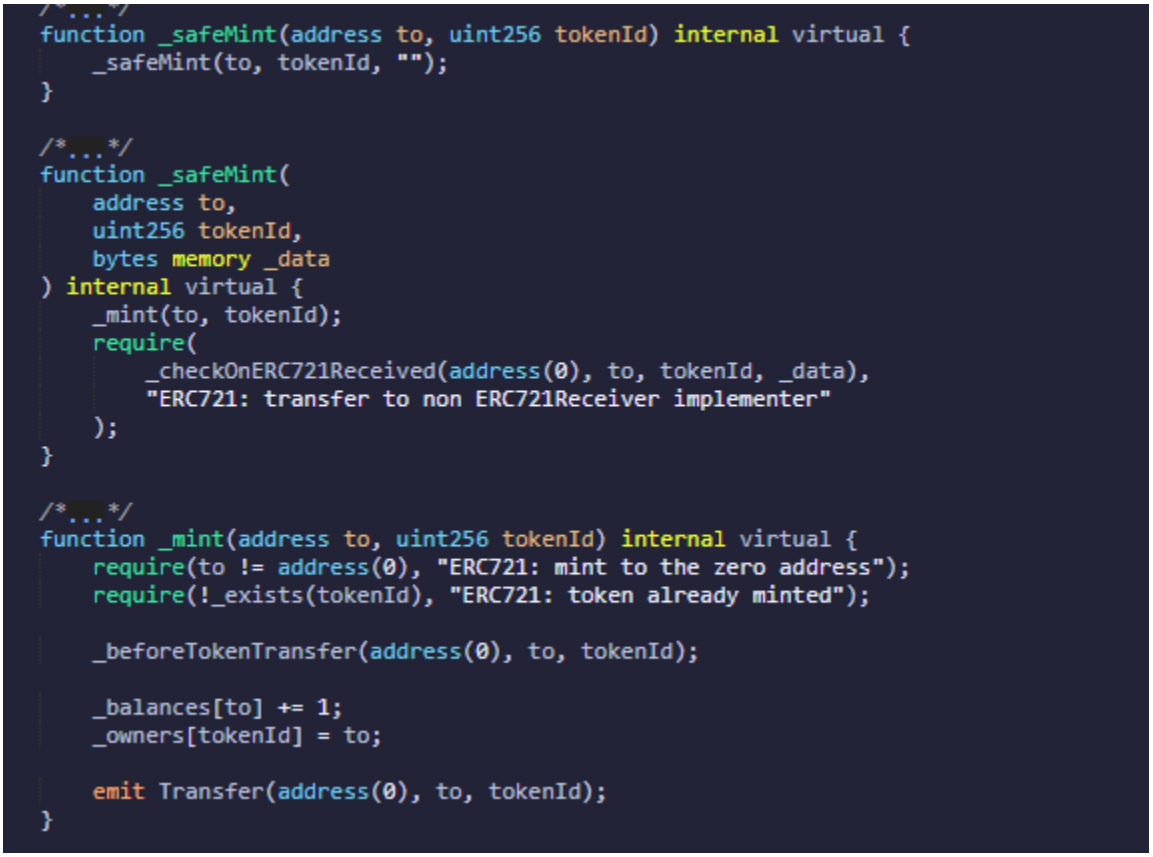


```
/**
 * @dev Implementation of https://eips.ethereum.org/EIPS/eip-721[ERC721] Non-Fungible Token Standard, including
 * the Metadata extension, but not including the Enumerable extension, which is available separately as
 * {ERC721Enumerable}.
 */
contract ERC721 is Context, ERC165, IERC721, IERC721Metadata {
    using Address for address;
    using Strings for uint256;

    // Token name
    string private _name;

    // Token symbol
    string private _symbol;

    // Mapping from token ID to owner address
    mapping(uint256 => address) private _owners;

    // Mapping owner address to token count
    mapping(address => uint256) private _balances;

    // Mapping from token ID to approved address
    mapping(uint256 => address) private _tokenApprovals;

    // Mapping from owner to operator approvals
    mapping(address => mapping(address => bool)) private _operatorApprovals;
```

La creación de los tokens se realiza a través de las funciones mint, que se encargan de establecer el id del token relacionado con su dirección propietaria. Como se puede apreciar en la **Figura 27.**

Figura 27.

ERC721 mint

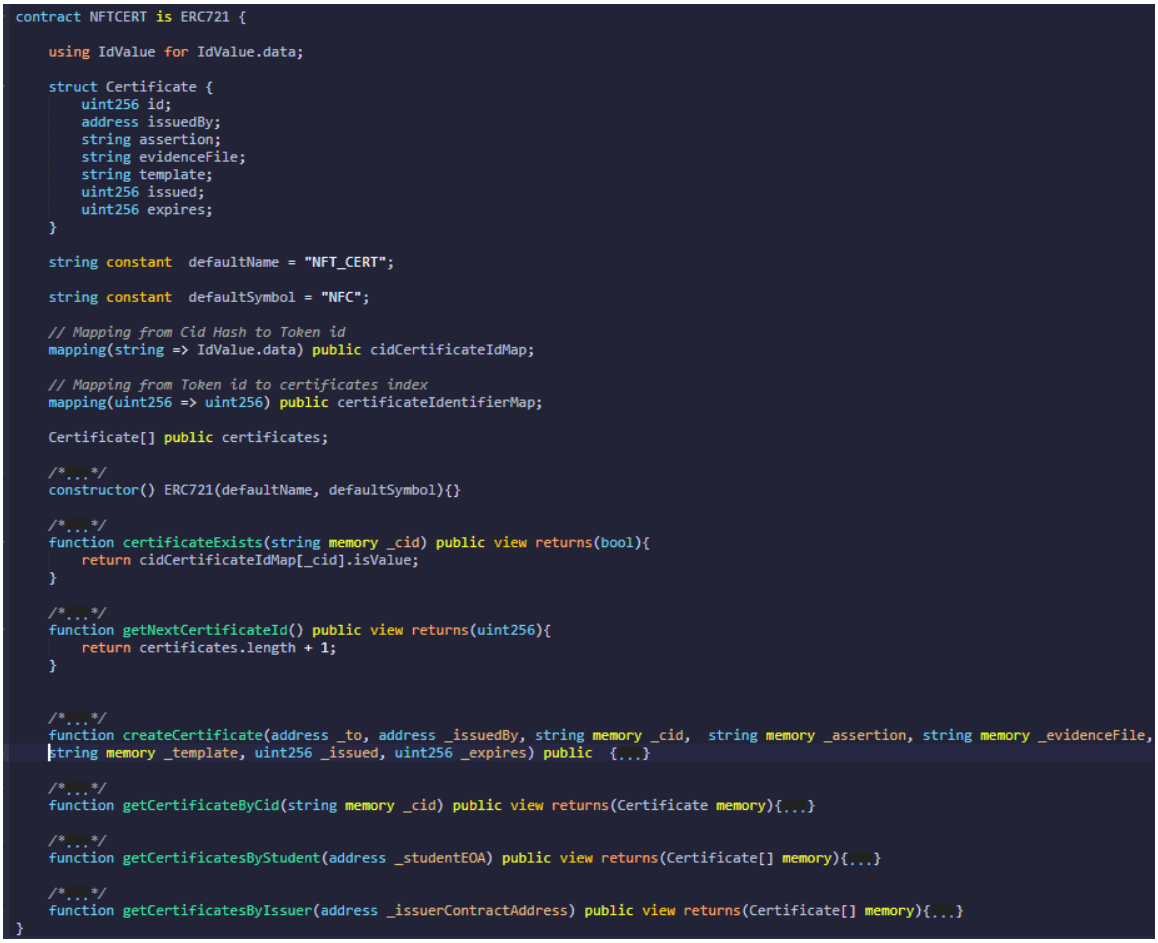


```
/*...*/
function _safeMint(address to, uint256 tokenId) internal virtual {
    _safeMint(to, tokenId, "");
}

/*...*/
function _safeMint(
    address to,
    uint256 tokenId,
    bytes memory _data
) internal virtual {
    _mint(to, tokenId);
    require(
        _checkOnERC721Received(address(0), to, tokenId, _data),
        "ERC721: transfer to non ERC721Receiver implementer"
    );
}

/*...*/
function _mint(address to, uint256 tokenId) internal virtual {
    require(to != address(0), "ERC721: mint to the zero address");
    require(!_exists(tokenId), "ERC721: token already minted");

    _beforeTokenTransfer(address(0), to, tokenId);

    _balances[to] += 1;
    _owners[tokenId] = to;

    emit Transfer(address(0), to, tokenId);
}
```

NFTCERT.sol. NFTCERT hereda las propiedades del contrato ERC 721 convirtiéndose así en un Non fungible token, pero se agrega la información relevante a un certificado educativo. A través de este contrato deberán realizarse las validaciones correspondientes para identificar la validez de un certificado como se puede ver en la **Figura 28.**

Figura 28.

NFTCERT.sol



```
contract NFTCERT is ERC721 {

    using IdValue for IdValue.data;

    struct Certificate {
        uint256 id;
        address issuedBy;
        string assertion;
        string evidenceFile;
        string template;
        uint256 issued;
        uint256 expires;
    }

    string constant  defaultName = "NFT_CERT";

    string constant  defaultSymbol = "NFC";

    // Mapping from Cid Hash to Token id
    mapping(string => IdValue.data) public cidCertificateIdMap;

    // Mapping from Token id to certificates index
    mapping(uint256 => uint256) public certificateIdentifierMap;

    Certificate[] public certificates;

    /*...*/
    constructor() ERC721(defaultName, defaultSymbol){}

    /*...*/
    function certificateExists(string memory _cid) public view returns(bool){
        return cidCertificateIdMap[_cid].isValue;
    }

    /*...*/
    function getNextCertificateId() public view returns(uint256){
        return certificates.length + 1;
    }


    /*...*/
    function createCertificate(address _to, address _issuedBy, string memory _cid,  string memory _assertion, string memory _evidenceFile,
    string memory _template, uint256 _issued, uint256 _expires) public  {...}

    /*...*/
    function getCertificateByCid(string memory _cid) public view returns(Certificate memory){...}

    /*...*/
    function getCertificatesByStudent(address _studentEOA) public view returns(Certificate[] memory){...}

    /*...*/
    function getCertificatesByIssuer(address _issuerContractAddress) public view returns(Certificate[] memory){...}
}
```

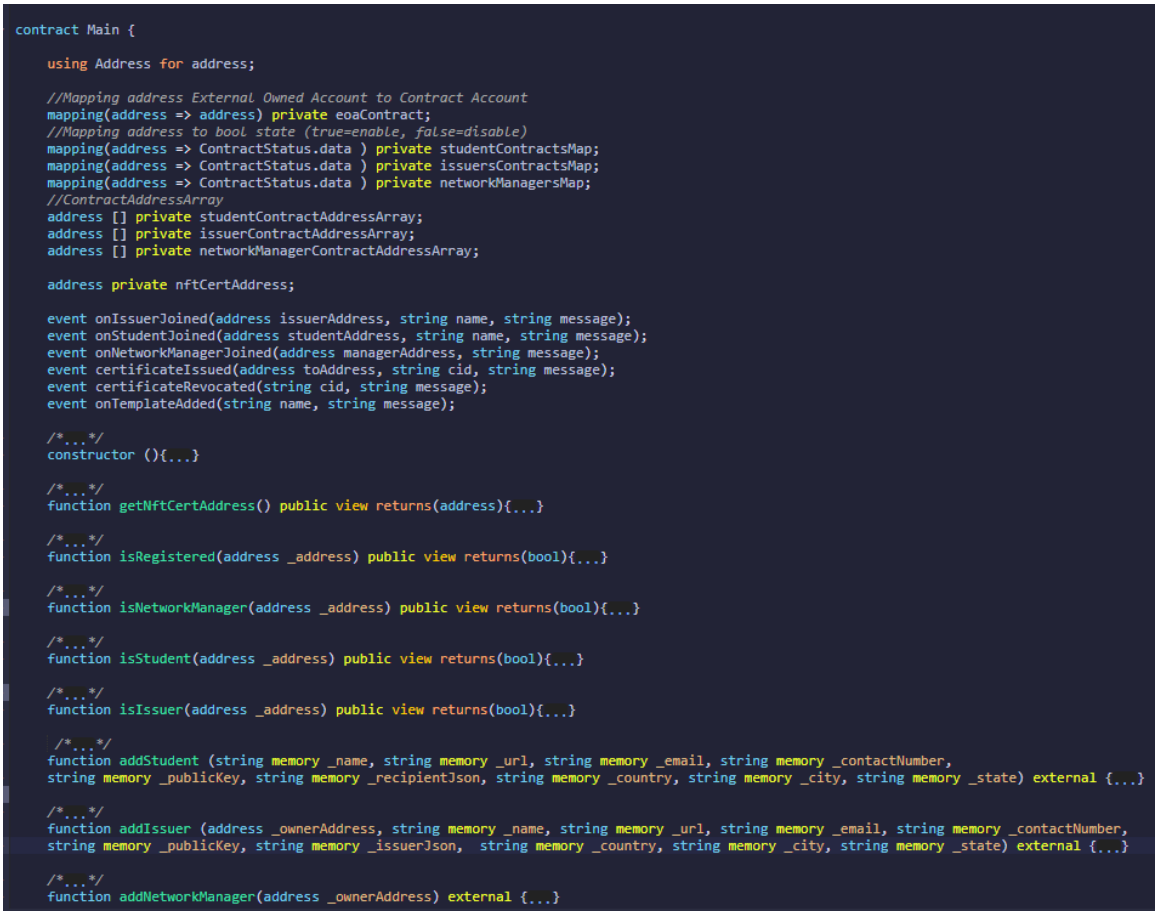
Main.sol. El contrato base se encarga de funcionar como una interfaz con la que interactuan los usuarios. Es en este contrato donde se instancia y se tiene registro de los demás contratos desplegados, de manera que solo a través de éste se pueda lograr interacción con estos. Además, define los perfiles de usuario para cada dirección de Ethereum mediante un mapeo de datos y así mismo la relaciona con sus respectivos contratos, como se puede ver en la **Figura 29.**

Para interactuar de manera externa con la aplicación cliente se especifican las funciones con la privacidad external, las cuales solo pueden ser ejecutadas

exclusivamente fuera del contrato, en donde tenemos la creación de usuarios, contratos, plantillas y revocaciones, así como funciones de validación y consulta de usuarios y certificados. Estas funciones, dependiendo de su grado de privacidad contienen validaciones que deben ser cumplidas para ejecutarse, caso contrario el contrato revertirá la transacción y el estado de la Blockchain a como era antes de la última llamada.

Figura 29.

Main.sol



```
contract Main {

    using Address for address;

    //Mapping address External Owned Account to Contract Account
    mapping(address => address) private eoaContract;
    //Mapping address to bool state (true=enable, false=disable)
    mapping(address => ContractStatus.data ) private studentContractsMap;
    mapping(address => ContractStatus.data ) private issuersContractsMap;
    mapping(address => ContractStatus.data ) private networkManagersMap;
    //ContractAddressArray
    address [] private studentContractAddressArray;
    address [] private issuerContractAddressArray;
    address [] private networkManagerContractAddressArray;

    address private nftCertAddress;

    event onIssuerJoined(address issuerAddress, string name, string message);
    event onStudentJoined(address studentAddress, string name, string message);
    event onNetworkManagerJoined(address managerAddress, string message);
    event certificateIssued(address toAddress, string cid, string message);
    event certificateRevocated(string cid, string message);
    event onTemplateAdded(string name, string message);

    /*...*/
    constructor (){...}

    /*...*/
    function getNftCertAddress() public view returns(address){...}

    /*...*/
    function isRegistered(address _address) public view returns(bool){...}

    /*...*/
    function isNetworkManager(address _address) public view returns(bool){...}

    /*...*/
    function isStudent(address _address) public view returns(bool){...}

    /*...*/
    function isIssuer(address _address) public view returns(bool){...}

     /*...*/
    function addStudent (string memory _name, string memory _url, string memory _email, string memory _contactNumber,
    string memory _publicKey, string memory _recipientJson, string memory _country, string memory _city, string memory _state) external {...}

    /*...*/
    function addIssuer (address _ownerAddress, string memory _name, string memory _url, string memory _email, string memory _contactNumber,
    string memory _publicKey, string memory _issuerJson,  string memory _country, string memory _city, string memory _state) external {...}

    /*...*/
    function addNetworkManager(address _ownerAddress) external {...}
```

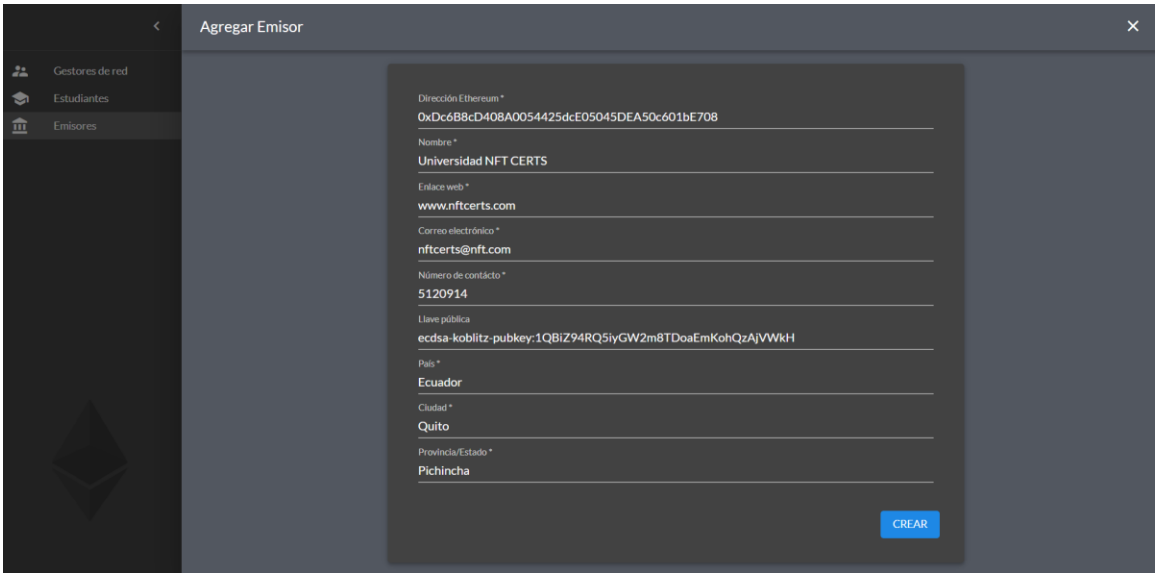
Desarrollo de la Dapp

Para agregar un emisor, la cuenta de Ethereum asociada al usuario debe corresponder a un gestor de red habilitado, se llena la información descrita en el la tabla 2, considerando que la dirección de Ethereum del emisor a registrar no haya sido registrada previamente.

En el ejemplo se puede observar como en la **Figura 30** que el gestor de red registra su propia dirección, esto es posible pues se definió que un gestor de red puede convertirse en emisor, representando de esta manera a una institución educativa en el mundo real que tiene el control sobre los registros de otros gestores y emisores a la red.

Figura 30.

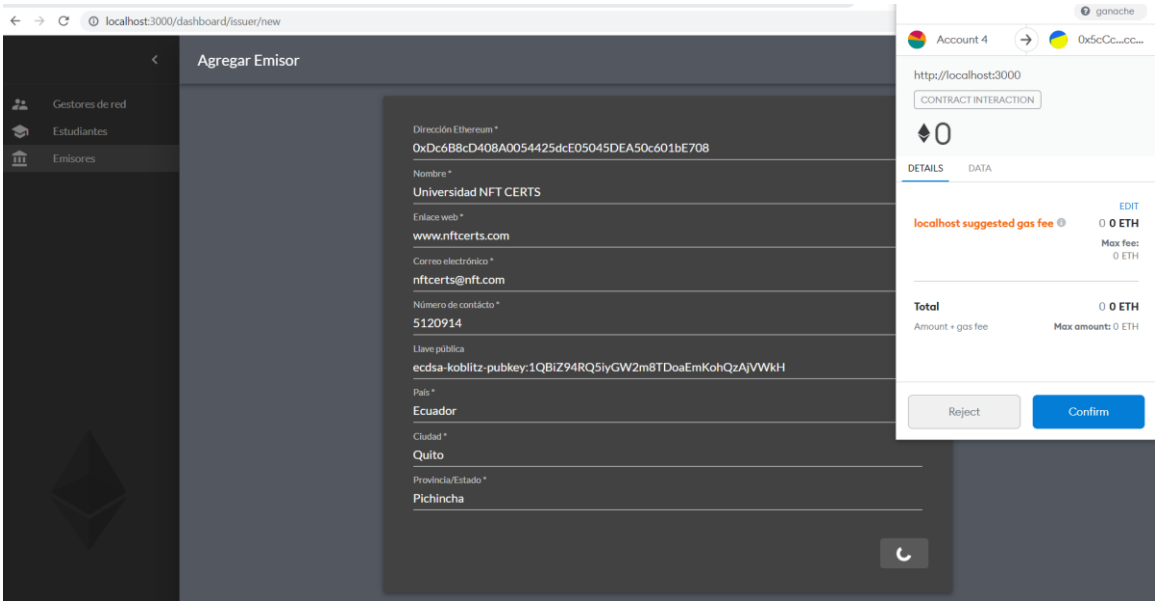
Desarrollo de la Dapp



Y se procede a firmar la transacción como en la **Figura 31** para dar paso a el procesamiento de datos en Blockchain. Mediante la aplicación Metamask, que actúa como el proveedor Web 3 para interactuar con el cliente y la red se realiza esta acción.

Figura 31.

Firma de la transacción



Como respuesta obtendremos la confirmación de usuario registrado como en la **Figura 32** y se puede acceder a la tabla de emisores para verificar el más reciente como en la **Figura 33.**

Figura 32.

Tabla de emisores

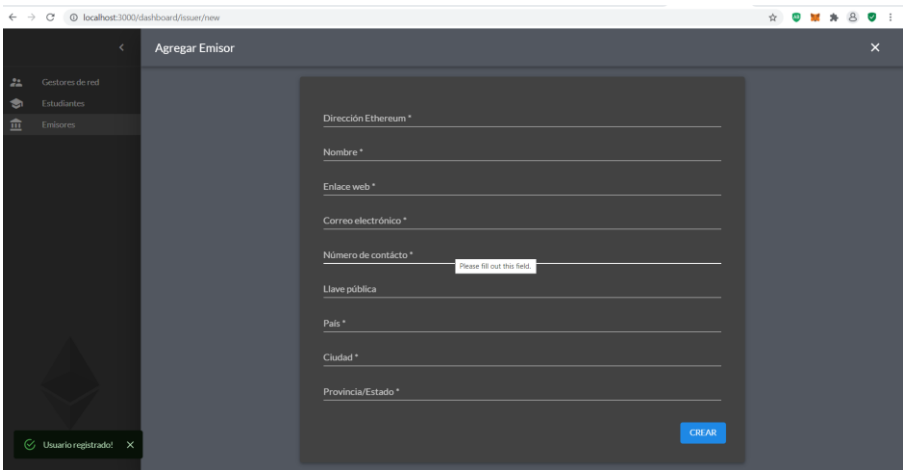
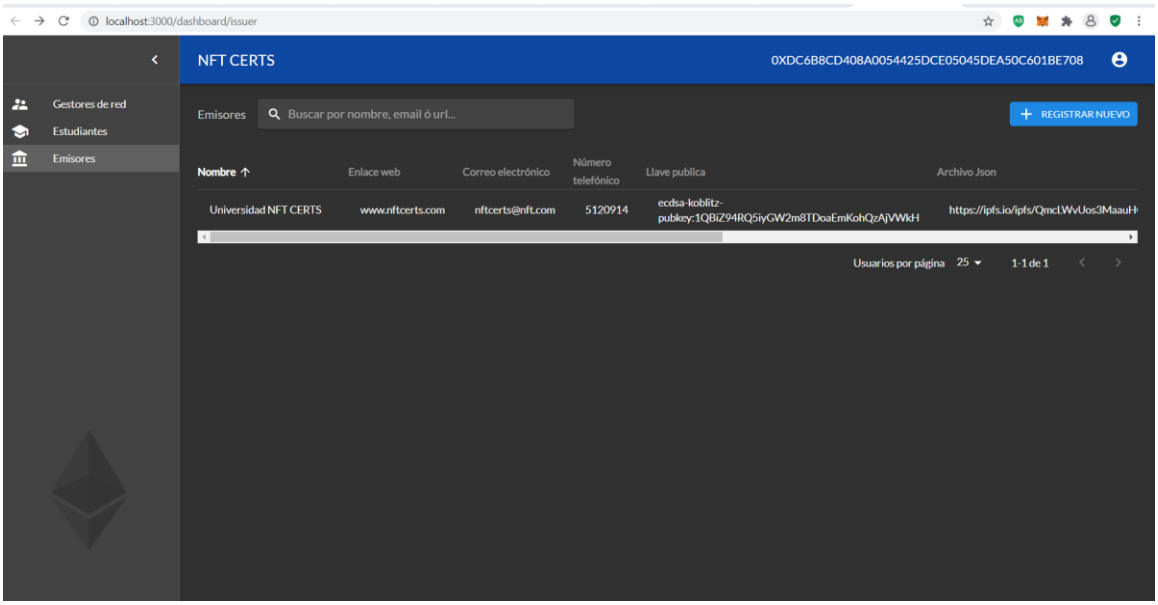


Figura 33.

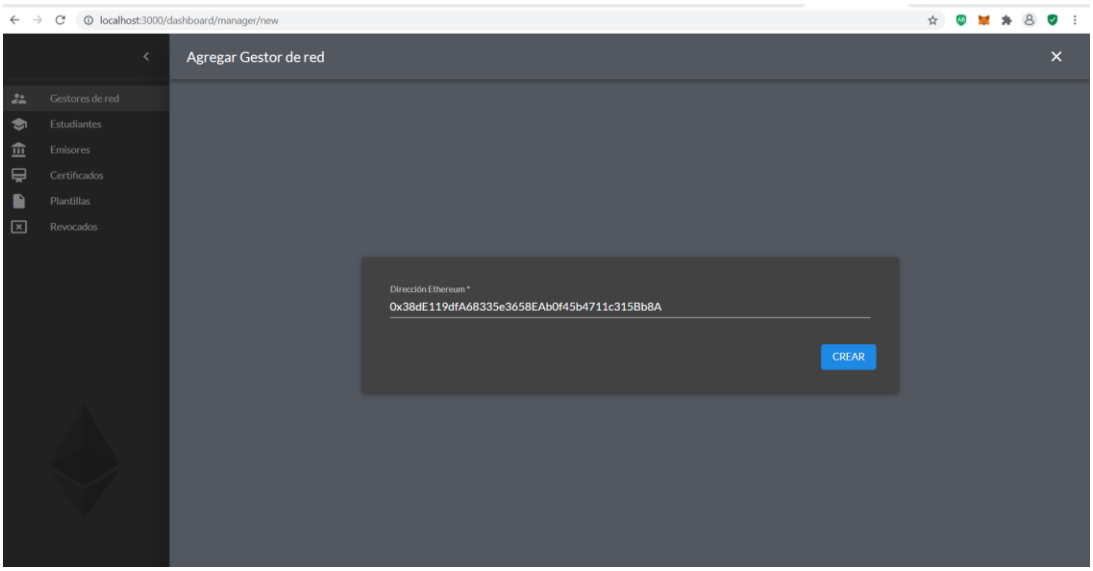
Resultado



Para agregar un gestor de red, la cuenta de Ethereum asociada al usuario debe corresponder a otro gestor de red habilitado, se especifica la dirección de Ethereum como en la **Figura 34** del nuevo gestor de red y se da clic en el botón crear. Se firma la transacción de forma similar como se vio en la **Figura 31** y se recibe la confirmación de manera similar a la **Figura 32.**

Figura 34.

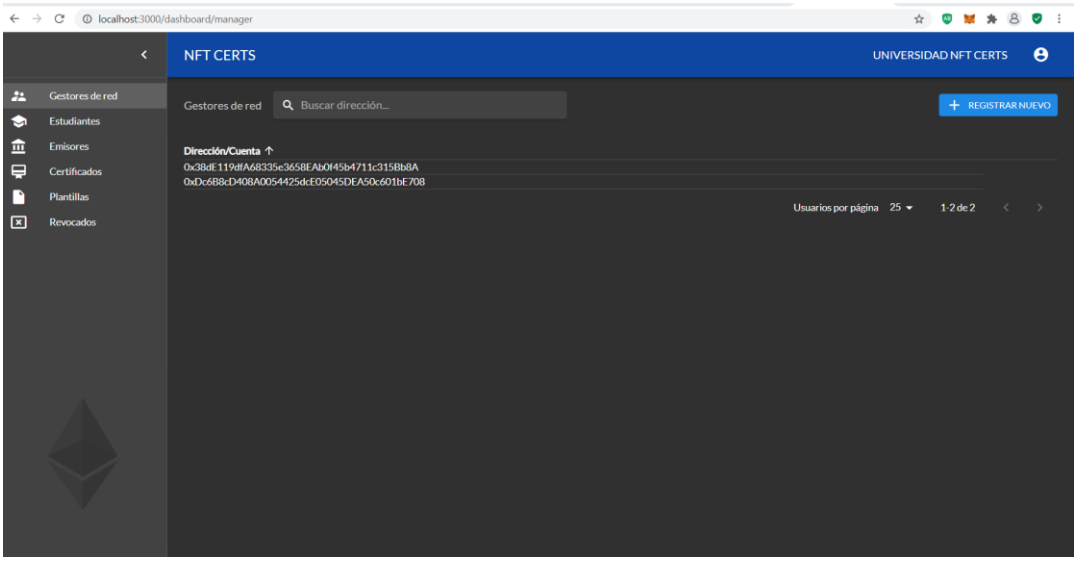
Gestor de red



Al cerrar el panel se podrá verificar en la tabla el gestor de red agregado recientemente como en la **Figura 35**.

Figura 35.

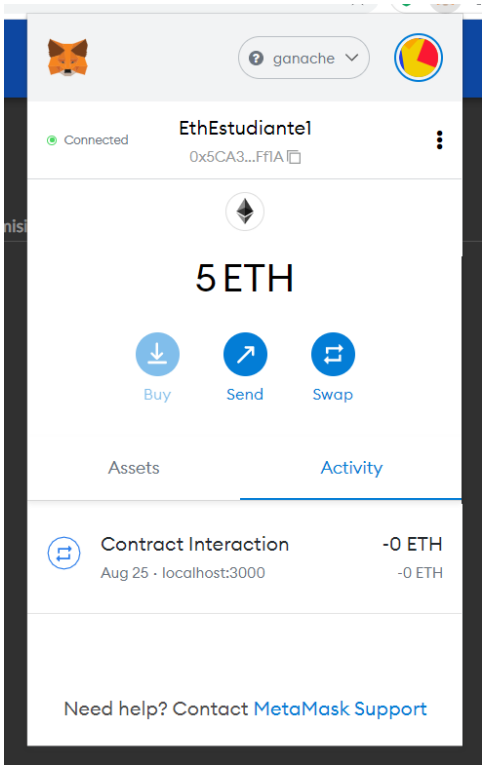
Gestor de red agregado



Para el ejemplo de registro de estudiante se utiliza la cuenta de número “0x5CA39f9903B3e34941f0dF42aEe4C323D062Ff1A” que se puede ver en la **Figura 36,** que ha sido previamente creada con la wallet de Ethereum, la información de esta cuenta no se debe especificar en el formulario de usuario, pero el software la detecta al tener como proveedor del servicio a Metamask y registra el usuario con esta cuenta.

Figura 36.

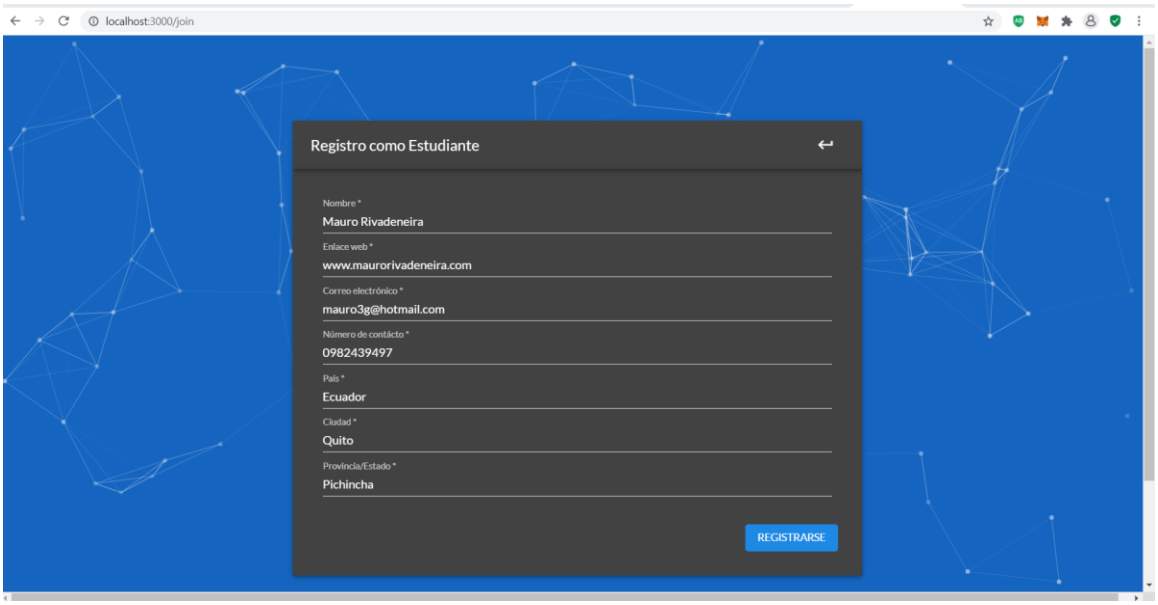
Billetera Metamask



Para el registro de estudiante, tener en cuenta que debe ser una cuenta que no haya sido registrada previamente como algún otro usuario del sistema, no se permite en este caso que un emisor pueda ser a su vez estudiante. Se da clic en registrarse y se obtiene como respuesta la confirmación de la creación del usuario. Esta funcionalidad se describe en la **Figura 37.**

Figura 37.

Registro como estudiante

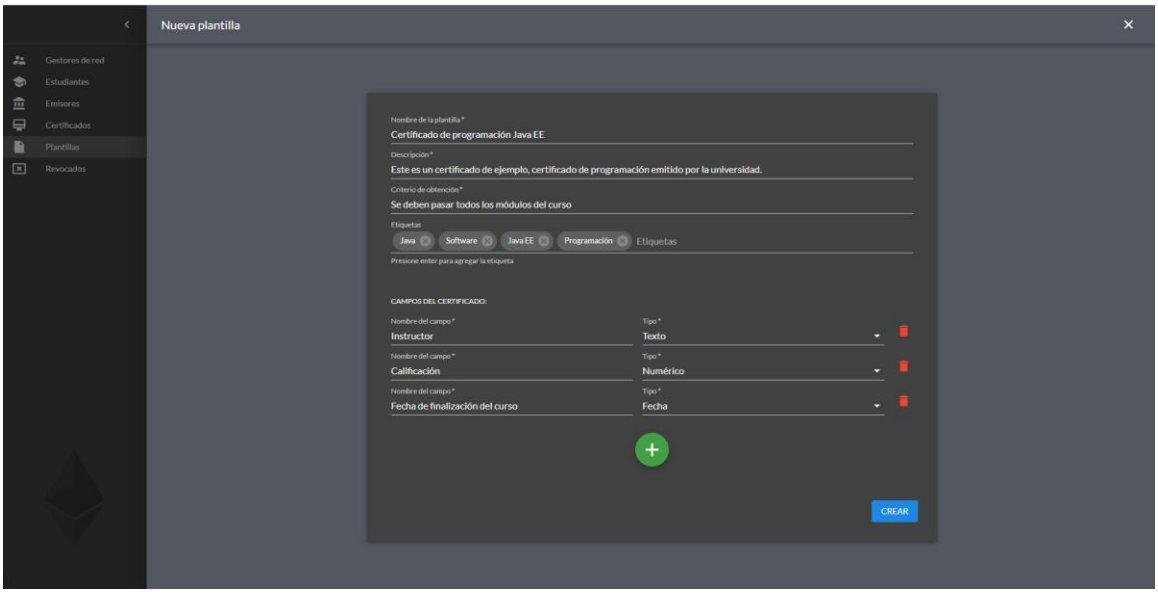


Para el registro de plantillas de certificados de la **Figura 38**, se debe tener conectada una cuenta de emisor, y dirigirse al apartado de Plantillas en el menú de opciones y registrar nuevo. Se deben llenar los datos correspondientes a la plantilla y además se permite agregar algunos campos adicionales como metadata, de manera que ésta se almacene dentro de la red IPFS al emitir el certificado, siguiendo con el estándar Open Badges.

De forma similar, se da clic en crear, se firma la transacción y se espera por la respuesta de creación como en la **Figura 39**.

Figura 38.

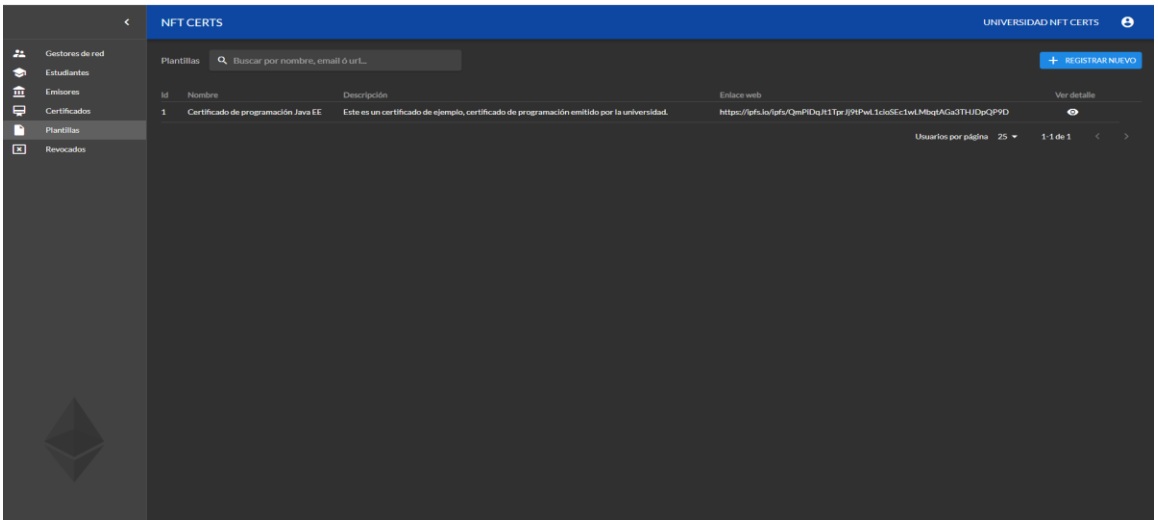
Registro de plantilla



Se puede constatar la creación de la plantilla en la respectiva tabla.

Figura 39.

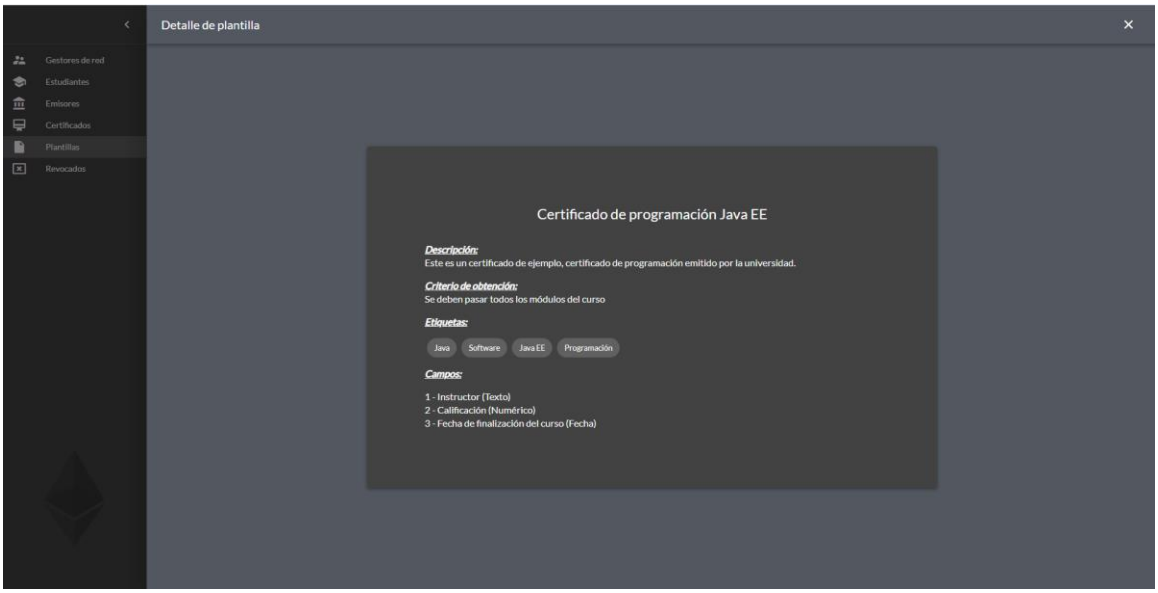
Creación de la plantilla



Para ver la metadata correspondiente se puede dar clic en el ícono de ver detalle, que despliega el panel de la **Figura 40.**

Figura 40.

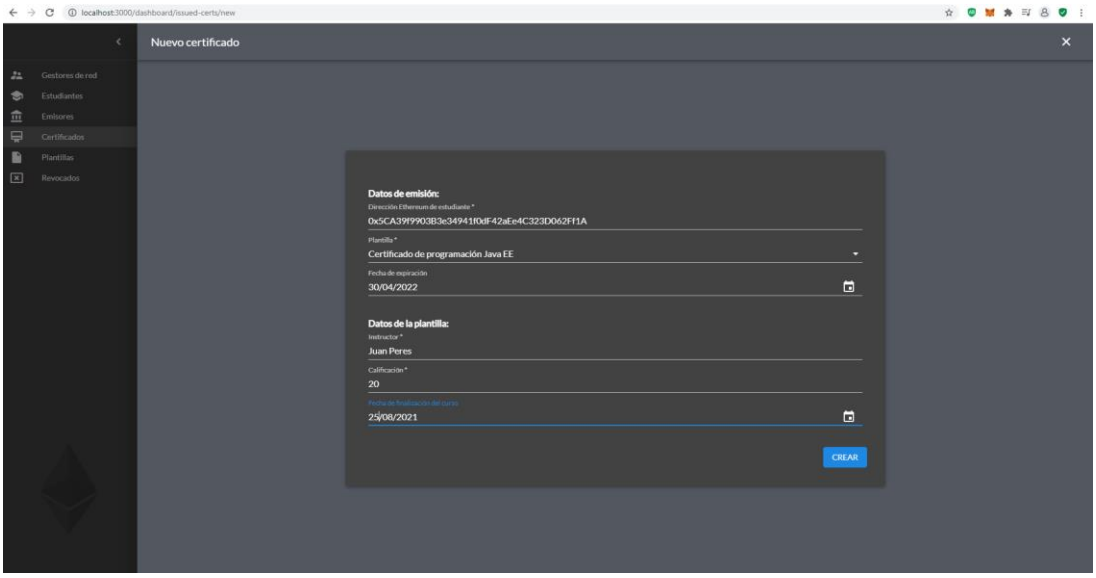
Metadata



Para el registro de plantillas de certificados, se debe tener conectada una cuenta de emisor y dirigirse al apartado de Certificados en el menú de opciones. A continuación se desplegará el formulario donde se debe especificar la dirección de Ethereum del estudiante como en la **Figura 41**, tener en cuenta que el estudiante debe estar previamente registrado, se deberá seleccionar una plantilla, e inmediatamente se muestra la metadata a completar correspondiente, se generará un archivo que contendrá el documento electrónico de evidencia del certificado. Se completa el proceso dando click en crear, se firma la transacción y se espera por la confirmación.

Figura 41.

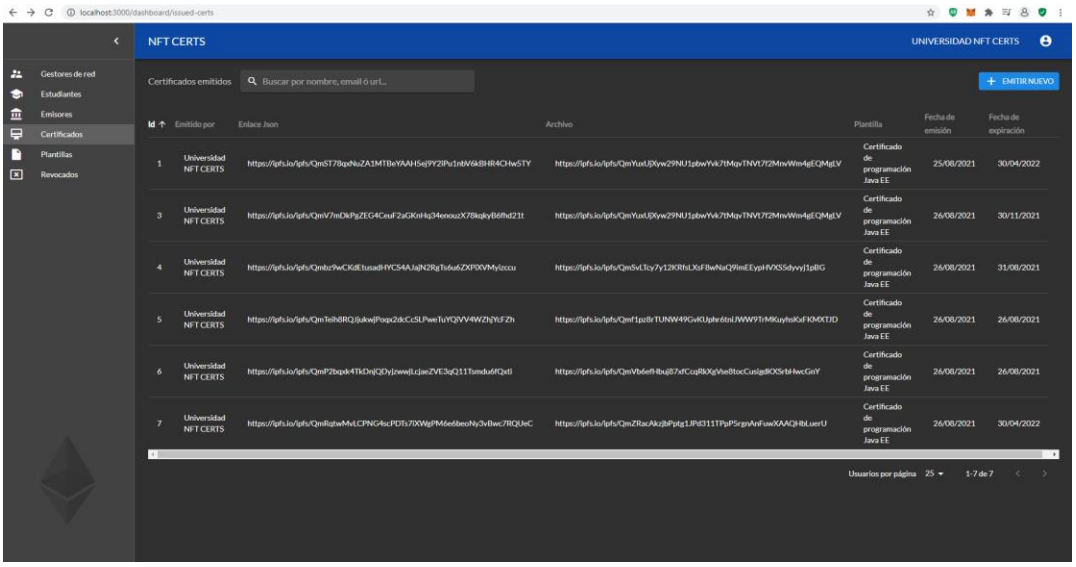
Nuevo certificado



Al regresar a la tabla se puede observar la información del certificado emitido recientemente como en la **Figura 42**.

Figura 42.

Certificado emitido



Si se sigue el link Enlace Json se puede encontrar disponible a través de IPFS la información del certificado de acuerdo al estándar Open Badges en el que se describe el mismo como una entidad de tipo “Assertion”, la cual contiene a las entidades “Recipient”, que describe al estudiante poseedor del certificado, “Badge”, que describe a la plantilla que contiene la metadata del certificado, así como el resto de campos especificados en el apartado de selección de Open Badges. La respuesta a esta consulta se visualiza en la **Figura 43.**

Y como se puede apreciar el archivo Json de la **Tabla 18,** contiene la siguiente información que corresponde a la entidad “Assertion”de Open Badges:

Tabla 18.

Resultado JSON Open Badges Assertion


```
{
 "@context": [
 "https://w3id.org/openbadges/v2",
 {
 "badgeFields": {
 "@id": "http://127.0.0.1/metadata",
 "@type": "https://json-schema.org/draft/2020-12/json-schema-core.html"
 },
 "evidenceFile": {
 "@id": "http://127.0.0.1/metadata",
 "@type": "URL"
 }
 }
 ],
 "type": "Assertion",
 "id": "http://127.0.0.1/verification/certificates/1",
 "recipient": {
 "@context": "https://w3id.org/openbadges/v2",
 "type": "Profile",
 "id": "http://127.0.0.1/verification/student/0x5CA39f9903B3e34941f0dF42aEe4C323D062Ff1A",
 "name": "Mauro Rivadeneira",
 "url": "www.maurorivadeneira.com",
 "email": "mauro3g@hotmail.com",
 "contactNumber": "0982439497",
 "publicKey": "",
 "studentAddress": "0x5CA39f9903B3e34941f0dF42aEe4C323D062Ff1A",
 "location": {
 "country": "Ecuador",
 "city": "Quito",
 "state": "Pichincha"
 }
 },
 "badge": {
 "@context": [
 "https://w3id.org/openbadges/v2",
 {
 "fields": {
 "@id": "http://127.0.0.1/metadata",
 "@type": "https://json-schema.org/draft/2020-12/json-schema-core.html"
 }
 }
 ],
 "type": "BadgeClass",

"id": "http://127.0.0.1/verification/issuer/0xDc6B8cD408A0054425dcE05045DEA50c601bE708/template/Certificado de programación Java EE",
 "name": "Certificado de programación Java EE",

"description": "Este es un certificado de ejemplo, certificado de programación emitido por la universidad.",
 "image": "",
 "criteria": {
 "type": "Criteria",
 "narrative": "Se deben pasar todos los módulos del curso"
 },
 "issuer": {
 "@context": "https://w3id.org/openbadges/v2",
 "type": "Profile",
 "id": "http://127.0.0.1/verification/issuer/0xDc6B8cD408A0054425dcE05045DEA50c601bE708",
 "name": "Universidad NFT CERTS",
 "url": "www.nftcerts.com",
 "email": "nftcerts@nft.com",
 "contactNumber": "5120914",
 "publicKey": "ecdsa-koblitz-pubkey:1QBiZ94RQ5iyGW2m8TDoaEmKohQzAjVWkH",
 "issuerAddress": "0xDc6B8cD408A0054425dcE05045DEA50c601bE708",
 "addedBy": "",
```

```
"location": {
"country": "Ecuador",
"city": "Quito",
"state": "Pichincha"
}
},
"tags": [
"Java",
"Software",
"Java EE",
"Programación"
],
"fields": [
{
"key": "Instructor",
"type": "text"
},
{
"key": "Calificación",
"type": "number"
},
{
"key": "Fecha de finalización del curso",
"type": "date"
}
]
},
"verification": {
"type": "HostedBadge",
"allowedOrigins": "http://127.0.0.1/"
},
"issuedOn": "2/09/2021",
"evidenceFile": "https://ipfs.io/ipfs/QmSoArZGoDGCHbFS4tPAqffZjDqZcxLUyzaUsUvuK4Hx4R",
"badgeFields": [
{
"key": "Instructor",
"value": "Juan Peres"
},
{
"key": "Calificación",
"value": "20"
},
{
"key": "Fecha de finalización del curso",
"value": "25/08/2021"
}
]
}
```

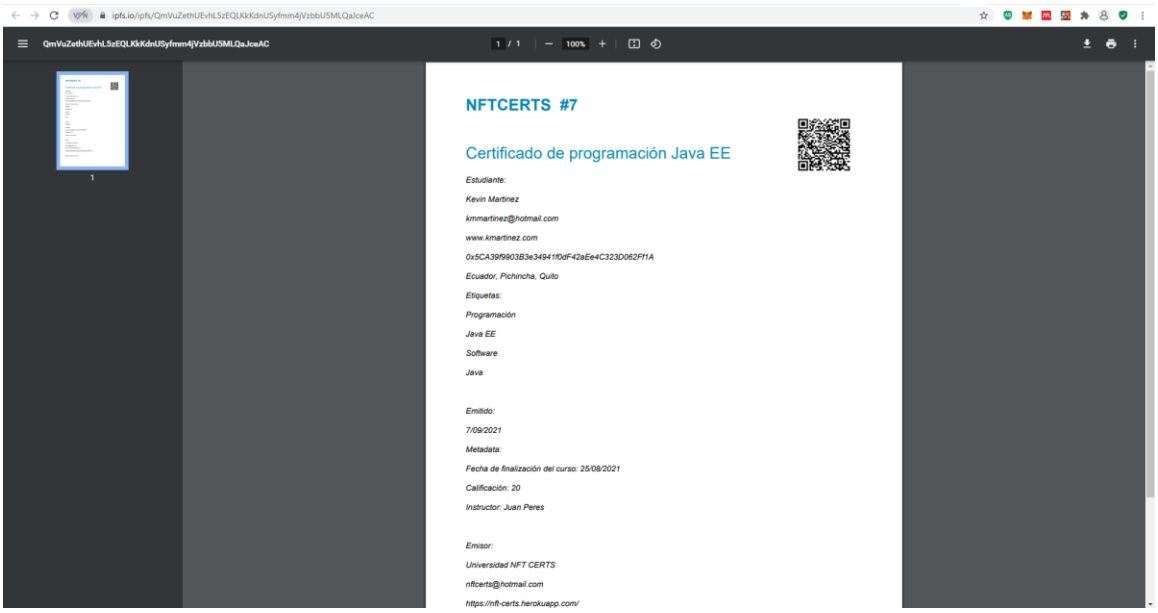

Figura 43.

Assertion Json desde IPFS

La columna archivo de la tabla contiene el enlace de IPFS hacia el documento electrónico almacenado y disponible como en la **Figura 44**. Par ambos casos los enlaces se construyen a partir del hash de la información registrada, por lo que su identificador es único e irrepetible. El sistema genera además un código QR que puede ser escaneado para acceder de manera directa a un enlace de validación dentro de la aplicación.

Figura 44.

Enlace de IPFS



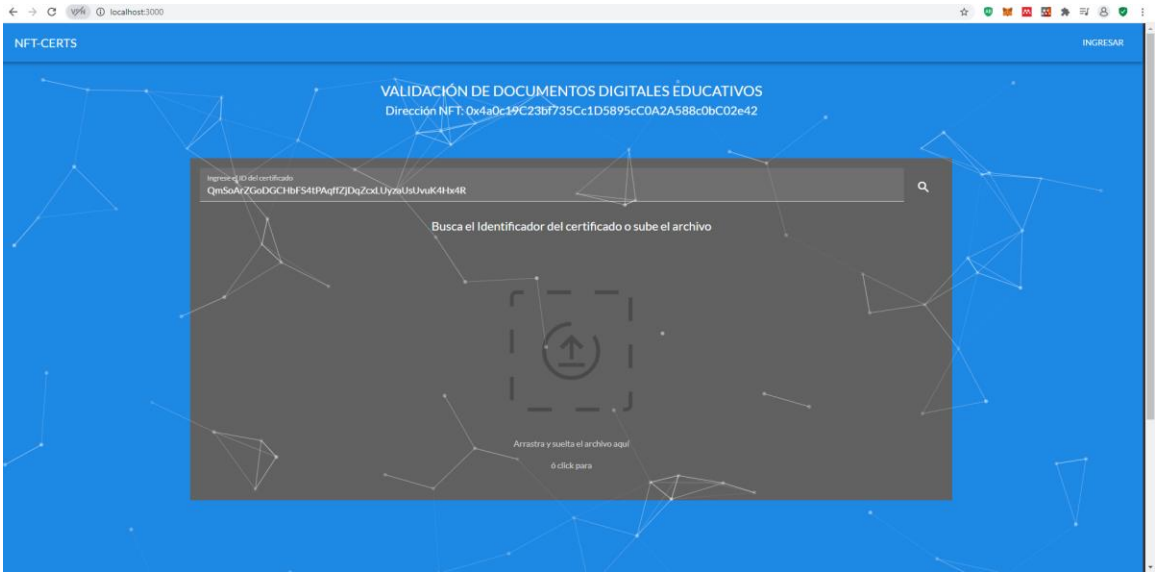
Para validar el certificado, se debe dirigir a la página principal de la **Figura 45**, en donde se encuentra un formulario que recibe el CID de un certificado emitido como también el archivo del correspondiente certificado.

En caso de escribir el CID del certificado se debe proceder dando clic en el ícono de la lupa.

En caso de subir el archivo solo se debe esperar a que éste sea respectivamente leído y validado.

Figura 45.

Validación de documentos digitales educativos

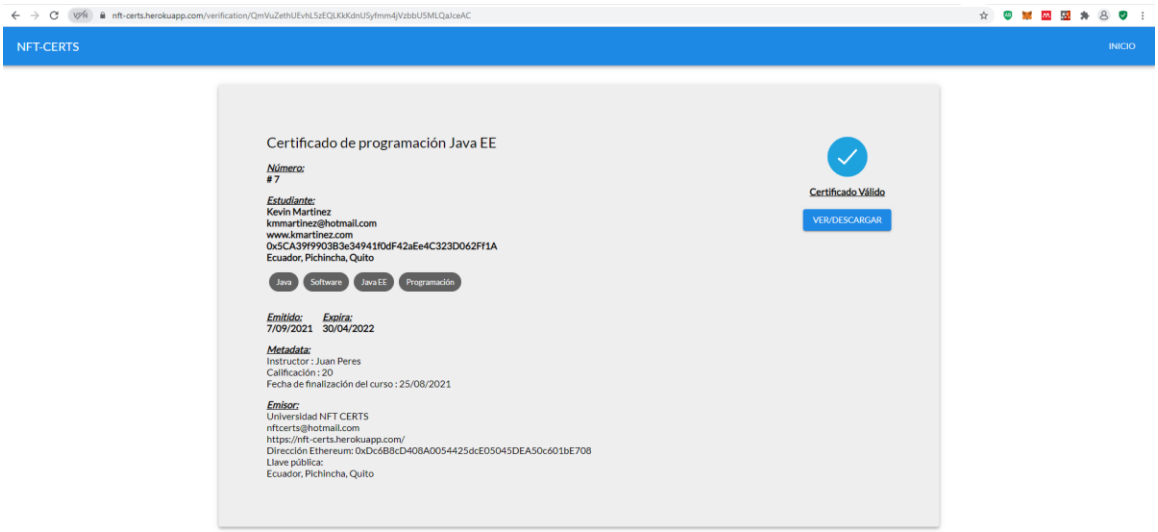


Una vez se han procesado los parámetros de búsqueda solicitados se valida el archivo con la información dentro de Blockchain almacenada por los contratos inteligentes, apareciendo la interfaz de la **Figura 46**, entre la cual se puede visualizar:

- Información del estudiante
- Nombre del certificado
- Validez del certificado
- Etiquetas correspondientes a la plantilla
- Fechas de emisión y expiración
- Metadata correspondiente a la plantilla
- Información del emisor del certificado

Figura 46.

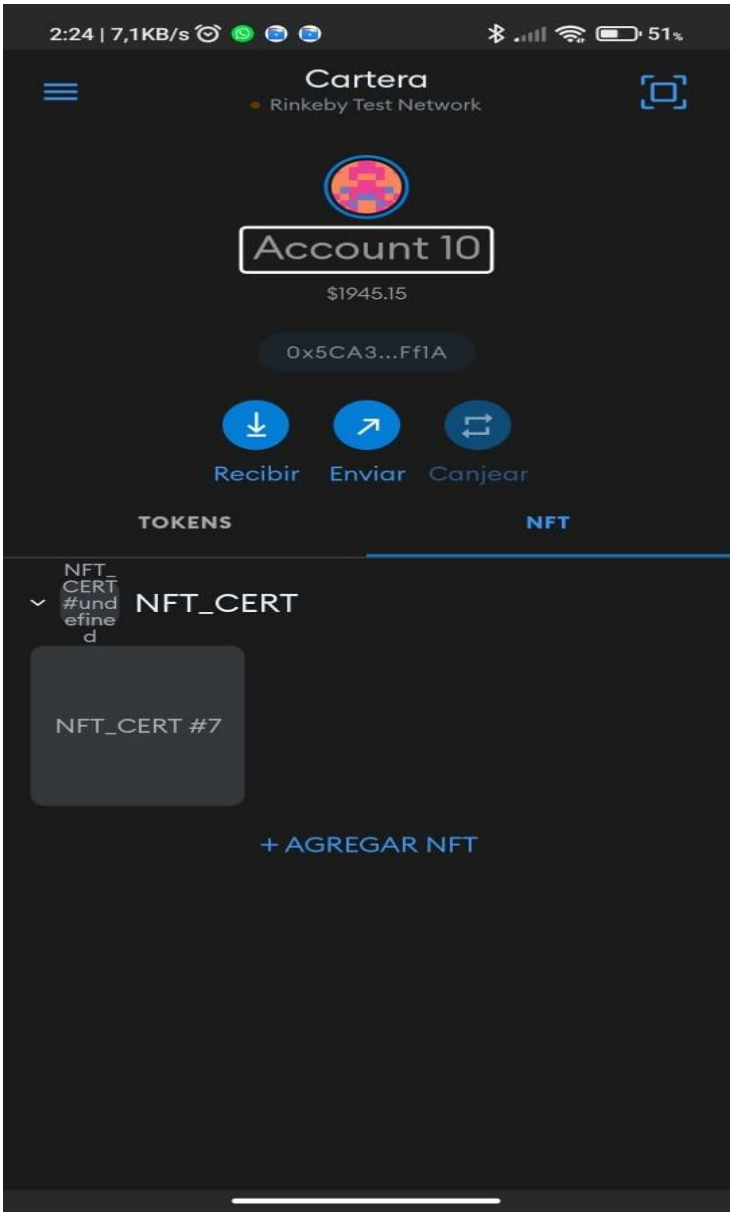
Certificado de programación JAVA EE



Al emitir el certificado es posible reclamar el NFT dentro de la cuenta de Ethereum del estudiante a través de la aplicación móvil de Metamask, puesto que hasta la fecha esta función no se encuentra disponible en la extensión del navegador. Además se debe realizar en el entorno de una red pública o a la que esta aplicación tenga acceso. Debido a esto se realizan los pasos anteriores en una red de Rinkeby, y se reclama el token utilizando la dirección del contrato NFT obteniendo el resultado de la **Figura 47.**

Figura 47.

Metamask mobile NFT



Capítulo V

Pruebas y resultados

Red de prueba Ganache

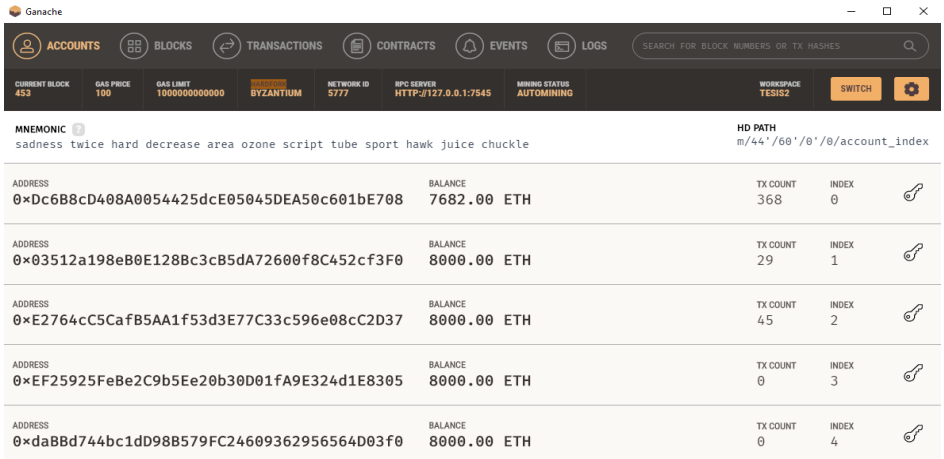
Para la realización de pruebas se considera una red de prueba desplegada con Ganache configurada con los siguientes parámetros:

- Precio del gas: 100
- Límite de gas: 1000000000000
- Hard fork: Byzantium
- Balance inicial: 8000 ETH

La red de prueba define un mnemónico a partir del cual se generan 5 cuentas Ethereum que se puede ver en la **Figura 48**, la primera cuenta generada se utilizó para realizar la transacción del despliegue de los contratos inteligentes, siendo esta la que asume los precios de gas.

Figura 48.

Ganache



Ejecución de casos de prueba

Se definen varios casos de prueba que siguen una secuencia lógica de las funcionalidades del sistema como en la **Figura 49**, teniendo en cada uno la llamada a una función existente en los contratos inteligentes.

Primero se define la cuenta de despliegue del contrato, esta será también el primer gestor de red del sistema, a partir del cual se realizaron las demás interacciones.

La primera prueba consiste en comprobar el despliegue del contrato, para lo cual se debe obtener como resultado una dirección de contrato de Ethereum que sea diferente a una dirección cero.

La segunda prueba realiza una comprobación para saber si la cuenta responsable de desplegar el contrato corresponde a la primera generada por el mnemónico.

La tercera prueba consiste en averiguar si la primera cuenta generada por el mnemónico se encuentra registrada como un gestor de red del sistema.

La cuarta prueba consiste en validar la siguiente cuenta generada por el mnemónico como una no registrada todavía. Esto para en la siguiente prueba registrarla como estudiante y volver a comprobar su perfil en la sexta prueba

Ocurre de manera similar con las pruebas séptima, octava y novena pero esta vez para repetir el ejercicio con un usuario emisor.

El resto de pruebas consisten en la inserción de data por parte de los usuarios gestor de red y emisor, en los cuales se prueban las funcionalidades de plantillas, emisión de certificados y revocación como se puede ver en la **Figura 49**.

Figura 49.

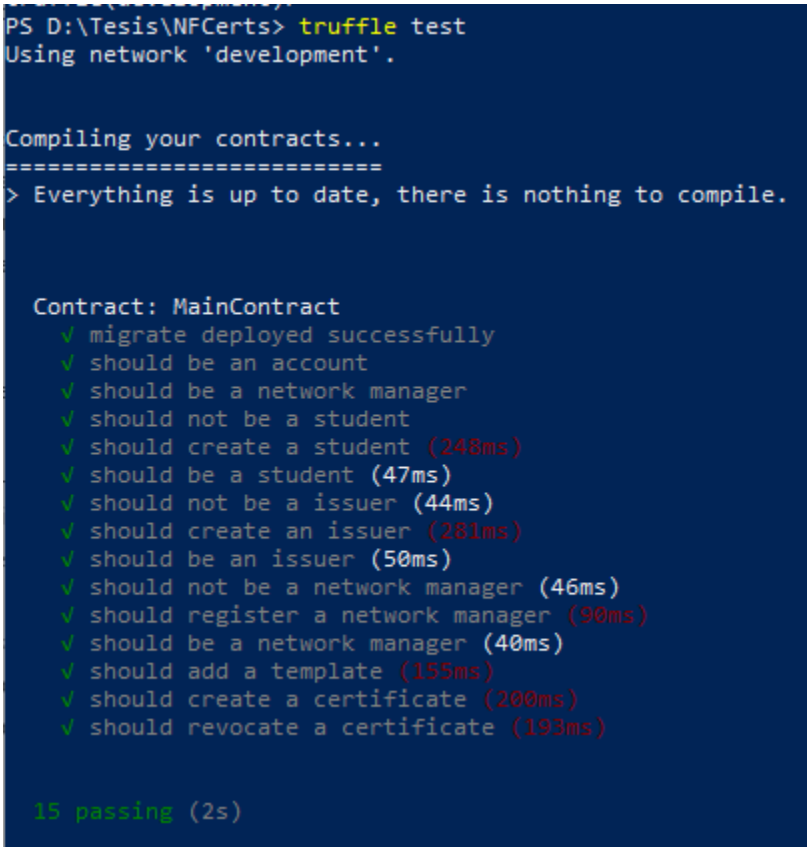
Casos de prueba

```
const MainContract = artifacts.require("Main")

contract("MainContract", (accounts) => {

    before(async () => {
        this.mainContract = await MainContract.deployed()
    })

    it('migrate deployed successfully', async () => {
        const address = this.mainContract.address
        assert.notEqual(address, null)
        assert.notEqual(address, undefined)
        assert.notEqual(address, 0x0)
        assert.notEqual(address, "")
    })

    it('should be an account', async () => {
        assert.equal(accounts[0], 0xDc6B8cD408A0054425dcE05045DEA50c601bE708, 'address is the first account')
    })

    it('should be a network manager', async () => {
        const managerFirstAccount = accounts[0]
        const isManagerRes = await this.mainContract.isNetworkManager(managerFirstAccount)
        assert.equal(isManagerRes, true, 'address is not a network manager')
    })

    it('should not be a student', async () => {
        const studentUnregisteredAccount = accounts[1]
        const isStudentRes = await this.mainContract.isStudent(studentUnregisteredAccount)
        assert.equal(isStudentRes, false, 'address is a student yet')
    })

    it('should create a student', async () => {
        const resStudentCreated = await this.mainContract.addStudent(
            'test student 1',           //student name
            'www.st1.com',              //url
            'st1@gmail.com',            //email
            '+593 123456789',           //contact number
            'ecdsa-koblitz-pubkey:1QBiZ94RQ5iyGW2m8TDoaEmKohQzAjVWkH', //public key
            'recp json link',           //recipient json
            'Ecuador',                  //country
            'Quito',                    //city
            'Pichincha',                //state
            { from: accounts[1] }
        );
        const resultLog = resStudentCreated.logs[0].args

        assert.equal(resultLog.studentAddress, accounts[1])
        assert.equal(resultLog.name, 'test student 1')
```

Al ejecutar los casos de prueba se puede apreciar que todos han pasado y se recibe un tiempo de respuesta que se registra para obtener los tiempos promedio en los que se ejecutan cada una de las llamadas al sistema como en la **Figura 50**.

Figura 50.

Ejecución de los casos de prueba



Resultados de casos de prueba. Una vez ejecutados los casos de prueba se puede observar que precisamente las llamadas que requieren escritura dentro de la red Blockchain presentan una mayor cantidad de trabajo reflejado en los tiempos de espera, a pesar de que esto se debería compensar aumentando la cantidad de gas en las transacciones, es probable que en una red con mayor cantidad de nodos estas tarifas lleguen a ser grandes, haciendo complicada la tarea de tener el suficiente balance para ejecutar las transacciones de un emisor, esto se pone a prueba a continuación con el despliegue de los Smart Contracts en una red pública de prueba.

Red de prueba Rinkeby

Para la realización de pruebas se considera una red de prueba desplegada con Ganache configurada con los siguientes parámetros, como se puede encontrar en la **Figura 51**:

- Precio del gas: 1000 gwei (equivalente a 0,000001 ETH)
- Límite de gas: 29970536
- Máxima tarifa por transacción: 29.97 ETH

Figura 51.

Rinkeby stats

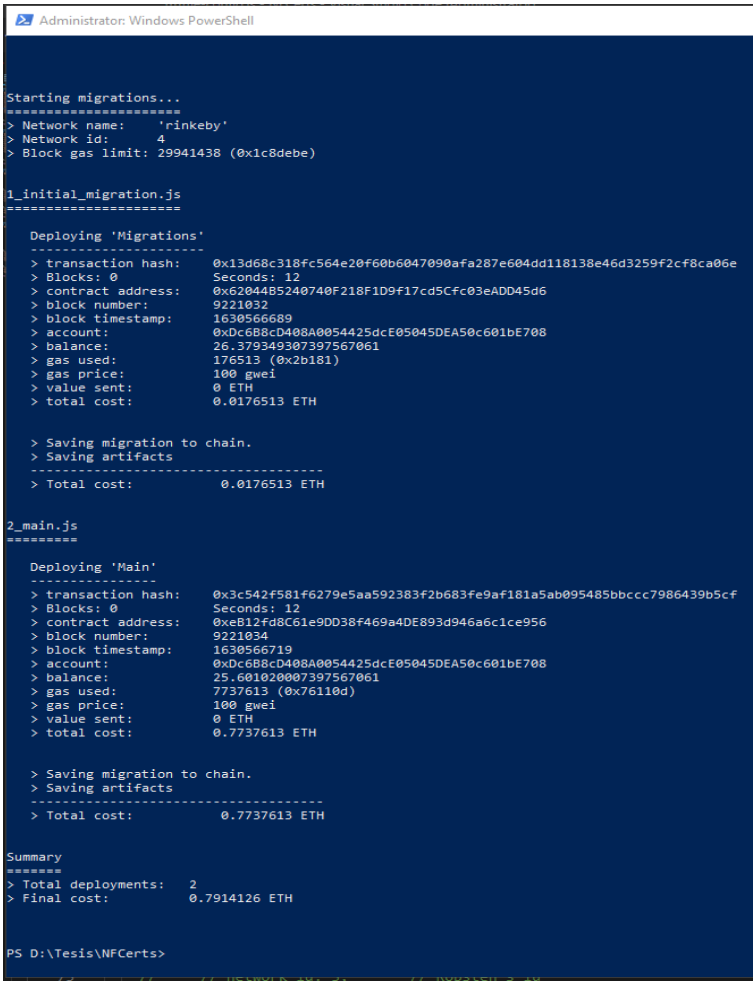
Nota. Se visualiza el monitoreo de la red pública, por (rinkeby.io, 2021)

Para la realización de estas pruebas se considera una conexión hacía un nodo como servicio conocido como Infura, el cual es accesible a través de un API para lograr una interacción con la red de prueba Rinkeby sin necesidad de configurar y sincronizar un nodo de la red.

Se realiza el despliegue de los contratos inteligentes en la red, teniendo como resultado, como se muestra en el coste final de la **Figura 52**, un valor de transacción de 0.7914126 ETH.

Figura 52.

Despliegue de Smart Contracts en Rinkeby



Una vez desplegados los contratos se procede a realizar pruebas a través de la aplicación cliente, y se registran los valores, resultado de las transacciones registrados en Metamask, obteniendo el costo por transacción de cada una de las operaciones de escritura en los Smart Contracts, los cuales nos permiten cuantificar el esfuerzo

computacional que implica ejecutar las operaciones. Estos valores se pueden apreciar en la **Tabla 19.**

Tabla 19.

Costos por transacción de Smart Contracts en Rinkeby

Prueba	Despliegue de la aplicación	Registro de Estudiante	Registro de Gestor de red	Registro de Emisor	Registro de plantilla	Emisión de Certificado	Registro de revocado
1	0,7800026	0,000931	0,000076	0,001588	0,000422	0,000505	0,000245
2	0,7914126	0,000976	0,000076	0,001531	0,000381	0,000491	0,000256
3	0,7826784	0,000951	0,000076	0,001486	0,000535	0,000553	0,000248
4	0,7819505	0,000937	0,000076	0,001562	0,000427	0,000535	0,000254
5	0,7885369	0,00096	0,000076	0,001504	0,000496	0,000506	0,000252
6	0,7845925	0,000971	0,000076	0,001548	0,000447	0,000496	0,000248
7	0,7837281	0,00096	0,000076	0,001487	0,0004	0,000502	0,000247
8	0,7836861	0,000939	0,000076	0,001514	0,000521	0,000499	0,000255
9	0,7820761	0,000934	0,000076	0,001558	0,000423	0,000536	0,000248
10	0,7889999	0,000966	0,000076	0,001555	0,000473	0,000513	0,00025
11	0,7899965	0,000967	0,000076	0,001532	0,00041	0,000531	0,000249
12	0,7892025	0,000949	0,000076	0,0015	0,000494	0,000525	0,000255
13	0,7892635	0,000939	0,000076	0,001519	0,000492	0,000506	0,000249
14	0,7817984	0,000971	0,000076	0,001535	0,00043	0,000497	0,000251
15	0,7824795	0,000951	0,000076	0,001548	0,000435	0,00051	0,000256
16	0,7898178	0,000962	0,000076	0,001564	0,000397	0,000528	0,000248
17	0,7812181	0,00096	0,000076	0,001576	0,000516	0,000515	0,000253
18	0,7871001	0,000967	0,000076	0,001565	0,000503	0,000533	0,000248
19	0,791115	0,000961	0,000076	0,001507	0,000399	0,000502	0,00025
20	0,7817502	0,000938	0,000076	0,001586	0,000505	0,000529	0,000251
21	0,7863935	0,000944	0,000076	0,001549	0,000408	0,000519	0,000252
22	0,7835676	0,000951	0,000076	0,00154	0,000509	0,000497	0,000253
23	0,7841019	0,000976	0,000076	0,001577	0,000498	0,0005	0,000247
24	0,7812913	0,000943	0,000076	0,001553	0,000398	0,000533	0,000248
25	0,7818261	0,000962	0,000076	0,001541	0,000535	0,000548	0,000255
26	0,7877553	0,000948	0,000076	0,001486	0,000437	0,000499	0,000247
27	0,7884867	0,000931	0,000076	0,001538	0,000419	0,000553	0,000255
28	0,7820279	0,000938	0,000076	0,00152	0,000388	0,000522	0,000256
29	0,7863599	0,000935	0,000076	0,001586	0,000424	0,000526	0,000248
30	0,789868	0,000964	0,000076	0,001578	0,000526	0,000536	0,00025
31	0,7802144	0,000958	0,000076	0,001487	0,000485	0,000523	0,000251
32	0,7817723	0,000961	0,000076	0,001535	0,000463	0,000521	0,000254
33	0,7892063	0,000932	0,000076	0,001557	0,000426	0,000516	0,000246
34	0,7866852	0,000941	0,000076	0,00154	0,00052	0,000528	0,000255
35	0,7812056	0,000947	0,000076	0,00158	0,000499	0,000527	0,000247
36	0,7817159	0,000968	0,000076	0,001568	0,000461	0,000534	0,000256
37	0,7889405	0,000941	0,000076	0,001487	0,000509	0,000531	0,000256
38	0,7814292	0,000966	0,000076	0,001505	0,000419	0,000534	0,000249
39	0,7851947	0,000944	0,000076	0,001577	0,000531	0,000534	0,000255
40	0,7905645	0,000962	0,000076	0,001549	0,000487	0,000537	0,000251
41	0,7847294	0,000952	0,000076	0,001527	0,000387	0,000541	0,000245
42	0,7903989	0,000962	0,000076	0,001515	0,000462	0,000521	0,000255
43	0,7858015	0,000957	0,000076	0,001564	0,000406	0,000491	0,000252
44	0,7878272	0,000965	0,000076	0,001498	0,000435	0,000506	0,000256
45	0,7823008	0,000953	0,000076	0,001505	0,000532	0,000538	0,00025
46	0,7830869	0,00097	0,000076	0,001563	0,00042	0,000498	0,000253
47	0,7907658	0,000943	0,000076	0,00149	0,000445	0,000497	0,000249
48	0,7904175	0,000934	0,000076	0,001534	0,000517	0,000516	0,00025
49	0,7841924	0,000976	0,000076	0,001558	0,000467	0,000527	0,000248
50	0,7900516	0,000957	0,000076	0,001579	0,000483	0,000498	0,000253
Promedio	**0,785591682**	**0,00095342**	**0,000076**	**0,00153902**	**0,00046004**	**0,00051926**	**0,0002511**

Considerando las mismas transacciones del apartado anterior, se procede también a realizar un conteo de los tiempos de respuesta en que las transacciones son minadas por la red y confirmadas. Para ello se considera el costo de transacción recomendado por Metamask, de manera que exista un tiempo aproximado de menos 30 segundos. Estos valores se pueden apreciar en la **Figura 54** y están representados en segundos**.**

Tabla 20.

Tiempos de respuesta Rinkeby

Prueba	Despliegue de la aplicación	Registro de Estudiante	Registro de Gestor de red	Registro de Emisor	Registro de plantilla	Emisión de Certificado	Registro de revocado
1	25	23	14	30	13	24	30
2	28	19	10	16	11	11	29
3	26	21	26	29	23	24	10
4	22	13	12	13	21	26	21
5	25	21	24	22	22	13	15
6	28	20	26	31	28	26	20
7	16	23	19	27	28	30	11
8	14	31	11	15	15	28	15
9	15	31	26	17	21	21	23
10	31	14	22	26	14	16	24
11	25	20	18	24	22	17	26
12	21	27	24	31	12	16	26
13	10	18	21	31	27	15	11
14	28	18	25	18	21	16	22
15	14	13	11	25	26	12	24
16	13	18	31	13	26	27	27
17	24	20	22	13	27	17	22
18	29	14	10	26	10	29	14
19	23	18	22	23	12	10	29
20	16	27	15	27	28	28	29
21	20	18	27	26	29	18	21
22	23	15	22	21	15	11	16
23	14	12	18	18	12	19	25
24	10	24	26	30	22	10	28
25	26	16	17	27	19	31	14
26	31	25	28	10	31	18	13
27	23	10	28	14	24	20	31
28	10	14	24	20	12	20	13
29	13	26	29	27	21	26	20
30	24	21	11	27	28	28	18
31	20	19	14	27	21	31	12
32	29	28	28	10	17	10	27
33	20	26	28	13	29	24	22
34	21	18	12	12	31	28	26
35	16	15	25	17	11	24	17
36	14	23	31	20	31	27	16
37	17	25	29	11	13	23	20
38	15	30	16	18	10	18	10
39	21	23	28	25	20	20	15
40	10	12	18	16	20	26	12
41	17	28	14	15	28	18	12
42	27	22	29	29	21	10	13
43	22	17	30	22	12	15	23
44	23	22	28	12	30	18	30
45	16	22	29	28	25	25	16
46	30	11	13	21	29	10	15
47	28	25	14	14	10	16	27
48	26	17	26	16	26	28	19
49	17	17	26	21	24	27	25
50	13	13	28	29	18	22	10
Promedio	**20,58**	**20,06**	**21,7**	**21,06**	**20,92**	**20,54**	**19,88**

Evaluación de rendimiento

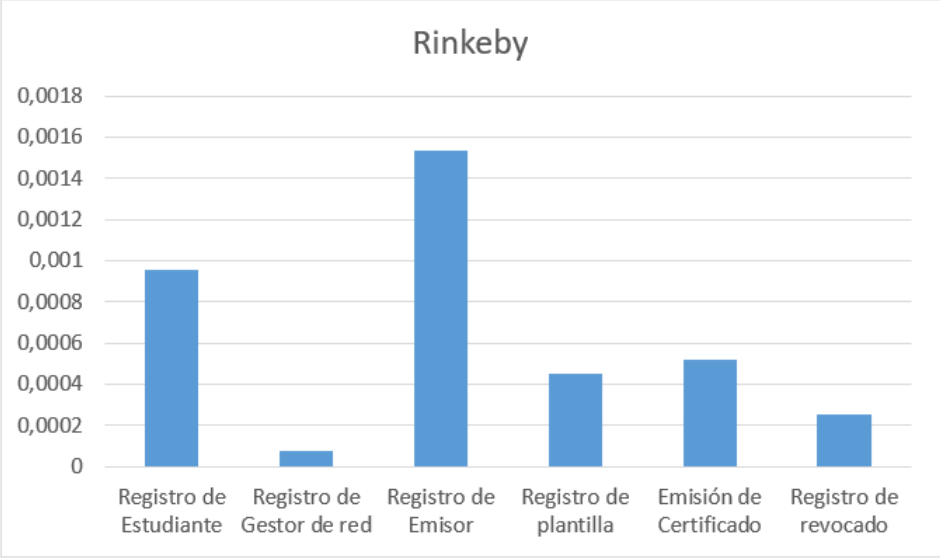
Para tener una mejor percepción del rendimiento del sistema, se realizan pruebas en un entorno que simula a los contratos inteligentes en producción dentro de una red pública, así como el despliegue de la aplicación cliente en una instancia de Cloud para poder acceder a través de internet.

Como muestra el gráfico de la **Figura 53**, las operaciones que más costo por transacción implican son las de Registro de Emisor y Registro de Alumno, implicando un costo aproximado de 0.001536 y 0.000951 ETH respectivamente.

Considerando el valor de la máxima tarifa por transacción de la red, se puede concluir que, a excepción del despliegue del contrato, el cual se realiza solo una vez, el resto de operaciones presentan una tarifa relativamente baja y accesible para los usuarios, debido a que no son valores realmente significativos.

Figura 53.

Comparación de costos por transacción de Smart Contracts

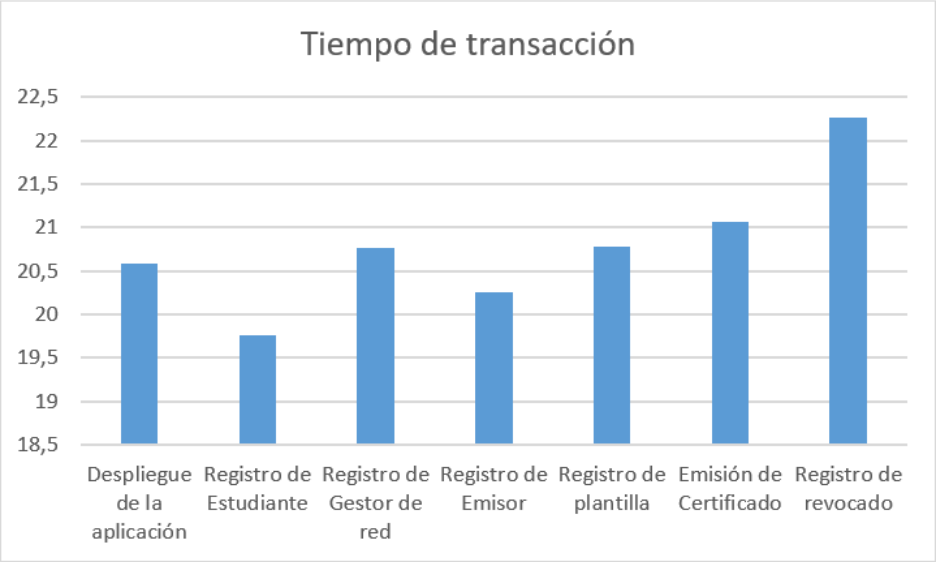


Sin embargo, se debe tener en cuenta que las operaciones de Registro de plantilla y Emisión de certificado, son claramente las más utilizadas dentro del sistema, pues el emisor puede crear tantas plantillas como necesite y emitir tantos certificados como requiera hacia varios estudiantes.

Se consideran además los tiempos promedio de respuesta recolectados en la **Tabla 20** para estimar un promedio en cada caso, como se puede ver en la **Figura 54**, la red es capaz de establecer un tiempo de respuesta óptimo de menos 30 segundos enviando solo la cantidad de gas suficiente para cada caso. Valores que corresponden a la **Tabla 19**.

Figura 54.

Tiempo de transacción promedio



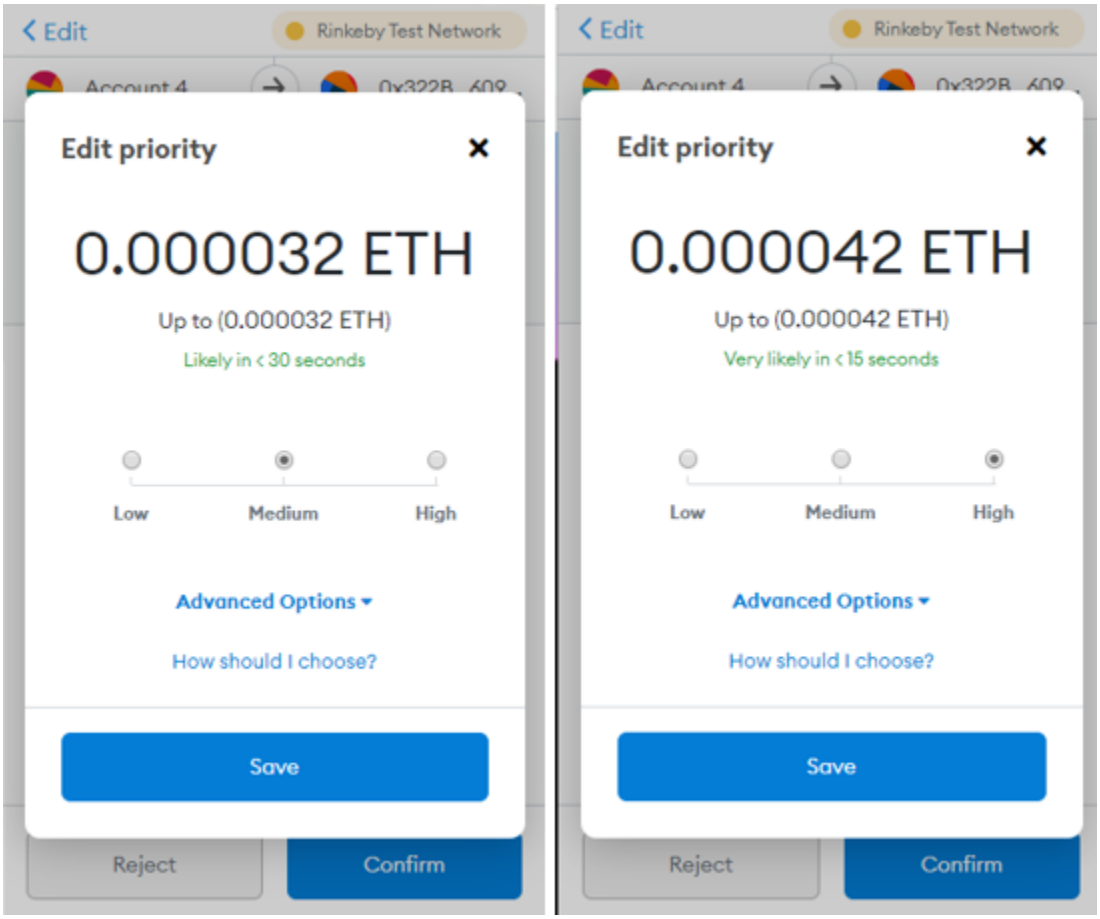
Como se puede evidenciar, las transacciones que requieren de mayor cantidad de gas requieren un mayor costo de transacción para poder ser minadas en menos de 30 segundos, a diferencia de las transacciones menos costosas, pero que de igual modo al requerir menos gas resultan en un tiempo de respuesta similar. Dependiendo

del caso es posible optimizar los tiempos de respuesta estableciendo una mayor cantidad de gas en las transacciones como se puede ver en la **Figura 55**.

Esto se puede lograr de manera sencilla con la aplicación Metamask, que brinda las mencionadas utilidades junto con estimaciones de costos y tiempo. En el ejemplo una transacción de alrededor de 30 segundos puede disminuir a la mitad aumentando de 0.000032 ETH a 0.000042 ETH.

Figura 55.

Optimización de transacciones



Evaluación de los aspectos de seguridad

Los contratos inteligentes han sido desarrollados teniendo en consideración las recomendaciones de seguridad de Solidity, esto debido a que, si bien existe la propiedad de inmutabilidad de la red, un contrato inteligente que no ha sido correctamente programado teniendo en cuenta a través de qué direcciones se pueden acceder a los datos, puede llevar a la perdida de la integridad y confidencialidad de la información. Cabe mencionar que

Los contratos se han previamente probado con la IDE Remix, que se encuentra disponible a través de internet, y brinda un entorno de pruebas seguro en donde se puede evaluar contratos inteligentes sin necesidad de preocuparse por el costo de transacciones o despliegue del mismo.

Durante el desarrollo de los contratos inteligentes se establecen restricciones de acceso y validaciones de seguridad a determinadas funcionalidades como son:

- Las direcciones a registrar deben ser direcciones externas válidas (External Owner Address) y no direcciones de contrato.
- Las direcciones de usuario a registrar no deben haber sido registradas previamente.
- Las operaciones correspondientes a los usuarios con perfil de Emisor y Gestor de red deben ser solicitadas por una dirección con el correspondiente perfil.
- La emisión de certificados solamente se puede realizar por usuarios Emisores y deben ser dirigidas a usuarios registrados como estudiantes.
- Para revocar un certificado, además de solicitarse la operación por un Emisor, éste debe ser el usuario que ha emitido el certificado previamente.

Se evalúan aspectos relacionados con la seguridad informática a continuación:

- Disponibilidad: La elección de una red pública como es Ethereum, y cuyos contratos en producción se despliegan sobre la red principal, u otras alternativas como Ropsten o Rikeby, aseguran que los contratos estarán disponibles de manera permanente debido a la cantidad de nodos que poseen este tipo de redes.
- Confidencialidad: Si bien la información sobre las habilidades demostradas en un certificado educativo, así como la valides del mismo y la identidad el poseedor se consideraron de carácter público, el sistema puede mantener la confidencialidad, se mantienen ciertos aspectos que mantienen confidencialidad de la información entre el emisor y el estudiante, como es el hecho de que el emisor deba conocer la dirección de Ethereum específica del estudiante para poder emitir certificados.
- Integridad: Este objetivo de la seguridad se cumple en el proyecto al utilizar estrategias de almacenamiento y validación de la información basadas en criptografía y el uso de códigos hash, que permiten identificar de manera exacta la modificación de un documento.
- Trazabilidad: El sistema mantiene por su cuenta una trazabilidad de toda aquella información que se escribe en Blockchain, en el caso de usuarios o certificados, siendo posible saber quién realizó el registro o la emisión de estos. Además, se debe considerar las herramientas existentes en redes públicas, que monitorean y permiten verificar las transacciones ocurridas dentro de la red.

- Autenticación y no repudio: El sistema requiere detectar las cuentas registradas en los contratos inteligentes para poder realizar la autenticación de los usuarios, esto significa que el usuario debe afirmar que es el poseedor de dicha cuenta desde su navegador, esto a través de una E-Wallet que custodia las claves pública y privada de la cuenta de usuario. Gracias a esto el usuario debe firmar las transacciones con sus llaves asegurando de esta manera que fue este quien las ejecutó, ganando de esta manera no repudio.

Capítulo VI

Conclusiones, Recomendaciones y Trabajo a futuro

Conclusiones

Se logró desarrollar un sistema Basado en Blockchain, que permite el almacenamiento y validación de certificados educativos manteniendo la integridad de los documentos emitidos por una entidad emisora con el uso de identificadores de contenido basados en hash, manteniendo una alta disponibilidad y seguridad gracias al uso de técnicas criptográficas.

La revisión literaria del estado del arte permitió conocer las últimas tendencias en el desarrollo de aplicaciones basadas en Blockchain, así como sus casos de éxito en el ámbito de la validación y seguridad, lo que permitió prestar especial atención a las soluciones que utilizan contratos inteligentes, los cuales tienen una gran capacidad de adaptarse a cualquier requerimiento y que en conjunto con el sistema de archivos interplanetario IPFS brindaron las condiciones necesarias para proponer sistema de validación documental.

El uso de las metodologías XP y Scrum permitió desarrollar de manera eficiente el prototipo del sistema manteniendo un entorno de trabajo basado en pruebas y que permitió soportar cambios, implementación de nuevas funcionalidades e integraciones con estándares, librerías, frameworks, redes de desarrollo, etc. Además de dar versatilidad en la búsqueda de información y análisis. Dando como resultado un prototipo completamente funcional que cumple con los requerimientos propuestos.

La evaluación de los casos de prueba, así como de rendimiento del sistema muestran que los contratos inteligentes realizan sus operaciones de manera exitosa, tanto en escritura de información dentro de Blockchain como en la validación de

usuarios que pueden ejecutar estas tareas y sus tiempos de respuesta dependen enteramente de la cantidad de trabajo cuantificado en cantidad de gas que se establezca para estos, realizando un procesamiento de las transacciones en tiempos más cortos si se incrementa dicha variable.

El sistema cumple con los casos de uso fundamentales de la seguridad de la información, como son disponibilidad, integridad, trazabilidad, no repudio, y confidencialidad. Pero teniendo en cuenta que la información concerniente a estudiantes, emisores y certificados se considera de carácter público.

La utilización del estándar OpenBadges, así como de la especificación ERC 721 para la representación de los certificados educativos en el sistema resulta de gran utilidad para crear información confiable, debido a que la metadata, información complementaria y de propiedad se encuentran enlazadas y disponibles tanto dentro del sistema como en el sistema de archivos IPFS.

Los componentes de la solución permiten una adecuada interacción con el usuario, teniendo interfaces gráficas intuitivas y amigables, sin embargo, el uso de las direcciones de Ethereum, así como los identificadores de contenido CID de los documentos almacenados en IPFS, pueden traer confusión a los usuarios debido a la longitud de los mismos, pues llegan a tener problemas al identificarlos, a pesar de esto, se sigue considerando una buena alternativa en vista de mantener integridad de la información en el sistema.

La utilización de contratos inteligentes, en conjunto con las especificaciones de Ethereum Request for comments brinda una versatilidad a la solución capaz de adaptarse al crecimiento del proyecto a futuro, teniendo en cuenta que, para realizar

cualquier cambio, los contratos deben volver a desplegarse en la red de Ethereum, lo cual conlleva que la información contenida previamente se perderá.

La combinación de un almacenamiento mixto con las redes de Blockchain e IPFS, en conjunto con el estándar Open Badges y la especificación ERC 721, ofrecen una alternativa descentralizada, eliminan la necesidad de intermediarios en el proceso de validación de certificados educativos, así como costos de almacenamiento físico y transporte de los mismos, pero se debe tener en cuenta los costes de transacciones realizadas dentro de la red, como el mantenimiento de ésta en caso de que se deseen configurar y sincronizar nuevos nodos.

Recomendaciones

El proceso de escritura en la red de Ethereum presenta costos para el usuario que ejecuta las transacciones, por tanto, siendo el usuario emisor el que realiza la mayor cantidad de tareas que requieren escritura e implican tarifas de gas, es recomendable que éste se encuentre como un nodo minero de manera que reciba el balance suficiente para compensar el coste de las operaciones que éste debe realizar.

Se debe considerar la realización de pruebas en redes públicas, pues ayudará a apreciar de mejor manera cuales son los costos y tarifas por la ejecución de los procesos del sistema e identificar cuales se podrían optimizar para reducir los mismos.

Se debe tener especial cuidado al momento de programar los contratos inteligentes, pues si bien Blockchain se toma como una garantía de inmutabilidad de la información, un contrato inteligente que cuente con propiedades de escritura de información que no estén debidamente protegidas puede llegar a tener fallas de seguridad importantes. Revisar las recomendaciones de seguridad de Solidity.

Trabajo a futuro

La línea de trabajo a futuro es extensa, se tiene planificada una segunda versión del prototipo, dónde las funcionalidades de validación y verificación de certificados educativos puedan integrarse a la herramienta de gestión de aprendizaje Moodle, siendo necesaria una evaluación para integrar esta compatibilidad con los componentes del sistema.

Adicionalmente se planea mejorar la herramienta de creación de plantillas para que sea capaz de personalizar la interfaz de los certificados educativos agregando estilos y propiedades gráficas.

Al tener la base de un sistema de almacenamiento para certificados educativos, esta podría comenzar a considerarse una alternativa para las instituciones de educación, así como para las entidades de control, que legalizan la validez de un certificado en un determinado sector o región, pudiendo ser estos los siguientes participantes a considerar dentro de la solución.

La validación de información académica internacional es otro de los grandes casos de uso en la certificación académica, pues si bien al tener un estándar como Open Bages, que funciona de manera global, no en todos los casos el resultado de un aprendizaje es equivalente al cambiar de región, por tanto, es un reto a considerar en virtud de desarrollar un sistema de validación útil en todas las regiones del mundo.

Se considera que el sistema puede mejorar su escalabilidad, así como el manejo de las transacciones de Ethereum. Existen soluciones que pueden brindar una manera más eficiente de cobrar por las transacciones sin necesidad que el cliente deba contar con balance para realizar la transacción, mediante las conocidas estaciones de gas, que se encargan de pagar las tarifas por ciertas transacciones, que sería muy útil al

momento de realizar el registro de estudiantes, entre otras funcionalidades. Además, se debe considerar la próxima llegada de la nueva versión de Ethereum, conocida como Ethereum 2.0, que promete resolver en gran medida problemas de escalabilidad y seguridad orientados al desarrollo de DApps.

Bibliografía

0xcert. (2020). *0xcert - EthHub*. https://unlock-protocol.github.io/ethhub/built-on-ethereum/infrastructure/0xcert/

Almeida, E. (2018). El papel de la seguridad en la Industria 4.0 - CIO MX. *CIO*. https://cio.com.mx/papel-la-seguridad-en-la-industria-4-0/

Amores, A. (2020). *Blockchain, algoritmos de consenso*.

Barrios, M. (2017). *IPFS: Interplanetary file system*.

BBVA. (2020). *Qué son los "smart contracts" o contratos inteligentes | BBVA*. https://www.bbva.com/es/smart-contracts-los-contratos-basados-blockchain-no-necesitan-abogados/

Blockerts. (2018). *Introduction - Blockcerts : The Open Standard for Blockchain Credentials*. https://www.blockcerts.org/guide/

Campaña, X. (2020). *Métodos de consenso sobre plataformas blockchain*.

Carrillo, A., & Roa, L. (2020). Desarrollo de un sistema para recuado y pago de tarifas de transporte público intermunicipal mediante tecnología Blockchain. *ResearchGate*.

Carrizosa, S. (2014, January 12). La deserción puede con los cursos 'online' | Economía | EL PAÍS. *El País*. https://elpais.com/economia/2014/01/10/actualidad/1389360489_728192.html

Casas, D. (2019). *Aproximación basada en blockchain para modelo de confianza en la enseñanza superior.*

Casino, F., Thomas, D., & Constantinos, P. (2019). A systematic literature review of

blockchain-based applications. *Telematics and Informatics*.

Cetina, C. (2020). *Blockchain e integridad: aplicaciones de política pública*. https://scioteca.caf.com/handle/123456789/1651

Chadwick, M. S., & Longley, D. (2021). *Verifiable Credentials Data Model 1.0*. https://w3c.github.io/vc-data-model/#what-is-a-verifiable-credential

Chávez, E. (2018). Industria 4.0: la automatización de las cosas. *Business Empresarial*. http://www.businessempresarial.com.pe/industria-4-0-la-automatizacion-de-las-cosas/

Cheng, J. C., Lee, N. Y., Chi, C., & Chen, Y. H. (2018). Blockchain and smart contract for digital certificate. *Proceedings of 4th IEEE International Conference on Applied System Innovation 2018, ICASI 2018*, 1046–1051. https://doi.org/10.1109/ICASI.2018.8394455

Chila, A., Gómez, I., & Lanza, F. (2020). Plataforma tecnológica de voto electrónico para elección de cuerpos colegiados de una universidad, que integra módulos blockchain y ciberseguridad. *ACOFI*.

Cortés, M. (2019). *Confianza digital, el camino hacia la innovación*. CIO. https://cio.com.mx/confianza-digital-el-camino-hacia-la-innovacion/

Credly. (2021). *Digital Credentials*. https://info.credly.com/

Derecho Ecuador. (2018). *Derecho Ecuador - Código Orgánico Integral Penal*. https://www.derechoecuador.com/codigo-organico-integral-penal-

Dolores, C. (2017). *LA BLOCKCHAIN: Fundamentos y propiedades*.

Ethereum. (2020). *Documentación de desarrollo de Ethereum | ethereum.org*.

https://ethereum.org/es/developers/docs/

Ethers. (2019). *JsonRpcProvider*. https://docs.ethers.io/v5/api/providers/jsonrpc-provider/

Fernández, X. (2018). *Gestión de identidades descentralizadas con Blockchain*.

Gartner. (2019). *Blockchain to Have Transformational Impact | Gartner*. https://www.gartner.com/en/newsroom/press-releases/2019-09-12-gartner-2019-hype-cycle-for-blockchain-business-shows

Gonzáles, C. (2021). *El documento electrónico como documento electrónico de archivo: componentes y características - Soaint*. Soaint. https://soaint.com/el-documento-electronico-como-documento-electronico-de-archivo-componentes-y-caracteristicas/

Gonzáles Glenda. (2018). *Certificaciones digitales registradas en blockchain, la nueva tendencia en educación*. Criptonoticias. https://www.criptonoticias.com/educacion/certificaciones-digitales-blockchain-educacion/

Guarín Cardona, N. (2019). *Blockchain, la tokenización de la economia y democratización de la inversión*. http://diposit.ub.edu/dspace/handle/2445/144157

Gupta, M. (2017). *Blockchain* (2018 John Wiley & Sons, Incorporated (Ed.); 2nd ed.). IMB.

Hernández Juárez, F. (2020). *La web descentralizada, un reto en internet | Decimo Septima Edicion - Agrotecnología*. https://revistaecys.github.io/17Edicion/articulo11.html

Kushmaro, P. (2018). How blockchain is impacting information security in companies.

CIO. https://www.cio.com/article/3293449/how-blockchain-is-impacting-information-security-in-companies.html

La Hora. (2003). Falsificación de documentos, delito que crece sin control. *La Hora*. https://lahora.com.ec/noticia/1000194134/falsificacin-de-documentos-delito-que-crece-sin-control

Miranda Palacios, V. (2018). *Explorando la Blockchain de Ethereum y el desarrollo de smart contracts* [Universitat Politècnica de Catalunya]. https://upcommons.upc.edu/handle/2117/127784

Morales Morales, M., Rosero Correa, L., & Morales Cardoso, S. (2020). Registro de títulos académicos mediante una aplicación basada en Blockchain y Smart Contracts. *Cátedra*, *3*(2), 73–98. https://doi.org/10.29166/catedra.v3i2.2200

Moreno, M. (2020). *Conceptos asociados a la blockchain*.

MyChoice2Pay. (2020). *¿Qué es un e-wallet o billetera electrónica? | MyChoice2Pay*. https://www.mychoice2pay.com/es/blog/que-es-ewallet

Navarro, A., Fernándezl, J., & Morales Vélez, J. (2013). *Revisión de metodologías ágiles para el desarrollo de software - Dialnet*. https://dialnet.unirioja.es/servlet/articulo?codigo=4752083

Novoa Z., E., Escobar M., C., Cajas A, M. J., & Fuentes O., L. (2020). Los Smart Contracts como alternativa para la modernización de recaudación tributaria en Ecuador. *Iuris Dictio*, 19. https://doi.org/10.18272/IU.V26I26.1831

Open Badges. (2016). *Guía sobre Mozilla Open Badges para centros de educación de adultos*.

OpenZeppelin. (2021). *GitHub - OpenZeppelin/openzeppelin-contracts: OpenZeppelin*

Contracts is a library for secure smart contract development. https://github.com/OpenZeppelin/openzeppelin-contracts

Pinasco, G. (2019, May 27). *Vistazo*. Ecuador Está 20 Años Atrasado En Innovación Científica. https://www.vistazo.com/estilo-de-vida/ciencia/ecuador-esta-anos-atrasado-en-innovacion-cientifica-KEVI137836

Preukschat, A. (2017). *Blockchain: La revolución industrial del internet.*

ProtoSchool. (2019). *Multiformats Tutorial* | *Anatomy of a CID (Lesson 1)* | *ProtoSchool*. https://proto.school/anatomy-of-a-cid/01

Ramirez Valencia, J. P. (2019). *SMART CONTRACTS*. https://www.researchgate.net/publication/336994225_CONTRATOS_INTELIGENTES_SMART_CONTRACTS

React. (2021). *React – Una biblioteca de JavaScript para construir interfaces de usuario*. https://es.reactjs.org/

Report, S. of A. (2020). *10th State of Agile Report*. 3–5.

REYES DELGADO, D. F. (2018). *APLICACIÓN DE BLOCKCHAIN PARA LA SEGURIDAD DE LOS DATOS DEL INTERNET OF THINGS* [UNIVERSIDAD TECNICA FEDERICO SANTA MARIA]. https://repositorio.usm.cl/handle/11673/47827

rinkeby.io. (2021). *Rinkeby: Network Dashboard*. https://www.rinkeby.io/#stats

Rodriguez, D. (2018). *TOKEN ERC: estándares más comunes e implementación TOKEN ERC: estándares más comunes e implementación*. https://blogthinkbig.com/token-estandares

Rosales, B. (2018). *BOLETÍN ANALÍTICO*. www.educacionsuperior.gob.ec

Sai, G., Unique, P., & Vineesha, P. (2019). *A BLOCKCHAIN BASED MICRO-CREDENTIALING SYSTEM*. ResearchGate. https://www.researchgate.net/figure/Use-Case-Diagram-of-the-proposed-system_fig2_333449543

Sánchez, M. (2015). *Pruebas de Software. Fundamentos y Técnicas*. UNIVERSIDAD POLITÉCNICA DE MADRID.

Sandner, P. (2017). *Comparison of Ethereum, Hyperledger Fabric and Corda*. https://philippsandner.medium.com/comparison-of-ethereum-hyperledger-fabric-and-corda-21c1bb9442f6

Solidity. (2021). *Solidity — documentación de Solidity*. https://solidity-es.readthedocs.io/es/latest/

Títulos falsos se ofertan en páginas de internet. (2013, December 2). *El Telégrafo*. https://www.eltelegrafo.com.ec/noticias/justicia/1/titulos-falsos-se-ofertan-en-paginas-de-internet

Torres, L., Penadés, C., & Orlando, P. (2017). *Metodologías ágiles para el desarrollo de software: eXtreme Programming*.

Tovar, M. (2020). *¿Puede 'blockchain' cambiar la forma en que compramos casas?* BBVA. https://www.bbva.com/es/puede-blockchain-cambiar-la-forma-en-que-compramos-casas/

Trigás, M., Cristina, A., & Troncho, D. (2018). *Desarrollo detallado de la fase de aprobación de un proyecto informático mediante el uso de metodologías ágiles*.

TruffleSuite. (2021a). *Ganache | Overview | Documentation | Truffle Suite*.

https://www.trufflesuite.com/docs/ganache/overview

TruffleSuite. (2021b). *Truffle | Overview | Documentation | Truffle Suite*. https://www.trufflesuite.com/docs/truffle/overview

Trujillo, S. (2017). Evolución de la tecnología de la cadena de bloques Ideas provenientes de la plataforma GitHub Un reporte de investigación del Deloitte Center for Financial Services Evolución de la tecnología de la cadena de bloques. In *Deloitte Insights* (p. 24). https://dupress.deloitte.com/dup-us-en/industry/financial-services/evolution-of-blockchain-github-platform.html.

Turkanović, M., Hölbl, M., Košič, K., Heričko, M., & Kamišalić, A. (2018). EduCTX: A blockchain-based higher education credit platform. In *IEEE Access* (Vol. 6). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ACCESS.2018.2789929

Typescriptlang.org. (2021). *TypeScript: JavaScript With Syntax For Types.* https://www.typescriptlang.org/

Udemy. (2019). *Udemy*. https://www.udemy.com/

Voutssas M, J. (2010). Preservación documental digital y seguridad informática. *Investigacion Bibliotecologica*, *24*(50), 127–155. https://doi.org/10.22201/IIBI.0187358XP.2010.50.21416

web3 js. (2021). *web3.js - Ethereum JavaScript API — web3.js 1.0.0 documentation*. https://web3js.readthedocs.io/en/v1.4.0/

Xertify. (2021). *Emitir Documentos Digitales | Blockchain Certificados - Xertify*. https://xertify.co/

Yogaterol. (2017). *Wallet de Ethereum: ¿Qué es Metamask?*

https://platzi.com/blog/wallet-ethereum-metamask/

Anexos