Facultad de Energía, las Industrias y los Recursos Naturales no Renovables

CARRERA DE INGENIERÍA EN SISTEMAS

Modelo de solución mediante el despliegue de un Contrato Inteligente en la red Blockchain de (EVM) para un módulo de registro de usuarios

TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS.

Autor:

- Jhonny Fernando Carrión Ramírez

Director:

- Ing. Cristian Ramiro Narváez Guillén, Mg.Sc.

LOJA - ECUADOR
2022

Certificación:

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

CERTIFICA:

Que el egresado **Jhonny Fernando Carrión Ramírez,** autor del presente trabajo de titulación, cuyo tema versa sobre "**MODELO DE SOLUCIÓN MEDIANTE EL DESPLIEGUE DE UN CONTRATO INTELIGENTE EN LA RED BLOCKCHAIN DE (EVM) PARA UN MÓDULO DE REGISTRO DE USUARIOS**", ha sido dirigido, orientado, discutido bajo mi asesoramiento y ha sido culminado al 100%, reúne a satisfacción los requisitos exigidos en una investigación de este nivel por lo cual autorizo su presentación y sustentación.

Loja, 27 de agosto del 2021

Firmado electrónicamente por:
CRISTIAN RAMIRO NARVAEZ GUILLEN

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

Autoría

Yo **Jhonny Fernando Carrión Ramírez,** declaro ser autor del presente trabajo de titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido del mismo.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi trabajo de titulación en el Repositorio Institucional - Biblioteca Virtual.

Firmado electrónicamente por:
JHONNY FERNANDO CARRION RAMIREZ

Cédula: 0705743151

jfcarrionr@unl.edu.ec

Fecha: 27/08/2021

CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA DE PRODUCCIÓN PARCIAL O TOTAL, Y PUBLICACIÓN ELECTRÓNICA DE TEXTO COMPLETO

Yo, JHONNY FERNANDO CARRIÓN RAMÍREZ, declaro ser autor de la tesis titulada: "**Modelo de solución mediante el despliegue de un Contrato Inteligente en la red Blockchain de (EVM) para un módulo de registro de usuario",** como requisito para optar el título de **INGENIERO EN SISTEMAS**; autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Institucional. Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad. La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los diecisiete días del mes de febrero del 2022.

Firma:

Autor: Jhonny Fernando Carrión Ramírez

Cédula: 0705743151

Dirección: Loja, sector la Argelia

Correo electrónico: jfcarrionr@unl.edu.ec

Celular: 0988823681

DATOS COPLEMENTARIOS

Director de Tesis: Ing. Cristian Ramiro Narváez Guillén, Mg.Sc

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

Ing. Oscar Miguel Cumbicus Pineda, Mg. Sc.

Ing. María del Cisne Ruilova

Dedicatoria

El presente trabajo va dedicado a Dios, quien siempre ha sido mi guía en toda mi vida.

A mi madre Marlene Ramírez y a mi padre José Luis Carrión por ser el pilar más importante en el transcurso de mi vida, y demostrándome siempre apoyo incondicional sin importar las adversidades.

A toda mi familia y personas más cercanas que de una u otra manera me apoyaron en momentos difíciles y así salir adelante.

Jhonny Fernando Carrión Ramírez

Agradecimiento

Mi gratitud infinita para Dios quien me ha brindado salud y valor para lograr todos mis propósitos establecidos en mi vida mediante la dedicación y esfuerzo.

A mis padres y a toda mi familia quienes han sido mi inspiración para esforzarme durante toda mi vida y formación académica.

A mi director del presente trabajo de titulación Ing. Cristian Ramiro Narváez Guillén Mg.Sc. por formar parte de este logro alcanzado e impartir sus conocimientos y brindarme el tiempo, dedicación y paciencia en mi persona.

A mi novia por el apoyo incondicional y emocional en todo este proceso de formación académica.

A todos los docentes que impartieron sus conocimientos y formaron parte de mi carrera universitaria.

Jhonny Fernando Carrión Ramírez

Índice

Índice de contenidos

Índice de figuras

Índice de tablas

1. Título.

Modelo de solución mediante el despliegue de un Contrato Inteligente en la red Blockchain de (EVM) para un módulo de registro de usuarios.

2. Resumen.

En el país existe un gran abandono por la implementación de tecnologías nuevas o emergentes que cada día se dan a conocer en el mundo, esto causa que dentro de las empresas e instituciones utilicen tecnologías con más de 5 años de antigüedad y lo corrobora el libro Blanco de la Sociedad de la información y del conocimiento creado por (MINTEL), por ello aplicar la tecnología Blockchain y los contratos inteligentes en busca de mayor seguridad en la información y permitir un rendimiento que facilite el registro de datos determinará un avance muy significativo para aplicar sistemas descentralizados.

En base a lo mencionado anteriormente, en el presente Trabajo de Titulación (TT) se estableció la integración de la Blockchain y los contratos inteligentes de Ethereum al sistema de control de acceso en el módulo de registro de los estudiantes como una alternativa innovadora que permita la optimizar este proceso, debido a que son tecnologías nuevas no existen metodologías aplicables para ello se adaptó un conjunto de pasos basándose en el método experimental. En base a una revisión de literatura sistemática se obtuvo que las herramientas como remix-IDE tiene todas las características necesarias en el entorno de Ethereum, como desarrollar, compilar y desplegar contratos inteligentes, de manera similar, MetaMask la billetera que ayuda al usuario a manejar sus transacciones, así también, el uso de librerías como Web3.js que facilita la interacción con la Blockchain y contratos, además se programó el contrato inteligente en el lenguaje Solidity. Para determinar si existe mayor seguridad se analizó tres parámetros esenciales en la seguridad de la información que son: la confidencialidad, integridad y disponibilidad donde la blockchain abarca de estos puntos con mayor firmeza que las bases de datos tradicionales, pero esta seguridad causa que exista un poco más de demora en cada proceso. Finalmente se aplicó el registro de tres estudiantes a ambos casos para analizar el rendimiento en base a los tiempos de respuestas los cuales fueron de 19,69 segundos y 615 milisegundos en comparación a los tiempos de una Base de datos de 540 milisegundos y 249 milisegundos denotando una diferencia dependiente de la arquitectura de la cadena de bloques.

Palabras claves: contratos inteligentes en control de acceso, blockchain, Ethereum and blockchain, remix IDE, Solidity, Smart Contract and node.js, MetaMask, web3, Ethereum virtual machine (EVM), Ropsten tesnet, etherscan.

Summary

In the country there is a great abandonment for the implementation of new or emerging technologies that every day are released in the world, this causes that within companies and institutions use technologies with more than 5 years old and this is corroborated by the White Paper of the Information and Knowledge Society created by (MINTEL), therefore applying Blockchain technology and smart contracts in search of greater security in the information and allow a performance that facilitates the registration of data will determine a very significant advance to implement decentralized systems.

Based on the above mentioned, in the present Degree Project (TT) the integration of the Blockchain and Ethereum smart contracts to the access control system in the student registration module was established as an innovative alternative that allows the optimization of this process, because they are new technologies there are no applicable methodologies for it was adapted a set of steps based on the experimental method. Based on a systematic literature review it was obtained that tools such as remix-IDE has all the necessary features in the Ethereum environment, such as developing, compiling and deploying smart contracts, similarly, MetaMask the wallet that helps the user to manage their transactions, as well as, the use of libraries such as Web3.js that facilitates the interaction with the Blockchain and contracts, in addition the smart contract was programmed in the Solidity language. To determine whether there is greater security, three essential parameters in information security were analyzed: confidentiality, integrity and availability, where the blockchain covers these points more firmly than traditional databases, but this security causes a little more delay in each process. Finally, the record of three students was applied to both cases to analyze the performance based on the response times which were 19.69 seconds and 615 milliseconds compared to the times of a Database of 540 milliseconds and 249 milliseconds denoting a difference dependent on the architecture of the blockchain.

3. Introducción.

La inmersión de Blockchain como tecnología disruptiva dentro de los sistemas centralizados que actualmente se manejan, llega para convertirlos en sistemas descentralizados, prometiendo innovación en el área financiera y comercial [1]. Vincular esta tecnología a los Smart Labs y el control de acceso para mejorar su integridad eficiencia y seguridad es un avance esencial gracias a la forma de almacenar la información en bloques unidos en forma de cadena, hacerla disponible para todos, y sin la necesidad de la intervención de intermediarios permite facilidad para poder comprobar y rastrear los datos.

Ecuador como un país en desarrollo busca mejorar la seguridad en la información de los sistemas mediante la implementación tecnologías emergentes, siendo así que según Deloitte[2] más del 50% de empresas describen que existe deficiencia en la seguridad en sistemas de información y que el personal no está debidamente capacitado para trabajar con tecnologías emergentes, por ello observando el enorme potencial que existe en la aplicación de Blockchain en sistemas, y con la aparición de Ethereum con el propósito de brindar a los desarrolladores la facilidad de crear aplicaciones descentralizadas que tengan escalabilidad, estandarización, exhaustividad y la interoperabilidad [3], es aquí donde se define a los contratos inteligentes que son protocolos informáticos encargados de verificar la efectividad de una transacción dentro de la Blockchain[1], esto quiere decir, que comprobarán que las instrucciones que sean programadas se cumplan entre dos o más partes para hacer efectiva o no una transacción eliminando la intervención de terceros.

Considerando lo antes mencionado en el presente Trabajo de Titulación (TT) se propone introducir la tecnología Blockchain integrando los contratos inteligentes en una aplicación de control de acceso desarrollada para el Smart Lab de la Universidad Nacional de Loja (UNL), donde se visualizará todas las características y funcionalidades que las tecnologías brindan. En base a esta propuesta se planteó ciertos objetivos los cuales permitirán realizar su implementación, para ello, se recopiló la información sobre los contratos inteligentes mediante la ejecución de una revisión bibliográfica aplicando la metodología de Bárbara Kitchenham [4], como resultado, se precisó las herramientas y componentes para el desarrollo, compilación y despliegue de los contratos inteligentes, destacando a Remix IDE ya que permite programar los contratos en Solidity y ofrece todas las características necesarias para usarlos. La aplicación se realizó dentro del módulo de registro de los estudiantes del sistema, el cual fue adaptado para un correcto funcionamiento en la red de pruebas Ropsten de la blockchain.

Todo este proceso fue desglosado en diversas secciones del presente (TT), que se explican a continuación: en primer lugar, tenemos la Revisión de literatura, en donde se establecieron conceptos generales y procesos sobre las tecnologías a utilizarse, así como el lenguaje de programación. Posterior, se presenta los Materiales y Métodos donde se determinó el tipo de investigación a realizar, los métodos de investigación propuestos y las técnicas aplicadas para la obtención y tratamiento de la información. Luego, se encuentra la sección de Resultados, es aquí donde se encuentra aplicado cada objetivo del presente trabajo de titulación, con sus respectivos resultados. Seguidamente se localiza la sección de Discusión, donde se destacó los resultados obtenidos en relación a la aplicación, comparándolos con el sistema actual y trabajos relacionados. Y para finalizar, las secciones de Conclusiones y Recomendaciones las cuales, contienen lo más relevante de la investigación en base a los resultados obtenidos, así como también, el punto de vista sobre la intervención de las tecnologías antes nombradas y las mejoras que se pueden aplicar.

4. Marco Teórico.

4.1. Cadena de bloques (BLOCKCHAIN)

Al hablar de las cadenas de boques o blockchain (inglés), se define como una tecnología disruptiva que ha demostrado un crecimiento en su uso y en el interés de su aplicación a partir del año 2008, donde surgió la criptomoneda Bitcoin [5]. Principalmente se la conoce como una base de datos distribuida que almacena información dentro de bloques encriptados [6], también una de las descripciones más acertadas es la de comparar con un libro mayor público que contiene todos los procesos y transacciones digitales que se han realizado y compartido entre las partes involucradas [7] o que se ejecutaron alguna vez en la red Blockchain [8] y que van a poder ser verificados en cualquier momento dentro de la red [6].

Una forma alterna de comprender esta cadena de bloques es definirla como un protocolo criptográfico, que trabaja integrando ficheros informáticos de manera cronológica relacionados por identificadores o códigos [7] y a su vez, se la puede entender como una tecnología que crea de un entorno descentralizado, permitiendo que las transacciones y los datos existentes criptográficamente no sean sometidos al control de terceras personas [7], además todas estas transacciones deben ser verificadas de alguna manera alterna, donde intervienen actores desconocidos [9], y así como resultado se entiende que una transacción es un proceso donde existió un cambio de estado en la Blockchain [10].

Como se ha hablado la unidad fundamental o principal componente del Blockchain son los bloques que tiene una estructura como se puede ver en la figura 5.4, donde observamos que se registra varias transacciones realizadas en base a un determinado tiempo dentro de un periodo [11]. Estos bloques se estructurarán de una manera en la que cada vez que se cree un bloque nuevo esta criptográficamente conectado al bloque anterior [12] y lo conforman dos partes [13]:

- Encabezado, este se conforma de metadatos como, la hora que se creó el bloque, un numero de referencia de bloque único y el enlace del bloque anterior.

- El contenido, se estructura por una lista de activos digitales validada, las instrucciones, las transacciones realizadas, montos y las direcciones de las partes que intervienen en esas transacciones.

Podemos observar que el crecimiento de la cadena de bloques se dará a medida que se agreguen nuevos bloques, requiriendo que el almacenamiento aumente conforme se maneje cierta cantidad de datos [14], esto sería una posición muy favorable para este contexto ya

que entre más datos existan más robusta se vuelve la red de la cadena de bloques (blockchain) [11]; esto se debe a que cuando una transacción ingresó a la cadena de bloques, esta no se borrará y se mantendrá ahí [6] y a que cada uno de estos bloques contendrá un hash

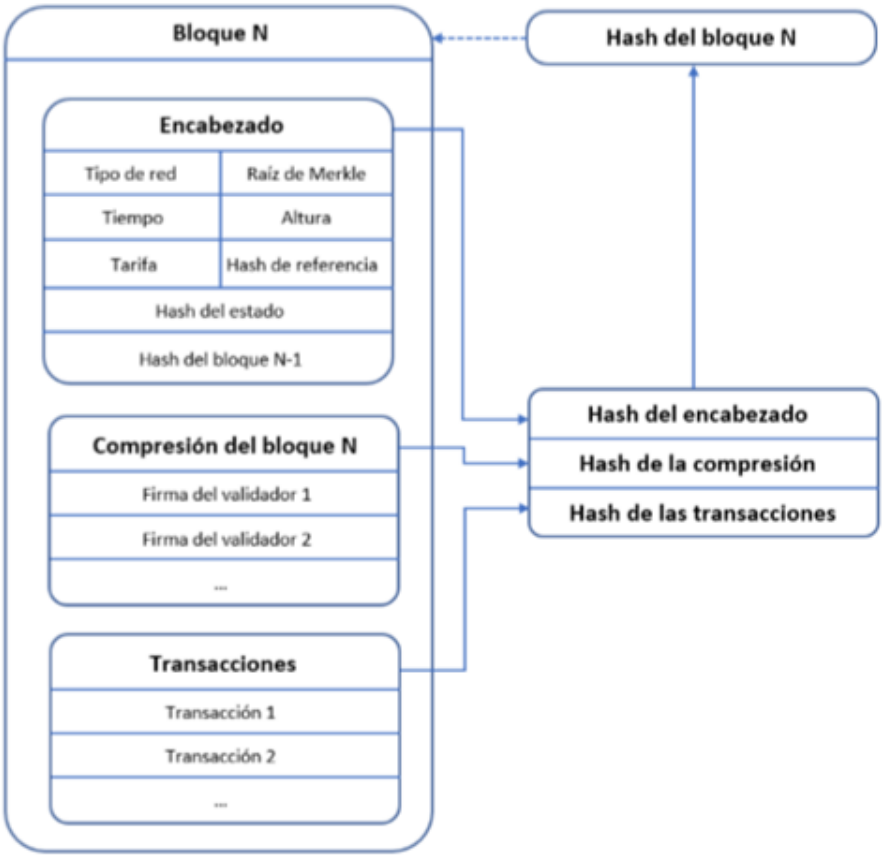


Fig. 1 Estructura de un bloque de la Blockchain [7].

(funciones criptográficas de descifrado) del bloque anterior, entonces si un individuo intenta realizar alguna modificación en el bloque o a la información existente, este deberá acceder a todos los hashes de los bloques anteriores para poder realizarlo [11].

Para consultar información en la cadena de bloques se lo puede realizar de forma sencilla, brindando el acceso a todas las transacciones que se han registrado dentro de ella y se podrían verificar y recopilarla en cualquier momento por cualquier entidad [9], esto lo logramos a ciertas características que presenta el blockchain y que son:

- **Es autónoma:** esto se debe a que la red no es controlada por una única entidad, entonces nos permite la comunicación entre participantes desconocidos y que se pueden ejecutar transacciones más seguras entre si sin que exista un tercero de confianza [8].

- **Es transparente:** nos brinda la visualización de los datos y procesos de actualización de estados, aunque esta información se puede controlar en que cantidad ve un observador, esto se logra regulando a los participantes necesitan tener control de la información [15]. Todos los integrantes que existan en una red pueden acceder a los datos y obtener fallas fácilmente cuando existe un error [11].

- **Es distribuida:** La red blockchain se estructura de forma P2P, entonces cada transacción que este firmada la transmitirá el nodo fuente, luego los pares a las transacciones las verificaran [16].

- **Es íntegra:** asegura que la información o datos que se manejen no pueda tener modificaciones no autorizadas y que los datos sean los verdaderos [15].

- **Es redundante:** esto lo maneja siempre la cadena de bloques siendo una característica importante, donde lo realiza replicando todos los nodos [15].

- **Es inmutable:** contiene un registro en el libro mayor global donde se almacenan todos los bloques y transacciones que han sido validados y son inalterables, esto se debe a que siempre será necesario la verificación de otros nodos y los cambios realizados [16].

- **Es confiable:** en cada nodo existente almacena una copia de la información que exista en la red, entonces la falla de un nodo se mitigaría fácilmente [16].

- **Es publica:** se puede verificar el estado del sistema por cualquiera, y todos pueden observar y comprobar el libro mayor si se realizaron cambios de acuerdo a lo establecido [15].

4.1.1. Algoritmo de consenso

Siempre son necesariamente aplicados los términos de protocolos y algoritmos dentro de una cadena de bloques, pero esto no quiere decir que sean lo mismo. Entonces tenemos de forma simplificada que un protocolo son las reglas establecidas por una cadena de bloques; y su algoritmo, como el mecanismo donde dichas reglas serán aplicadas e implementadas [17].

Para definir a un algoritmo de consenso de forma técnica según [18] lo expone así:

Los llamados algoritmos de consenso son procesos que sirven dentro de un grupo para la toma de decisiones, donde cada integrante existente ayudará a construir y apoyar la decisión que funcione mejor para ellos. Esta es una forma de resolución donde la mayoría será

apoyada por todos los miembros para una decisión que se determine como la más apta, les guste o no. En términos más simples, es un método o forma de tomar decisiones dentro de un grupo.

En la cadena de bloques no existe un nodo central, asegurando en todos los demás nodos distribuidos sean iguales, los nodos no necesitan confiar en otros para garantizar que los registros sean consistentes para ello son indispensables algunos protocolos dentro de la cadena de bloques [7], [14].

Aclarando los algoritmos de consenso son métodos usados para dar equidad e igualdad dentro de la internet. Podemos decir que son la base para el funcionamiento de todas las cadenas de bloques siendo la parte más importante dentro de las plataformas donde se ejecutan, básicamente se enfocan en ejecutar la validación de las transacciones que ingresan a la red blockchain no sean corrompidas, es decir, que no se encuentren errores o inconsistencias en los datos que se puedan producir durante la transmisión, transacción e introducción de cambios no especificados a los datos previamente establecidos durante una transacción, brindando integridad y seguridad de esta información [17].

El reto clave que se trata de lograr es un consenso sobre las transacciones ya establecidas en una red distribuida [9], sin una instancia central, y mucho menos relaciones de confianza mutua entre los participantes [19], es decir que los vinculados en la transacción no es necesario que confíen entre si [9].

4.1.2. Lista de algoritmos de consenso

Según [18] presenta la siguiente lista de algoritmos existentes:

- Prueba trabajo (PoW)
- Prueba participación (PoS)
- Prueba participación delegada (DPoS)
- Prueba participación arrendada (LPoS)
- Prueba tiempo transcurrido (PoET)
- Práctica tolerancia a faltas bizantinas
- Tolerancia a faltas bizantina simplificada
- Tolerancia a faltas bizantina delegada

- Grafo acíclico dirigido
- Proof-of-Activity
- Proof-of-Importance
- Proof-of-Capacity
- Proof-of-Burn
- Proof-of-Weigh

Como se puede visualizar existen una gran cantidad de algoritmos de consenso. De estos los mayormente implementados son PoW y PoS, describiremos algunos de ellos a continuación:

Prueba de trabajo o Proof of Work (PoW): Este algoritmo de consenso fue el primero en ser creado y conoció por ser empleado en Bitcoin y algunas de las otras criptomonedas actualmente existentes. Este algoritmo es considerado parte esencial de todo el proceso de minado dentro de una cadena de bloques. Este algoritmo se encuentra vinculado de forma cercana con las funciones de derivación de claves basadas en contraseña o PBKDF por sus siglas en ingles [20] y se trata de resolver una tarea computacionalmente intensiva para generar un nuevo bloque [9]. Esta tarea trata del cálculo de una función hash criptográfica con cierto grado de dificultad [21], [7].

En [7] habla del proceso de minado PoW donde intervienen numerosas pruebas de hashing, causando cada vez más necesidad computacional esto se traduce en más intentos por segundo. Tratándolo de otra forma los mineros que poseen una tasa de hash demasiado alta tienen más posibilidades de encontrar una solución válida para el siguiente bloque. Este algoritmo de consenso se asegura de que los mineros sean capaces de verificar un nuevo bloque de transacciones y agregarlo al blockchain, pero solo si los nodos distribuidos de la red logran consenso y aceptan el hash block que nos da el minero como la comprobación de trabajo valida.

Prueba de participación o Proof of Stake (PoS): Este algoritmo de PoS fue desarrollado en el 2011 [7] como una opción a PoW. Estos dos algoritmos se relacionan en tener objetivos comunes, pero así mismo existen varias diferencias y particularidades fundamentales. Esto se relaciona exclusivamente a la validación de nuevos bloques.

Según [22]: el algoritmo PoS reemplaza al minado Pow por un mecanismo de bloques ya validados en base al “stake” (cantidad de monedas acumuladas) de los participantes. Existe

el validador de cada bloque, (también conocido como forger o minter en inglés), esto se basa en la inversión en la criptomoneda y no por la cantidad del poder computacional que se asignó. En PoS existen muchas formas de aplicar el algoritmo, pero en la cadena de bloques será asegurada por selección pseudoaleatoria que toma consideración la existencia de capital del nodo y edad de la moneda (tiempo que ha permanecido inmóvil o depositada), junto con un factor de aleatorización.

4.2. Contratos inteligentes.

Los contratos inteligentes en la actualidad tienen una gran cantidad de definiciones y temas en los que abarca su uso, pero generalmente se lo conoce como un acuerdo comercial informático o un pedazo de código programable que se ejecuta automáticamente en una cadena de bloques (Blockchain), para que se lleve a cabo un acuerdo previamente establecido entre las partes que se involucran en una transacción o acuerdo [10]. Como una tecnología emergente, los contratos inteligentes en la actualidad se vienen aplicando en diversas áreas, pero especialmente en las de negocio, pero una parte importante que se vincula a los contratos inteligentes como área de investigación emergente es la seguridad que nace de la ejecución de estos en la cadena de bloques [10], ya que cada contrato inteligente se ejecuta en cada nodo de la cadena.

Fig. 2 Funcionamiento de los Contratos Inteligentes [17].

Los contratos inteligentes se rigen por un protocolo de consenso que verifican la correcta ejecución en la cadena de bloques [23], [24]. Las capacidades de codificar un gran conjunto

de reglas necesarias típicamente en transacciones determinan el lugar de aplicación del contrato.

Para que un contrato sea desplegado es necesario que ambas partes estén de acuerdo a los términos establecidos, ya que una vez en ejecución estos son inmutables, y las cláusulas introducidas por las partes que conforman el contrato serán obligatoriamente respetadas, esto se debe a la naturaleza computacional del sistema [25].

4.2.1 Propiedades dentro de un contrato inteligente:

Algunas de las propiedades según [26] que deben existir para que sea un contrato inteligente son:

Confiable: Los contratos inteligentes se cargan a la cadena de bloques. Esto quiere decir:

- Solo usuarios implicados pueden leerlo debido a que este encriptado, y
- El riesgo de estafa se evita permitiendo la interacción entre personas que no se conocen entre sí.

Determinista: en base a una entrada, el resultado será el mismo debido a que todos los nodos están distribuidos simultáneamente. Eso afirma que el código no debería tener ninguna aleatoriedad.

Seguridad: Los datos o se pueden perder al trabajar en la cadena de bloques pública de Ethereum. Los datos registrados estarán de forma inmutable en esta red. El sistema descentralizado de bloques elimina el riesgo de mal uso de datos, ya que la ejecución es gestionada automáticamente por toda la red, en lugar de una sola persona.

Veraz: Una vez desplegado se conseguirá una dirección única. Antes de usarse, la verificación por ambas partes interesadas es necesaria para observar y verificar el código escrito previamente para una mayor seguridad, existente en los contratos del mundo real.

Existen diferentes cadenas de bloques capaces de ejecutar programas que implementan Contratos Inteligentes.

4.2.2. Partes de un contrato inteligente.

En la parte conceptual, los Contratos Inteligentes se forman de tres partes [27], [25]:

- El código de un programa que se convierte en la expresión de una lógica contractual;

Fig. 3 Seis puntos en el diseño de la anatomía de un contrato inteligente, describe los puntos más relevantes sobre el comportamiento y estructuración de un contrato inteligente [17].

- El conjunto de mensajes que el programa puede recibir, y que representan los acontecimientos que activan el contrato;

- El conjunto de métodos que activan las reacciones previstas por la lógica contractual.

Un contrato inteligente se ejecutará en una cadena de bloques conocida como (blockchain), esta cadena será dependiente de la plataforma en la que se esté manejando sea (Ethereum, Hyperleager, etc.), aquí las transacciones que realiza el contrato se registraran eternamente en un entorno siendo inalterables.

4.2.3. Funcionamiento de un contrato inteligente.

Como hemos hablado los contratos inteligentes, no solo se establecen reglas y condiciones en torno a un acuerdo contractual establecido entre varias participantes sino de igual forma que lo hace un contrato tradicional, también compromete a cumplir esos lineamientos de forma automática [28], así también permiten el intercambio de dinero, propiedades, acciones o cualquier cosa que tenga un valor y esto se realice de una manera transparente y sin conflictos, al tiempo que evita los servicios de un intermediario [29]. Estos pueden activarse por medio del envió de transacciones las cuales cumplen todas las reglas con las que cuenta un contrato [12].

Describiendo una manera general, las transacciones que recibe un contrato inteligente pasan por 3 fases, como se lo muestra en la figura 3:

- **Entradas:** Aquí se especifican la demanda de transacción, el id del contrato, las dependencias que se deban establecer y el estado actual [7].

- **Interprete del contrato:** Esta fase trata con el estado actual en el que se encuentra el libro mayor y el código del contrato. Cuando el intérprete del contrato recibe una solicitud, este valido inmediatamente y después procede a rechazar cualquier solicitud no valida [29]. Este contrato puede, dependiendo de la transacción que ejecuta, leer/escribir datos en su almacenamiento, agregar o debitar dinero de su cuenta, enviar/recibir datos(información) o dinero de usuarios/otros contratos dependiendo de las cláusulas o incluso crear nuevos contratos [30].

- **Salidas:** esto se realizará si la solicitud que se envía es válida entonces se generarán salidas como, actualización de un estado y cualquier otra orden enviada. Al terminar todo el proceso, el intérprete conserva el nuevo estado, una declaración de corrección y las sugerencias que han de ser requeridas para los servicios de consenso. Ese paquete se enviará al servicio de consenso para el compromiso final con la cadena de bloques [7].

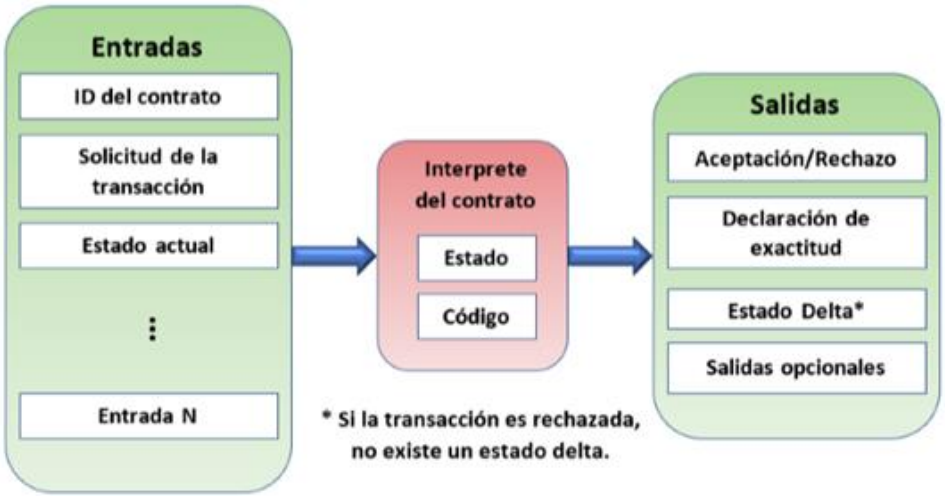


Fig. 4 Fases para la ejecución de un contrato inteligente [7].

4.3. Ethereum

Según el libro blanco de Ethereum [31], lo que trata de realizar Ethereum es fusionar y mejorar todo lo relacionado a scripting, altcoins y meta-protocolos de cadena, permitiendo a los desarrolladores de estas tecnologías emergentes crear aplicaciones arbitrarias basadas en el consenso que se formen de escalabilidad, la estandarización, la integridad de las características, la facilidad de desarrollo y la interoperabilidad que ofrecen estos diferentes paradigmas todo al mismo tiempo.

Ethereum se centra en construir una cadena de bloques e implementar su propio lenguaje de programación completo, que permite a cualquier desarrollador programar contratos

inteligentes y aplicaciones descentralizadas en las que podrá crear sus propios regímenes sobre el manejo de propiedad y dinero, los formatos de transacción y las funciones de transición del estado.

4.3.1. Cuentas Ethereum.

Es uno de los principales componentes de la plataforma Ethereum, las cuales están formadas por 20 bytes, según [31] una cuenta está formada por 4 campos:

- El nonce, un contador utilizado para asegurarse de que cada transacción sólo pueda ser procesada una vez.
- El saldo actual del éter de la cuenta.
- El código de contrato de la cuenta, si está presente.
- El almacenamiento de la cuenta (vacío por defecto).

4.3.2. Éther

Se puede decir que es un token o combustible, que se utiliza para poder realizar el pago de los recursos de las transacciones o aplicaciones que se vayan ejecutar.

Éther es un activo digital perteneciente a Ethereum al igual que Bitcoin, conocido como (criptomoneda), y tal cual el dinero en efectivo, no requiere de la intervención de un tercero que gestione una transacción. Pero este no opera como una moneda de pago, el Éther proporciona “combustible” a las aplicaciones no centralizadas en la red, tomando en cuenta que existe una tarifa por transacción que se realice para que la red procese ese cambio [17].

4.3.3. Gas

Una vez que se ha creado una transacción dentro de Ethereum, a cada una se le carga una determinada cantidad de gas, donde lo que pretende es limitar la cantidad de trabajo que se necesita para ejecutar una transacción y realizar el pago por esta ejecución. Mientras la EVM (Máquina virtual de Ethereum) ejecuta la transacción, el gas se disminuye gradualmente según las reglas preestablecidas [32].

El precio del gas será un valor que se establece por quien cree la transacción, además si existe alguna sobra de este será devuelto a la cuenta patrocinadora.

4.3.4. EVM (Máquina Virtual de Ethereum)

Según [33], describe a la máquina virtual de Ethereum y su funcionamiento de la siguiente manera:

La Máquina Virtual de Ethereum se encarga de la ejecución de contratos inteligentes dentro de su entorno, basándose en brindar seguridad y ejecutar código no confiable en computadoras de todo el mundo. Siendo específicos se centra en prevenir ataques de denegación de servicios, que en la actualidad lo realizan mucho. Otra característica de la EVM es que asegura que los sistemas no tengan alguna vinculación al estado de otro, conservando la comunicación evitando la interferencia.

La EVM se estructuro para servir como un entorno de tiempo de ejecución de contratos inteligentes basados en Ethereum, que pueden estar en un lenguaje que todos puedan entender. Dado que está básicamente aislada de toda la red principal sirve para pruebas de funcionamiento; entonces cualquier empresa que desee crear un contrato inteligente, sin que esto afecte a las operaciones principales de Blockchain.

Probar esta tecnología es de mucha importancia ya que el código defectuoso puede causar la desaparición para los contratos, además el EVM es tratado como “un entorno de aprendizaje” para desarrollar contratos más estructurados, mejores en desempeño y más robustos. También es importante recalcar que cada nodo que existe en la red Ethereum ejecuta una implementación EVM propia y le da la capacidad de ejecutar las instrucciones por igual. La facilidad de información disponible para construir contratos inteligentes adecuados, tanto para expertos como para novatos es extensa en otros países; además el EVM se ha implementado en Python, Ruby, C ++ y algunos otros lenguajes de codificación.

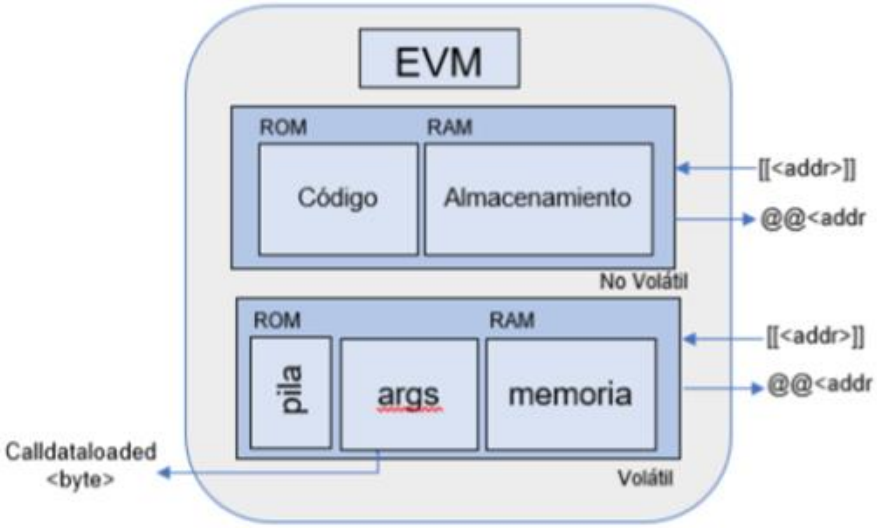


Fig. 5 Arquitectura del EVM [34]

4.3.5. Transacciones

Se determina como transacción a un mensaje que contiene datos o dinero que se envía de una cuenta a otra (que debería ser la misma o la especial cuenta-cero). Puede contener datos como binarios (payload) y Éther. Si la cuenta del usuario destino alberga código, este será ejecutado y el payload se enviara como la información de entrada [35].

Si el destinatario es la cuenta-cero (la cuenta cuyo valor de dirección es 0), se creará un nuevo contrato. Como se dijo, para el contrato no es la dirección cero, sino que se asignara una dirección derivada del emisor y su número de transacciones que fueron enviadas (el "nonce"). Los datos de la transacción serán binarios estos son obtenidos como bytecode por la EVM y ejecutados. La salida de toda la ejecución que se realice es estable y almacenada como el código del contrato. Esto significa que, para crear un contrato, no se envía el código actual del contrato, realmente se envía código que nos devuelve ese código final [32].

4.4. Solidity

Solidity es un lenguaje de programación de contratos inteligentes que tiene un parecido a JavaScript, además algunas semejanzas también con C. Es un lenguaje determinado como estático, sensible a mayúsculas, minúsculas y orientado a objetos (OOP), es decir, permite algunas características limitadas de orientación objetada. Esto quiere decir que los tipos de datos variables deben determinarse y expresarse en tiempo de compilación. Las funciones y variables deben escribirse en OOP de la misma manera que se definen. El código de Solidity está desarrollado en archivos que poseen la extensión .sol. Un contrato escrito en Solidity no es más que una agrupación de código (funciones y variables) que residen en una dirección especifica de la blockchain de Ethereum. Todos los contratos contendran variables de estado, funciones, modificadores, eventos, estructuras y herencias de otros contratos [36].

El concepto de valores "indefinidos" o "nulos" no existe en Solidity, pero las variables recién declaradas siempre tienen un valor predeterminado que depende de su tipo. Para controlar los valores inesperados, debe usar la función revert para revertir toda la transacción o devolver una tupla con un segundo valor que denote el éxito [35].

4.5. Trabajos relacionados

1. Según [37] el trabajo llamado "Contratos electrónicos autoejecutables (smart contract) y pagos con tecnología blockchain", se desarrolló para la ejecución de pagos de forma electrónica los Contratos Inteligentes puedan ser fácilmente acoplados en este caso, aplicando como el asegurador de un pago electrónico, buscando que una vez realizado el pago se pueda cumplir la otra parte de la transacción sin que intervengan

acciones humanas ósea son autoejecutables; se determina en este trabajo que la aparición de los Smart contracts autoejecutables dentro de la búsqueda de eficacia son una necesidad para operaciones en masa, así como en el ámbito electrónico de micro operaciones y la creciente demanda de estructuración de operaciones de IOT. Las dificultades para expandir el conocimiento de los contratos y la normalización causan una incertidumbre al que muchas empresas quieran aplicas estas tecnologías nuevas.

2. Según [17] según su trabajo titulado “Modelo de solución mediante el uso de Smart Contracts para el registro de matrículas de estudiantes en la UCE”, desarrolló mediante la utilización de contratos inteligentes una solución para el registro de las matrículas de estudiantes, brindando la integridad de los datos de los estudiantes con el fin de implementar la descentralización de la información de los estudiantes basados en las cadenas de bloques (Blockchain), de la red de Ethereum debido a que es una de las tecnologías emergentes de gran prospecto en estos usos. La rede que se estableció se ve como un sistema transparente, descentralizado e inmutable, donde las nuevas tecnologías como el Blockchain puede mejorar la eficiencia, la reducción de costos y promover la democratización en la celebración de contratos inteligentes, el uso de estos contratos se puede resumir en 3 palabras: autonomía, seguridad y confianza. El uso de Ethereum para la realización de los contratos inteligentes hay que tomar en cuenta el gas que llevara en vista que mayor robustez del contrato ese necesitara mayor cantidad donde se podrían encontrar métodos o componentes que ayuden a la optimización de este proceso.

3. Según [7] en su trabajo titulado ”Propuesta de una aplicación basada en la tecnología blockchain para el registro de títulos académicos”, donde habla sobre la combinación del uso de contratos inteligentes con la red de cadena de bloques (Blockchain), ofrece una aplicación apropiada en la idea de asignar los títulos académicos a los estudiantes, además que es una solución alternativa e innovadora a las ya conocidas dentro de la cual se explora muchas posibilidades que están fuera del alcance de las aplicaciones que son basadas en estereotipos monolíticos Las herramientas necesarias se las selecciona de acuerdo a las características de desarrollo y el aporte que nos darán en el trabajo, además de que deben ayudar a la interacción y las pruebas de funcionamiento del resultado final. La tecnología Blockchain al principio genero un poco de desconfianza por ser algo nuevo, novedoso y desconocido, pero a medida que se avanza con la investigación se observó el enrome potencial que posee y combinando con la plataforma Ethereum dentro de la EVM permitir la

descentralización de procesos es algo necesario para la seguridad, pero sobre todo para brindar una buena calidad de servicios que es lo que se busca en la actualidad. Al hablar de la cadena de bloques es hablar de revolución tecnológica que ofrece un potencial extraordinario generando un enrome impacto en la educación superior, en este caso mantener un registro de títulos de manera confiable, descentralizada y disponible en el momento que se cumpla todas las restricciones puestas.

4. Según [38] en su trabajo que lo titulado “Análisis de la utilización de la tecnología blockchain para la gestión de la información en sistemas de alarmas residenciales”, analiza la tecnología de cadenas de bloques (Blockchain), la cual tiene uso en muchas implementaciones como una tecnología emergente e innovadora en este caso para un sistema de alarmas residenciales, tomando en cuenta la plataforma Ethereum que brindara la forma de ofrecer el servicio sin contar con un intermediario. La gestión y adquisición de la información ha evolucionado, basarse en sistemas descentralizados y contratos inteligentes ha mejorado las ineficiencias de los actuales servicios centralizados. Al comparar estos servicios centralizados y el descentralizado mediante Ethereum se mejora mucho ofrece mejores características y la seguridad es mucho más eficiente que la que posee un sistema centralizado. En Ecuador se apunta a la innovación e incorpora el desarrollo de nuevas tecnologías, en este caso la cadena de bloques causo la eliminación de la necesidad de depender de un servidor central, ofreciendo la disponibilidad del servicio de 24 horas en caso de una situación de emergencia convirtiendo a los sistemas de alarmas residenciales en el modelo descentralizado una herramienta eficiente y segura.

5. Metodología.

5.1. Tipo de investigación

Para la implementación del presente Trabajo de Titulación se consideró la investigación cualitativa, estableciendo una estructura para la implementación de contratos inteligentes en el registro de estudiantes, esta metodología hace referencia al modo en que se enfocan los problemas y se establecen soluciones, a la manera de realizar la investigación produciendo datos descriptivos [39].

De la misma manera intervino la investigación bibliográfica, considerando la finalidad del cumplimiento de la presente investigación, para recolectar y obtener estudios relacionados al desarrollo de contratos inteligentes en la plataforma Ethereum en conjunto con la tecnología blockchain la cual permitirá realizar el registro de estudiantes en un contrato desplegado en la red de pruebas Ropsten y poder simular un entorno real con la arquitectura ya mencionada.

5.2. Métodos de investigación

Método experimental

Según [40], el método experimental es un conjunto de técnicas que se utilizan para investigar fenómenos, adquirir nuevos conocimientos o corregir e integrar conocimientos previos. Se utiliza en la investigación científica y se basa en la observación sistemática, la toma de medidas, la experimentación, la formulación de pruebas y la modificación de hipótesis.

A través del método científico experimental, los científicos intentan predecir y quizás controlar eventos futuros basados en el conocimiento presente y pasado. También llamado método inductivo, es el más utilizado dentro de la ciencia por los investigadores, siendo esta parte de la metodología científica, se caracteriza por el hecho de que los investigadores pueden controlar deliberadamente las variables para delimitar las relaciones entre ellas.

Teniendo en cuenta estos aspectos, se estableció la siguiente estructura para el desarrollo del presente trabajo:

Herramientas a utilizar.

Se determinaron en base a la contestación de las preguntas de investigación planteadas en la revisión bibliográfica para saber cuáles son las más beneficiosas y con mayor implementación dentro de la plataforma Ethereum, además se tomó en cuenta las que dieron

un gran soporte al desarrollo de la arquitectura en base a las características que las diferencian como las ventajas y desventajas.

A pesar que se incluyeron algunas herramientas que no constan en los resultados de la revisión, se tomaron en cuenta en el objetivo de la selección de herramientas que era poder generar un aporte importante y que optimice el proceso de registro de estudiantes, brindando así al desarrollador una facilidad para realizar las tares con respecto a la creación, compilación, despliegue, etc.

Descripción de Procesos.

En este punto se tiene como objetivo realizar un breve análisis de los requerimientos del contrato inteligente y de los procesos de agregar estudiantes y obtener la información, la idea es establecer el entorno de trabajo adecuado para cumplir con los requerimientos necesarios del sistema.

Desarrollo de un contrato inteligente.

Con las herramientas establecidas se pudo empezar a desarrollar el contrato inteligente, esta fase se centró en el código a programar y en las partes que brindaran la funcionalidad necesaria.

Destacándose los aspectos relacionados a las necesidades del proyecto con el fin de mantenerlo estructurado y ordenado para mejorar la funcionalidad.

Algunos puntos a tomar en cuenta se los describe a continuación:

- Creación de una estructura de datos, la cual facilitara el agrupamiento de la información para poder realizar él envió de forma más fácil.
- Nombrar las variables y funciones de forma en la que se pueda intuir que contienen o que van a realizar lo más claro posible.
- Nombrar el archivo en este caso al contrato, para que referencie a el proceso que está realizando

Despliegue del contrato inteligente.

El propósito de esta fase es describir todos los pasos necesarios para realizar el despliegue del contrato inteligente en la cadena de bloques en este caso de la red Ropsten, además se monitoreó todas las transacciones que se realizaron dentro del mismo.

A continuación, se describen pasos para el despliegue del contrato:

- Realizar la depuración del código la cual comprueba que no existan errores para evitar acciones inesperadas.
- Realizar la compilación del contrato para que se forme en una estructura entendible para la cadena de bloques.
- Desplegar el contrato inteligente, consiste en enviarlo a la cadena de bloques para que esté disponible para la interacción y se ejecute en la máquina virtual de Ethereum.

Interacción con el contrato inteligente.

En esta parte se utilizó las librerías para proceder a instanciar al contrato e interactuar con este mediante el uso de llamadas a las funciones previamente creadas para agregar estudiantes y obtener su información, además se establece el nodo para ejecutar la cadena de bloques y en esta monitorear las generaciones las transacciones que se van adhiriendo. El control se lo realiza mediante el uso de la plataforma Etherscan de la red Ropsten la cual cuenta con una interfaz con información de las transacciones en cada bloque que se minó con su respectivo hash.

Pruebas funcionales del contrato inteligente.

En esta fase el propósito fue ejecutar ciertas pruebas unitarias creadas por defecto y que se realizan dentro del entorno de desarrollo Remix IDE para comprobar que las cuentas tengan saldo y además de que el contrato tenga procesos funcionales, además se realizaran dos pruebas para las funciones en específico de registrar estudiantes y obtener la información, los valores de la prueba se los presentara dentro del entorno de desarrollo.

5.3. Técnicas para la recolección de información

Recopilación documental

Para realizar la revisión bibliográfica se incluyó el uso de la herramienta online Parsifal, debido a que brindó ayuda para planificar, realizar e informar sobre la revisión de manera rápida y precisa. Para lograr su aplicación intervino la metodología de Barbara Kitchenham [4] ya que presentó los procedimientos adecuados para las revisiones sistemáticas que fueron necesarios en el cumplimiento de un objetivo. En la ejecución de la metodología se obtuvieron 18 estudios en relación a nuestro tema de investigación, siendo estos los que brindaron una sustentación científica del tema propuesto.

La metodología de Barbara Kitchenham presentó una gran adaptabilidad de las etapas que comprenden sus tres fases principales, es decir, que permite omitir ciertas fases en caso de no ser necesarias o importantes dentro del cumplimiento de un trabajo investigativo, por lo que en esta investigación se omitieron algunas de ellas, las cuales no eran relevantes al cumplimento de los objetivos.

A continuación, se presentan las fases que se manejan en la revisión.

Planificación de la revisión

- Identificación de la necesidad de la revisión
- Formulación de las preguntas de investigación
- Desarrollo de un protocolo de revisión

Realización de la revisión

- Identificación de la investigación
- Selección de estudios primarios
- Evaluación de la calidad del estudio
- Extracción de datos y monitoreo
- Síntesis de datos

Presentación del informe

- Discusión y análisis

6. Resultados.

En esta sección se presentan los resultados del desarrollo del trabajo titulación los cuales se los detalla a continuación.

Objetivo N°1: Realizar el análisis de los requerimientos necesarios para el desarrollo de un Contrato Inteligente y determinar las herramientas y plataforma a usar.

6.1. Planificación de la Revisión.

Para el análisis de los requerimientos para el desarrollo de un contrato inteligente se procedió con lo siguiente:

6.1.1. Identificación de la necesidad de una revisión.

Dentro de los sistemas actuales el manejo de registro de usuarios y control de acceso para verificar la autenticidad de un usuario para así brindar seguridad es muy importante, según [41] una organización debe brindar seguridad al almacenar, recuperar y acceder a los servicios o datos mediante las tecnologías actuales para estar a la vanguardia.

Mediante el manejo de sistemas seguros y descentralizados aplicando el uso de la tecnología emergente blockchain, se puede ofrecer al propietario de una cuenta, seguridad al registrar y almacenar sus datos privados de conocimiento único, previniendo el robo de información, uso de servicios y el acceso ilegal a los recursos existentes dentro de una cuenta por terceros [42].

Teniendo en cuenta que el registro y control de acceso de los usuarios es importante dentro de un sistema informático, podemos observar la importancia de un análisis para implementar contratos inteligentes, debido a que son parte de una tecnología emergente del blockchain y que son capaces de solventar todo lo expuesto.

En la actualidad la implementación del blockchain en conjunto con los contratos inteligentes para crear sistemas descentralizados, es necesario para evitar los sobre precios en costos, evitar la intervención de terceros y sobre todo que la seguridad de la información pueda ser confiable.

6.1.2. Preguntas de investigación

En relación al objetivo planteado y a poder determinar lo necesario para realizar su cumplimiento, se planteó las siguientes preguntas de investigación.

- ¿Qué estudios discuten acerca de los componentes para el desarrollo de contratos inteligentes en el registro y control de usuarios en EVM?
- ¿Qué estudios presentan herramientas para el desarrollo de contratos inteligentes basados Ethereum para el registro y control de usuarios?

6.1.3. Protocolo de revisión

El protocolo a implementar es el propuesto por Mark Petticrew y Helen Roberts[43], PICOC (Población, Intervención, Comparación, Resultado y Contexto), el cual ayuda a definir la estructura de las cadenas de búsquedas, además con la ayuda de la herramienta Parsifal se puede proceder a la revisión con mayor facilidad.

- Población (P): Smart Contract, Contratos inteligentes
- Intervención (I): Ethereum, EVM
- Comparación (C): no aplica
- Resultados (O): Smart Contract tools, components of smart contracts
- Contexto (C): Blockchain

6.1.3.1. Fuentes bibliográficas.

Previo a realizar búsqueda y selección de los artículos vinculados a la fase 1 del presente proyecto, se procedió a usar bases de datos científicas con acceso gratuito y otra de pago, las cuales brindan una búsqueda más profunda, permitiendo enfocar con mayor facilidad al área de investigación prevista, además la información que brindan estos documentos es de calidad y confiable.

Se obtiene la información de manera virtual y se procede a descargar el articulo para mayor facilidad, las plataformas de las bases de datos que se usaron se las puede visualizar a continuación, (ver TABLA I).

TABLA I

BASES DE DATOS (TABLA PROPIA)

Plataforma	Enlace (URL)
SCOPUS	http://www.scopus.com
IEEEXPLORE	http://ieeexplore.ieee.org
SCIENCE DIRECT	http://www.sciencedirect.com

6.1.3.2. Selección de palabras clave

Para la obtención de palabras clave previo a la búsqueda de los artículos relacionados al tema del proyecto (ver TABLA II), se realizó una revisión de literatura.

TABLA II

ARTÍCULOS DE REVISIÓN PREVIA (TABLA PROPIA)

Título del artículo	Palabras clave
RBAC-SC: Role-Based Access Control Using Smart Contract[41]	Blockchain technology, role-based access control, smart contracts.
A Secure Cloud Storage Framework With Access Control Based on Blockchain[44]	Cloud storage, access control, Ethereum, blockchain, smart contract.
EIDM: A Ethereum-Based Cloud User Identity Management Protocol[45]	Cloud computing, identity management, blockchain, reputation, smart contract.
Smart Contract-Based Access Control for the Internet of Things[42]	Internet of Things, access control, blockchain, smart contract.

Posterior a la revisión de los documentos anteriores y en base al tema planteado, se obtuvieron las siguientes pablaras claves:

- User register
- Access control
- Smart contracts
- Smart contract
- Ethereum
- Blockchain technology

6.1.3.3. Creación de las cadenas de búsqueda

En base a las palabras clave, las preguntas de investigación y al método PICOC se formaron las cadenas de búsqueda en las diferentes bases de datos y se las presenta a continuación (ver TABLA III).

TABLA III

CADENAS DE BÚSQUEDAS (TABLA PROPIA)

Cadenas de búsqueda		
Base de datos	**Código**	**Cadena**
IEEEXPLORE	C01	(((((((("All Metadata":smart contract) OR "All Metadata":contratos inteligentes) OR "All Metadata":SmartContract) OR "All Metadata":smart contracts) AND "All Metadata":login) OR "All Metadata":register users) OR "All Metadata":ethereum) NOT "All Metadata":bitcoin)
SCIENCEDIRECT	C02	(("Smart Contract" OR "Pactos Inteligentes" OR "SmartContract") AND ("ethereum" OR "EVM") AND (register% OR user%) NOT(BITCOIN))
SCOPUS	C03	TITLE-ABS-KEY (("Smart Contract" OR "Pactos Inteligentes" OR "SmartContract") AND ("ethereum" OR "EVM") AND (register% OR user%) AND NOT (bitcoin)) AND (LIMIT-TO (DOCTYPE , "ar")) AND (LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016))

6.1.3.4. Criterios de inclusión.

Para que los estudios sean incluidos se tomarán en cuenta los siguientes criterios:

- Estudios publicados entre el 2016 y 2020.
- El idioma inglés será el más relevante en los documentos y también se aceptará en español.
- En el resumen deberán contener información sobre la aplicación de contratos inteligentes y el uso de herramientas dentro del mismo.
- Artículos que sean clasificados como revistas.

6.1.3.5. Criterios de exclusión

Se procedió a excluir los estudios que no cumplan con las siguientes condiciones:

- Estudios que no cumplan con los criterios anteriores
- Artículos que se clasifican como conferencias.
- Publicaciones que estén relacionadas en mayor parte con criptomonedas y bitcoin.
- Estudios parciales o que estén en desarrollo

6.2. Realización de la revisión.

Los procesos para desarrollar la revisión son los siguientes:

6.2.1. Identificación de investigación

La finalidad de la revisión sistemática de literatura, es responder a las preguntas planteadas a través del análisis de los estudios que se obtengan y contribuyan con información confiable y verídica.

6.2.2. Selección de estudios primarios

En este punto se describe el proceso a seguir sobre la selección de estudios primarios basado en los criterios tanto de inclusión como de exclusión, además en las preguntas de investigación y la evaluación de calidad los cuales permitirán obtener resultados fiables.

6.2.2.1. Estudios seleccionados y rechazados.

Luego de realizar la aplicación de las cadenas de búsqueda en cada base de datos y de la aplicación de los criterios de inclusión y exclusión se presentan los siguientes resultados (ver TABLA IV).

TABLA IV

NÚMEROS DE ARTÍCULOS OBTENIDOS, SELECCIONADOS Y RECHAZADOS

Fuentes Bibliográficas	Total de estudios	Artículos Seleccionados	Artículos rechazados
IEEEXPLORE	87	52	35
SCIENCEDIRECT	51	16	35
SCOPUS	89	58	31
Total	**227**	**126**	**101**

En base a la tabla anterior se puede observar que se seleccionaron un total de 126 artículos y se los representa en la figura: de los cuales el 46% pertenecen a Scopus, el 41,3% a IEEE XPLORE y el 12,7% pertenecen a SCIENCE DIRECT, (ver figura 6).

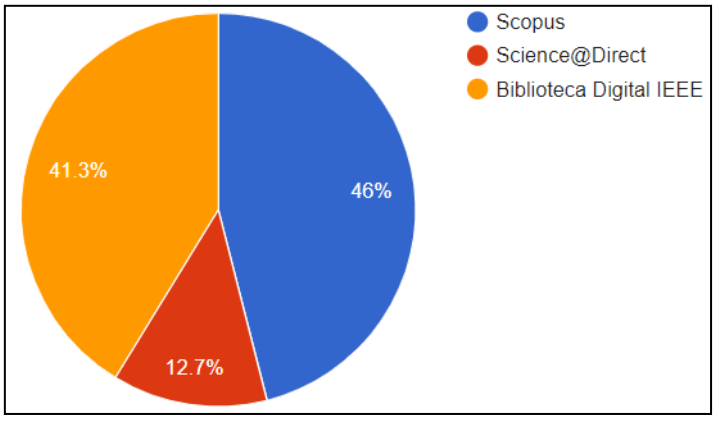


Fig. 6 Porcentaje de estudios seleccionados (Imagen propia)

6.2.3. Evaluación de calidad.

Para definir la calidad de cada artículo se definió las siguientes preguntas:

- **PE1:** ¿El artículo se enfoca en el desarrollo de contratos inteligentes en el módulo de registro de usuarios?
- **PE2:** ¿Describe los componentes del contrato inteligente?
- **PE3:** ¿Se identifica alguna herramienta para el desarrollo de contratos inteligentes?

TABLA V

CALIFICACIÓN DE PREGUNTAS DE CALIDAD (TABLA PROPIA)

Selección	Calificación
Si	1
Parcialmente	0.5
No	0.0

La calificación de cada pregunta se asignó 1 si se selecciona “SI”, 0.5 si se contesta que “Parcialmente” y 0.0 si se calificó con “NO”, además para cada artículo se obtendrá la calificación de 0 a 3, de los cuales se tomarán en cuenta los que tengan mayor o igual a 1,5 (ver TABLA V).

6.2.3.1. Aplicación de las preguntas de calidad

Se selecciono los artículos los cuales obtuvieron una calificación mayor o igual a 1,5 se puede ver los resultados a continuación, (ver TABLA VI).

TABLA VI

APLICACIÓN DE PREGUNTAS DE CALIDAD

Artículo	Preguntas			Calificación
	PE1	PE2	PE3	
D01	0.0	0.5	1	1.5
D02	0.0	0.5	1	1.5
D03	0.0	0.5	1	1.5
D04	0.5	1	0.5	2
D05	1	0.5	1	2.5
D06	1	0.5	0.5	2
D07	1	0.5	1	2.5
D08	1	1	1	3
D09	1	0.5	0.5	2
D10	1	1	0.5	2.5
D11	0.5	0.0	1	1.5
D12	1	1	1	3
D13	1	1	0.5	2.5
D14	0.5	0.0	1	1.5
D15	0.0	1	1	2
D16	1	1	1	3
D17	0.5	1	1	2.5
D18	0.0	1	1	2

6.2.4. Extracción y gestión de datos

Para obtener la selección de los artículos que contienen información en base al tema planteado se utilizó la herramienta Parsifal y a la información más relevante (ver Anexo 1), en este análisis se tendrá en consideración lo siguiente:

- Implementación de contratos inteligentes.
- Herramientas para implementar los contratos inteligentes.
- Conclusiones relevantes.

Luego de la aplicación de la extracción de los documentos relacionados con los parámetros de selección se obtuvo la siguiente cantidad de documentos para el análisis, (ver TABLA VII).

TABLA VII

SELECCIÓN DE ESTUDIOS ACEPTADOS, RECHAZADOS Y REPETIDOS

Fuentes Bibliográficas	Estudios			
	Seleccionados	**Repetido**	**Rechazado**	**Aceptado**
IEEE XPLORE	52	11	33	8
SCIENCE DIRECT	16	0	15	1
SCOPUS	58	5	44	9
Total	126	16	92	18

Luego de realizar el análisis de la tabla 6, se escogió un total de 18 artículos, donde 9 pertenecen a SCOPUS, 8 de estos son de IEEE XPLORE y 1 de SCIENCE DIRECT, (ver Fig. 7).

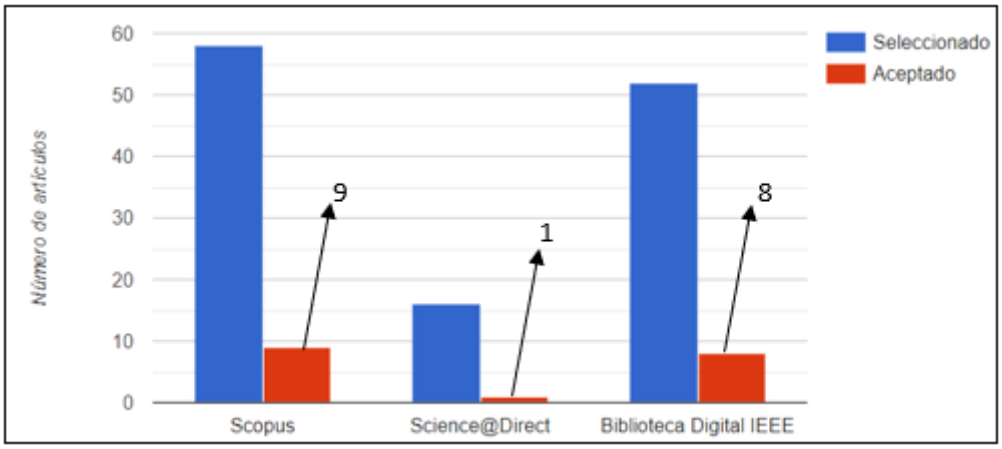


Fig. 7 Número de estudios aceptados (Imagen propia)

Luego de aplicar todos los procesos de selección de los documentos con ayuda de la herramienta Parsifal se obtuvo 18 documentos, (ver TABLA VIII).

TABLA VIII

DOCUMENTOS SELECCIONADOS

Nº	Titulo	Tipo	Año
D01	Combating Deepfake Videos Using Blockchain and Smart Contracts[46].	Jurnals	2019
D02	Blockchain-Based Proof of Delivery of Physical Assets With Single and Multiple Transporters[47].	Jurnals	2018
D03	Proof of Delivery of Digital Assets using Blockchain and Smart Contracts[48].	Jurnals	2018
D04	Smart Contract Based Data Trading Mode Using Blockchain and Machine Learning[49].	Jurnals	2019
D05	A Secure Cloud Storage Framework With Access Control Based on Blockchain[44].	Jurnals	2019
D06	Integrated Application of Blockchain in the Electric Information Management System[50].	Jurnals	2019
D07	Monetization of Services Provided by Public Fog Nodes Using Blockchain and Smart Contracts[51].	Jurnals	2020
D08	EIDM: A Ethereum-Based Cloud User Identity Management Protocol[45].	Jurnals	2019
D09	On the Design of a Flexible Delegation Model for the Internet of Things Using Blockchain[52].	Jurnals	2020
D10	Smart contract-based access control for the internet of things[42].	Jurnals	2019
D11	Using Ethereum blockchain to store and query pharmacogenomics data via smart contracts[53].	Jurnals	2020
D12	IoT Public Fog Nodes Reputation System: A Decentralized Solution Using Ethereum Blockchain[54].	Jurnals	2019
D13	A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems[55].	Jurnals	2018
D14	Smart Contract-Based Review System for an IoT Data Marketplace[56].	Jurnals	2018
D15	Caterpillar: A business process execution engine on the Ethereum blockchain[57].	Jurnals	2019
D16	RBAC-SC: Role-based access control using smart contract[41].	Jurnals	2018
D17	Blockchain for Secure EHRs Sharing of Mobile Cloud Based E-Health Systems[58].	Jurnals	2019
D18	Blockchain-based decentralized reverse bidding in fog computing[59].	Jurnals	2020

Los estudios que fueron aceptados están agrupados por su relevancia en el año de publicación teniendo lo siguiente: en el 2019 se obtuvieron 9 artículos, en el 2020 un total de 4 artículos y en el 2018 se encontraron 5 artículos, en los años 2016 y 2017 no se encontraron artículos que se relacione al tema, (ver Fig. 8).

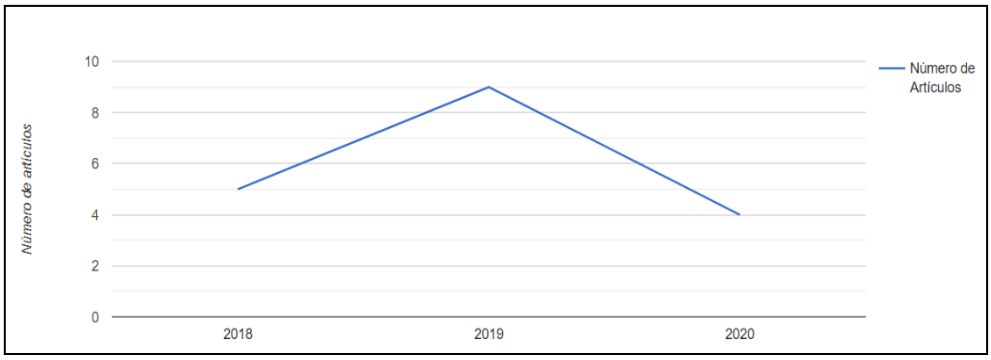


Fig. 8 Número de artículos por años (Imagen propia)

6.2.5. Síntesis de datos

El desarrollo del presente análisis brindara una pauta para el desarrollo del primer objetivo del presente trabajo de titulación, y obtener la información para el desarrollo de contratos inteligentes y las herramientas necesarias para implementarlos dentro de la plataforma Ethereum.

Respuestas a las preguntas planteadas.

6.2.5.1. ¿Qué estudios hablan sobre los componentes para el desarrollo de contratos inteligentes en el registro y control de usuarios en EVM?

En la TABLA IX se puede visualizar la cantidad de documentos que contienen los componentes para el desarrollo de los contrataos inteligentes en Ethereum.

TABLA IX

DOCUMENTOS SELECCIONADOS POR LOS COMPONENTES DE ACUERDO AL SISTEMA

<table>
<tr><th>Tipo</th><th>Cantidad</th><th colspan="3">Componentes</th></tr>
<tr><td rowspan="2">Sistemas en nube</td><td rowspan="2">2</td><td>D05</td><td colspan="2">D17</td></tr>
<tr><td>• Cuentas de Ethereum (msg.sender).
• El hash de transacciones.
• Estructuras en contratos inteligentes.</td><td colspan="2">• Direcciones en blockchain.
• Manejo de funciones.</td></tr>
<tr><td rowspan="2">Sistema en la niebla</td><td rowspan="2">1</td><td colspan="3">D12</td></tr>
<tr><td colspan="3">• Cuentas en Ethereum.
• Despliegue de contratos inteligentes.</td></tr>
<tr><td rowspan="2">Sistemas de información</td><td rowspan="2">3</td><td>D06</td><td>D10</td><td>D16</td></tr>
<tr><td>• Autentificar por métodos (msg.sender).</td><td>• Cuentas de Ethereum.
• Uso del ABI de los contratos.
• Funciones
• Autentificar</td><td>• Funciones en los contratos
• Modificadores
• Creación de roles.</td></tr>
<tr><td rowspan="2">Sistemas de almacenamiento</td><td rowspan="2">1</td><td colspan="3">D13</td></tr>
<tr><td colspan="3">• Transacciones en Ethereum
• id de la transacción
• Dirección del contrato inteligente
• El ABI del contrato inteligente
• Variables especiales: msg.sender, msg.valu, tx.origin.
• Variables.</td></tr>
</table>

6.2.5.2. ¿Qué investigaciones utilizan herramientas para desarrollo de un contrato inteligente en Ethereum para el registro y control de usuarios?

De los documentos analizados se puede observar en la TABLA X la cantidad de estudios que contienen las herramientas para el desarrollo de contratos inteligentes.

TABLA X

DOCUMENTOS SELECCIONADOS POR HERRAMIENTAS

Herramientas	Remix IDE			Ethereum Geth client		Truffle Suite	solc-js compilador estándar de Solidity
Artículos	**Desarrollo**	**Pruebas**	**Depuración**	**Despliegue**	**Interacción**	**Acceso a la red y Pruebas**	**Compilación**
D01	✓	✓	✓				
D02	✓	✓	✓				
D03	✓	✓	✓				
D04	✓	✓	✓				
D05	✓	✓	✓	✓	✓		
D06							
D07	✓	✓	✓				
D08	✓	✓	✓				
D09				✓	✓		✓
D10	✓	✓	✓	✓	✓		
D11						✓	
D12	✓	✓	✓			✓	
D13							
D14	✓	✓	✓				
D15							✓
D16							
D17				✓	✓		
D18	✓	✓	✓				
Cantidad	11			4		2	2

6.3. Discusión de análisis

Una vez realizado el análisis de la revisión sistemática de literatura se puede decir lo siguiente:

- Durante la etapa de revisión se observa que los diferentes artículos analizados presentan diversidad de herramientas para las diferentes etapas como son: desarrollo, despliegue, compilación y las pruebas de los contratos inteligentes dentro de la plataforma de Ethereum la cual mediante un lenguaje de programación le permite crear software para gestionar las transacciones y automatizar resaltados, estas herramientas permiten observar la simulación y el comportamiento en la red blockchain, así mismo como los costos que se efectuaran debido al consumo de gas por transacciones, además se puede destacar que una de las herramientas más usadas que trabaja en base a un navegador es Remix IDE la cual brinda una interfaz e implementa la programación con Solidity que es el lenguaje base y reconocido por Ethereum para el desarrollo de contratos inteligentes, en contraste se puede usar librerías dentro de IDEs Y otros lenguajes para realizar las algunas de las funciones o conexiones a la red blockchain.

- En de los estudios D05, D12, D17 se realiza el análisis y la arquitectura de sistemas que utilizan servicios de la nube o de la niebla (fog) y por medio de Ethereum con la red blockchain intentan que estos sistemas sean descentralizados e independientes en el manejo de los datos, los cuales integran el registro y control de usuarios para validar su información, además describen como actuara el contrato inteligente entre las partes que intervengan y la red blockchain, también se observa que la implementación de esta tecnología es factible dentro de la nube y vinculados a ella.

- Los estudios D06, D10, D16 se explica la estructura de sistemas centralizados de la administración de los roles de los trabajadores, entonces mediante la implementación de contratos inteligentes y blockchain descentralizan el sistema aplicándolos en el registro y control de acceso de usuarios el cual aumenta significativamente la seguridad y la eficiencia al crear los usuarios, además se observa que la intervención de la cadena de bloques permite transparencia y anonimato de los usuarios, cabe mencionar que en el estudio D10 está orientado a Iot y el control de acceso a sus dispositivos mediante contratos inteligentes con una red blockchain (p2p) lo cual da a entender que estas tecnologías tiene una gran inmersión en los diferentes tipos de sistemas para descentralizarlos.

- En el estudio D13 presenta un sistema de almacenamiento en el cual combina la tecnología ABE (Encriptación en Base Atributos) y contratos inteligentes en conjunto de la blockchain para obtener un control de acceso seguro e intercambio descentralizado de datos, también mediante este sistema el propietario será el único que controlara esta información y la cadena de bloques administrara la clave de cada usuario el bajo precio que se realiza por las transacciones y el alto rendimiento determinan una gran intervención de estas tecnologías emergentes como un punto importante para implementar módulos que resguarden los datos de los usuarios que se registran en los sistemas.

- En los estudios D01, DO2, DO3, D04, D05, D07, DO8, D10, D12, D14, D18 presentan una potente herramienta muy utilizada Remix IDE en la actualidad sirve para el desarrollo, depuración, pruebas e implementación de contratos inteligentes con base en el lenguaje Solidity, al tener todas estas capacidades la convierte en ideal para implementarla en sistemas que trabajen con la red blockchain de Ethereum, además los estudios D05, D07, D08, D12, D18 están orientados a sistemas implementados en computación en la nube(cloud computing) y en computación en la niebla(fog computing) que utilizan esta herramienta para el desarrollo de los contratos inteligentes, estos intervienen principalmente para almacenar y recuperar datos, monitorear y medir sus servicios, además la interacción entre nodos de la blockchain forman esquemas de control de acceso distribuidos. Dentro de los estudios D04, D14 hablan sobre el comercio de datos donde debe existir seguridad una parte muy importante, ahí es donde intervienen los contratos inteligentes para verificar al propietario de estos datos y su calidad evitando la intervención de terceros, en base a lo observado, el uso de Remix IDE dentro de la plataforma Ethereum es esencial pese a que se encontró muy poca especificación de la herramienta en los documentos, se desconoce las versiones con las que se trabajó y las especificaciones en las que intervino, aun así la capacidad de Remix es robusta para implementar en todo tipo de sistema e importante para el desarrollo de este trabajo.

- Los estudios DO5, D09, D10 y D17 detallan el uso de Geth client una herramienta basada en lenguaje Go de Google que nos sirve para ejecutar un nodo de Ethereum y así poder realizar transacciones e interacciones con los contratos desplegados en la blockchain, para poder comprobar el funcionamiento se ve necesario crear estos nodos por medio de esta herramienta y comprobar un correcto funcionamiento, además en los estudios D09 y D17 no se especifica sobre el uso de un determinado IDE del desarrollo de los contratos pero se analiza las interacciones con los nodos mediante el Geth, en contraste en los estudios D05 y d10 nombran como IDE la

intervención de Remix que en combinación de la creación de nodos Geth interactúan y comprueban el funcionamiento en la red Ethereum, como se puede observar las facilidades que puede conllevar el uso de esta herramienta la hace muy importante.

- En los estudios D11 y D12 emplean la herramienta Truffle muy conocida e implementada que sirve como Frond-End, además de que abarca el marco de pruebas y canalización de activos de la blockchain, en el documento D12 realizo la combinación del funcionamiento de Remix y Truffle para abarcar el desarrollo y las pruebas dentro de IOT dentro de una arquitectura den la niebla(Fog), con esto se pretende ver la amplitud de herramientas que trabajan en conjunto con Ethereum y definirlas para ser aplicadas en el trabajo, pese a que Truffle tiene muchas capacidades Remix demuestra ser más robusta y abarcar muchas fases que los contratos inteligentes conllevan y eso se demuestra en la cantidad de estudios en la que interviene .
- Los estudios D09 y D15 utilizan una herramienta para realizar la compilación de contratos inteligentes de forma local conocida como Solc que es del lenguaje Solidity, que trabaja mediante línea de comandos evitando el uso de IDEs, aunque presente menos características que Remix su implementación también se lleva a cabo en ambientes locales, pero para casos de presenciar un funcionamiento dentro de Ethereum y la blockchain se necesita una herramienta más completa como es Remix.
- La metodología para implementar contratos inteligentes dentro de la plataforma Ethereum y su red blockchain está definida por su documentación, se trata de un diseño por contrato, la cual toma a los elementos del diseño como participantes de una relación similar al contrato de negocios. Aunque en los estudios acoplan sus sistemas a sus propias metodologías para la arquitectura, la escritura y diseño de los contratos inteligentes está dirigida por esta metodología de diseño por contrato que es la que procederemos a aplicar para programarlo.

6.3.1. Selección de herramientas

En la actualidad existen muchas herramientas que ayudan a implementar contratos inteligentes en la blockchain, pero en base a nuestro previo análisis y consulta de estas se determina el uso de las siguientes herramientas las cuales se pueden describir a continuación:

6.3.1.1. MetaMask

El funcionamiento de MetaMask es gracias al uso de web3.js, una librería que forma parte del desarrollo oficial de Ethereum. web3.js fue creada con el fin de permitir la creación de aplicaciones web que pudieran interactuar con la blockchain de Ethereum. Gracias a ella, páginas web y extensiones pueden aprovechar el poder de Ethereum y sus características. Es decir, MetaMask no solo genera un monedero de criptomonedas, sino que controla cada interacción del usuario con la DApp, y realiza las operaciones necesarias para que dichas operaciones se lleven a cabo[60].

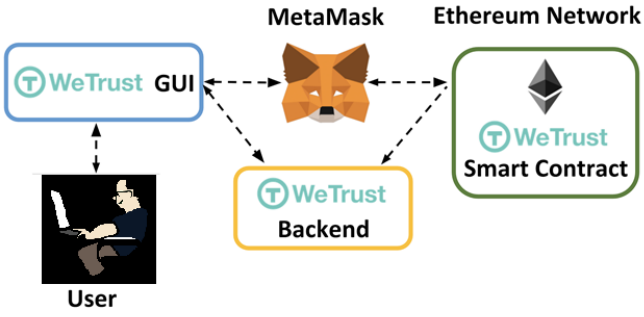


Fig. 9 Proceso de interacción Ethereum-MetaMask [61].

6.3.1.2. Remix IDE

Remix es una potente herramienta de código abierto que le ayuda a escribir contratos de Solidity directamente desde el navegador. Escrito en JavaScript, Remix admite tanto el uso en el navegador como localmente, también admite pruebas, depuración e implementación de contratos inteligentes y mucho más[62].

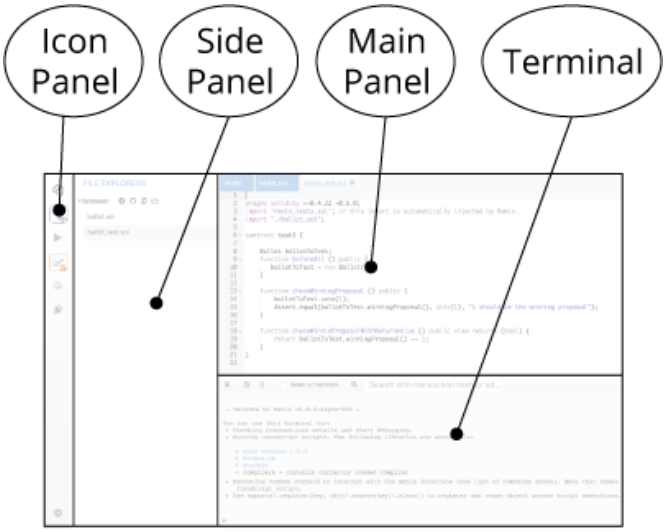


Fig. 10 Estructura de Remix[63].

Según[63] la estructura actual de Remix es la siguiente:

- **Panel de iconos:** haga clic para cambiar qué plugin aparece en el panel lateral
- **Panel lateral:** La mayoría, pero no todos los plugin tendrán su GUI aquí.
- **Panel principal:** En el diseño antiguo esto era sólo para la edición de archivos. En las pestañas pueden ser plugin o archivos para que el IDE se compile.
- **Terminal:** donde verá los resultados de sus interacciones con las GUI. También puede ejecutar scripts aquí.

6.3.1.3. Ropsten Red de Pruebas

Ropsten Ethereum, también conocido como "Ethereum Test net", son como su nombre indica, una red de pruebas que ejecuta el mismo protocolo que Ethereum y se utiliza para probar los propósitos antes de implementar en la red principal (Mainnet). Los ETH Ropsten se utilizan con fines de prueba. Cuando los desarrolladores están creando dApps, o experimentando en la red, para evitar perder dinero pagando ETH real por las tarifas de transacción y las implementaciones de contratos inteligentes, es mejor usar la red Ropsten[64].

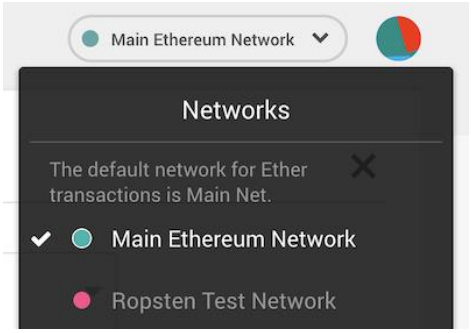


Fig. 11 Redes de prueba Ropsten [65] .

6.3.1.4. Visual Studio

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C ++, C #, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity) [66].

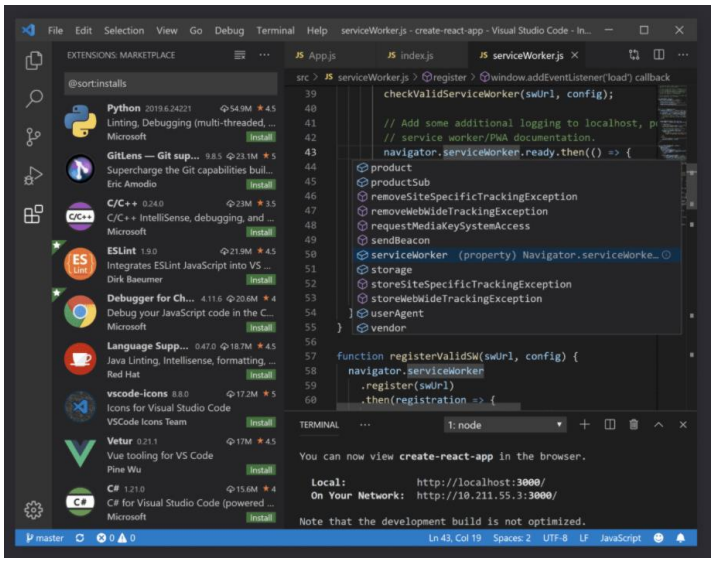


Fig. 12 Editor Visual Studio Code [67].

Objetivo N.º 2: Desplegar el Contrato Inteligente y el módulo de registro de usuarios.

6.4. Plan de Investigación (Smart Lab).

La UNL como una institución de educación superior por medio de sus investigadores y docentes se intenta implementar un Smart Lab para las actividades prácticas y así lograr mejorar las destrezas de los profesores perfeccionando el proceso de enseñanza-aprendizaje de la universidad [68].

La intervención de nuevas tecnologías para lograr estructurar un Smart Lab siempre serán necesarias, en este caso el tema de investigación propuesto se puede adaptar como una opción en el control de acceso de los usuarios, logrando incluir la tecnología Blockchain y Ethereum con sus contratos inteligentes para realizar estos procesos.

El propósito de los “laboratorios inteligentes” en la educación según [69] son:

- Aumentar la cantidad de habilidades aprendidas.
- Mayor logro de objetivos en menos tiempo.
- Disminución de costos de equipo.
- Aumentar el interés de los estudiantes en el proceso educativo.
- Disminuir el tiempo que se requiere para adquirir un cierto nivel de las habilidades de aprendizaje.

6.5. Diseño del módulo de registro de usuarios

En la Fig. 13 se muestra la arquitectura de todo el módulo de registro de estudiantes su proceso con los contratos inteligentes y como se agrega la información en la blockchain, se puede observar todas las herramientas y características necesarias para conseguir todo este proceso.

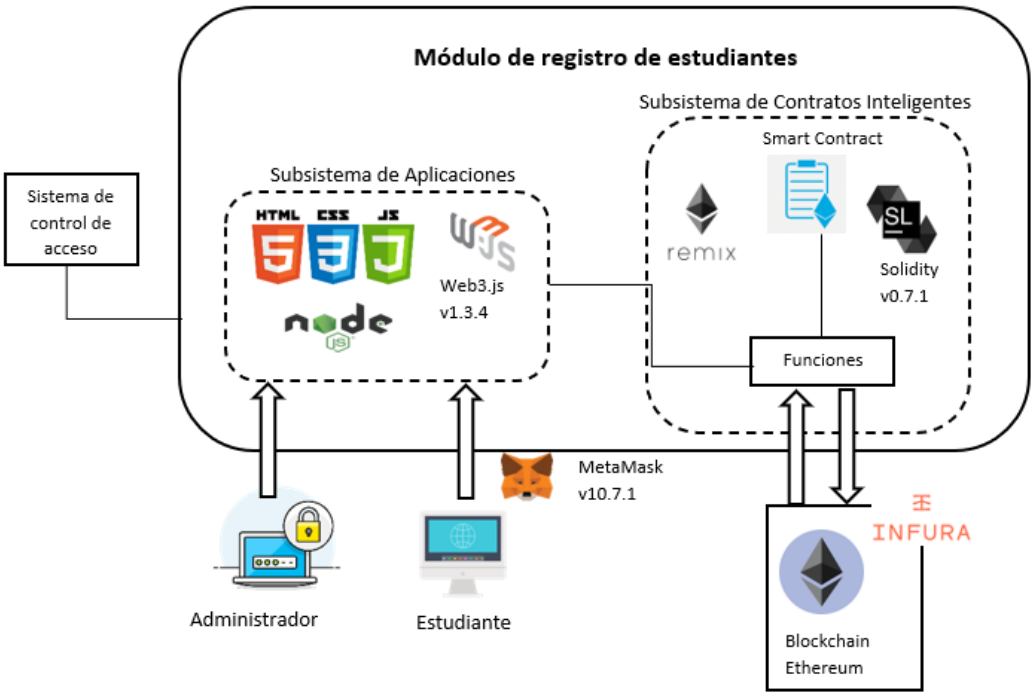


Fig. 13 Arquitectura del módulo de registro de estudiantes en la tesnet Ropsten de Ethereum.

Al observar la Fig. 14, encontramos los escenarios en los cuales el administrador podrá agregar la información del estudiante para guardarla o visualizarla obteniéndola desde la cadena de bloques.

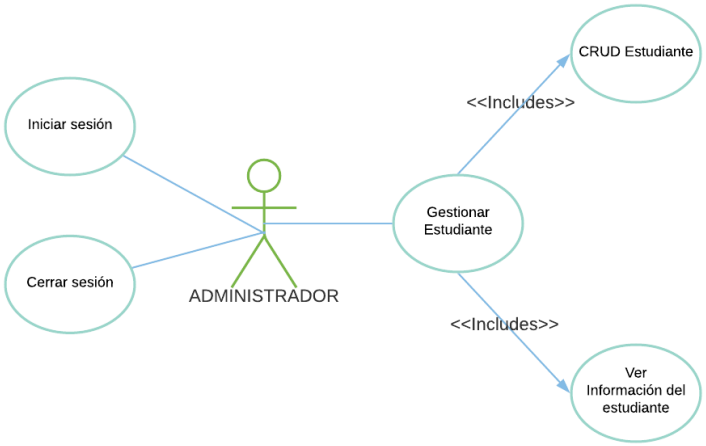


Fig. 14 Escenarios de gestión de los estudiantes

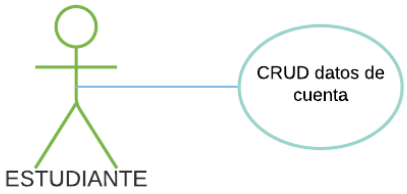


Fig. 15 Escenario de confirmación del estudiante

Luego en la Fig. 15, se encuentra el escenario donde el estudiante llenara los datos de su cuenta para completar el registro realizado por el administrador.

6.5.1. Procesos del funcionamiento de registro y visualización de información

Para agregar a los estudiantes se desplego un formulario con información del estudiante, la cual se puede observar en la Fig. 16, donde abrirá el módulo correspondiente al registro, además se agregó el campo de correo electrónico para realizar la confirmación de la cuenta del estudiante, los datos se almacenarán temporalmente hasta esperar la validación.

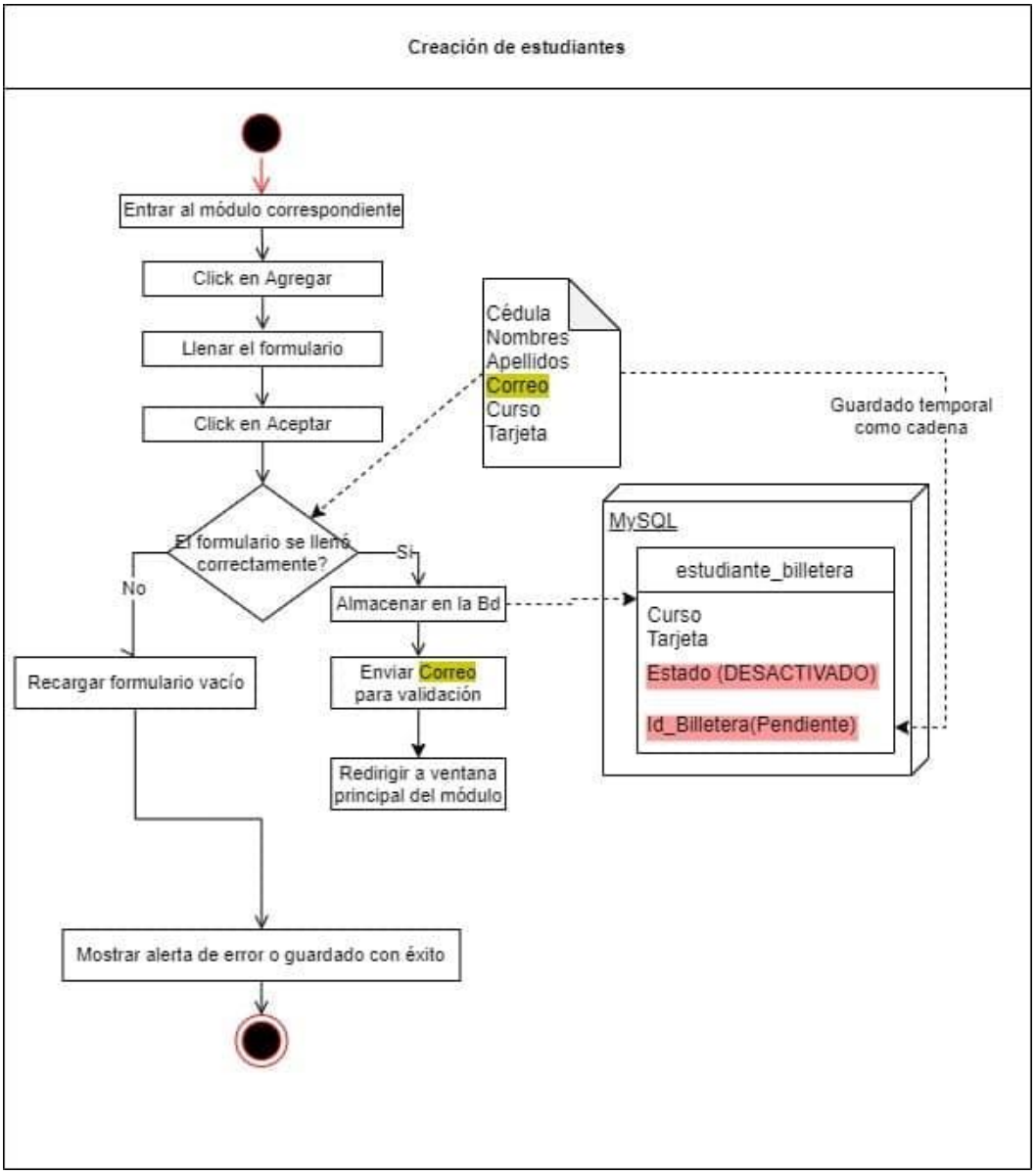


Fig. 16 Flujo de trabajo para añadir un nuevo alumno al contrato inteligente.

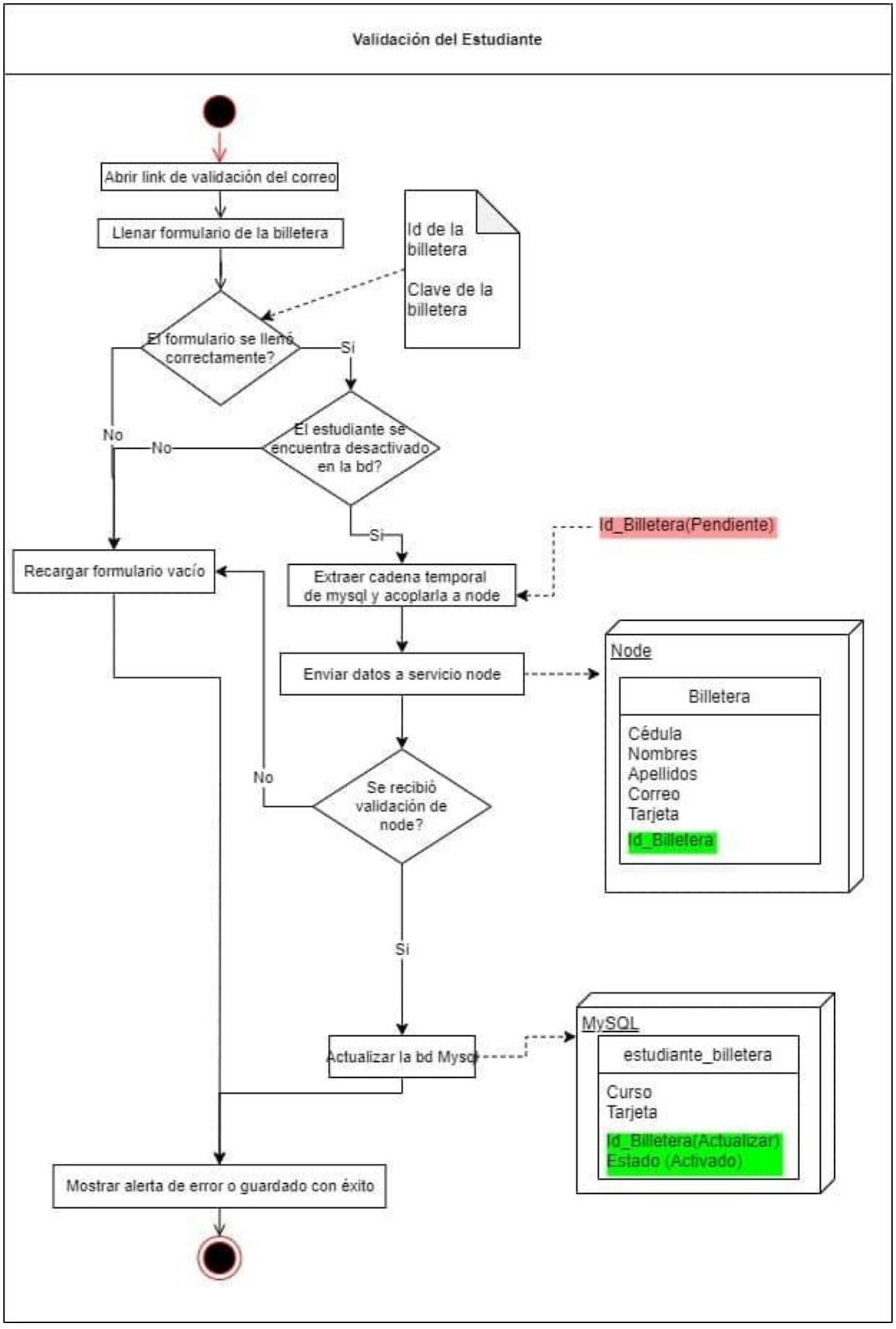


Fig. 17 Flujo de trabajo para confirmar que se añaden nuevos estudiantes al contrato inteligente a través de MetaMask.

Una vez que el administrador realice el registro del estudiante se enviará un correo a la dirección que se especificó en el cual se le direccionará a un módulo de validación (ver Fig.

17), en el cual deberá llenar con el numero de la cuenta de su MetaMask y de su clave privada y proceder a enviarlo para que se pueda guardar en la cadena de bloques.

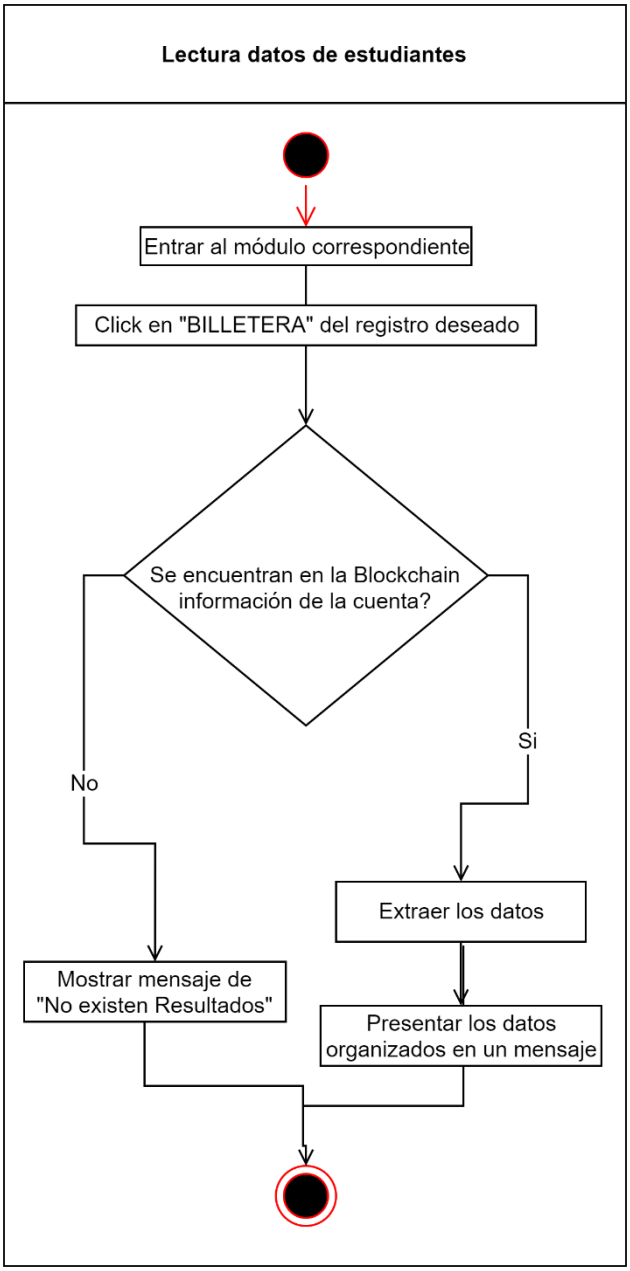


Fig. 18 Flujo de trabajo para la Lectura de datos

Al momento de que la información con los datos respectivos del estudiante sea confirmado y agregado a la blockchain mediante la transacción, se podrá verificar y obtener la información que está en el contrato inteligente dependiendo de la cuenta del estudiante existirá un botón “BILLETERA” el cual al presionarlo realizará una consulta a la red Ropsten y se enviarán los datos de esa cuenta solicitada para proceder a presentarlos (ver Fig. 18).

6.6. Creación y despliegue del contrato Inteligente

Para el desarrollo del contrato se utilizó el lenguaje Solidity que es el mejor y más seguro actualmente para implementar contratos inteligentes y para procederlo a desplegar en la herramienta en línea del editor Remix IDE tomando en cuenta algunos requerimientos.

6.6.1. Características Funcionales y No Funcionales

Según [70] define a los requerimientos de la siguiente manera:

Requerimiento: Un requerimiento es una característica que un sistema debe tener o es una restricción que un sistema debe satisfacer para ser aceptada por el cliente.

Requerimiento Funcional: Describe la interacción entre el sistema y su ambiente independientemente de su implementación, el ambiente incluye al usuario y cualquier otro sistema externo que interactúa con el sistema.

Requerimiento No Funcional: son los requisitos que no se refieren directamente a las funciones específicas suministradas por el sistema (características de usuario), sino a las propiedades del sistema: rendimiento, seguridad, disponibilidad. En palabras más sencillas, no hablan de “lo que” hace el sistema, sino de “cómo” lo hace [71].

6.6.1.1. Requerimientos Funcionales

En la TABLA XI se presentan los requerimientos funcionales que involucran a el registro de estudiantes, la cadena de bloques y el contrato inteligente.

TABLA XI

REQUERIMIENTOS FUNCIONALES

RF-1	Creación del contrato inteligente
Descripción	El contrato será creado por una cuenta que hará la función del administrador y desplegado a la red Ropsten.
Entrada	-
Salida	-
Condición	Conexión con Ethereum (Red Ropsten).
Pre requisito	Proceso de la transacción.
Post requisito	Dirección del contrato.
RF-2	**Agregar el contrato inteligente a la cadena de prueba.**
Descripción	Después de la creación del contrato inteligente se agrega a la cadena de prueba y se replica en los nodos.
Entrada	Dirección de almacenamiento del contrato.
Salida	Código final.

Condición	Conexión con los nodos de Ethereum.
Pre requisito	El contrato y su dirección.
Post requisito	El contrato inteligente se vincula a la cadena de bloques de forma permanente.
Excepciones	Si no existe la dirección del contrato, se notifica.
RF-3	**Registro de Usuario**
Descripción	Para registrar al usuario dentro de la cadena de bloques se necesita obligatoriamente su cedula, nombres, apellidos, correo, curso, número de tarjeta.
Entrada	cedula, nombres, apellidos, correo, curso, número de tarjeta.
Salida	Espera del servicio.
Condición	Conexión con Ethereum.
Pre requisito	Llenar todos los campos de entrada.
Post requisito	Informar del registro.
RF-4	**Envío de datos a registro**
Descripción	Se realiza el registro de los datos o transacción en la cadena de bloques, que está conformada por la red de prueba.
Entrada	Hash de la transacción.
Salida	Mensaje de información del proceso.
Condición	Conexión en la red.
Pre requisito	-
Post requisito	Mensaje informativo del resultado de la transacción.
Excepciones	Si fallo la transacción, se informa.
RF-5	**Finalizar la Transacción**
Descripción	Se registra el contrato en la cadena de prueba y es almacenado en la EVM de Ethereum.
Entrada	-
Salida	-
Condición	Conexión con los nodos de Ethereum.
Pre requisito	Contrato registrado en la cadena.
Post requisito	Disponibilidad del contrato.
Excepciones	No se pueden realizar modificaciones al contrato.

6.6.1.2. Requerimientos No Funcionales

Coste: En Ethereum para ejecutar cada transacción como el registro de un contrato y sincronizar la información requiere un consumo que es el gas, por eso se colocan porciones muy pequeñas de código, ya que es muy costoso.

Escalabilidad: La realización de las transacciones son más lentas en la web3 porque se encuentra descentralizada en la cadena de bloques.

Almacenamiento: En la cadena de bloques al almacenamiento no es económico, así que por esto se añade información solo necesaria para un registro.

Usabilidad: El sistema es desarrollado para el uso de forma sencilla y opciones visibles al usuario, además de una interfaz visual amigable.

Extensibilidad: el funcionamiento del sistema puede ser adaptado a diferentes amplitudes conforme se avance en el desarrollo de la plataforma.

6.6.2. Estructura del contrato

En la Tabla XII se puede observar la estructura del contrato para agregar la información del usuario a registrase.

TABLA XII

ESTRUCTURA DEL CONTRATO

Registro.sol		
Estructuras de datos (Structs)	**Variables (Campos)**	**Función del contrato**
Register	- Cedula - Nombres - Apellidos - Curso - Número de tarjeta - Id de la cuenta - Clave	- se registran a los usuarios por su cuenta. - Añadir usuarios - Solo la cuenta del administrador puede registrar

6.6.3. Funciones del contrato inteligente

En la TABLA XIII se presentan las funciones que se implementan en el contrato inteligente y que ejecutaran el modelo a implementar.

TABLA XIII

FUNCIONES DEL CONTRATO

Registro.sol		
Nombre de la función	**Tipo**	**Explicación**
constructor ()	Principal	Se obtiene la dirección del contrato y la dirección de quien lo creo.
addNewUser ()	Principal	Se registra los nuevos usuarios.
getInfoUser()	Principal	Obtiene la información de la cuenta que se envié como parámetro.

6.6.4. Funcionalidad del contrato inteligente

- Para registrar a los usuarios solamente lo puede hacer el creador del contrato inteligente o quien vendría a ser el administrador. En este caso sería la institución la que los agregara al sistema de forma real.
- Cada usuario tendrá su propia cuenta la que es al que le permite ser diferenciado en la red blockchain.
- Cada transacción será guardada directamente en la red de prueba en este caso Ropsten.

6.6.5. Creación del contrato inteligente

La palabra **Pragma** son instrucciones comunes para los compiladores sobre cómo tratar el código fuente. Posterior a esto se especifica que el código fuente está escrito para Solidity versión 0.7.1, como se puede ver en la Fig. 19. Esto es para asegurarse de que el contrato no se puede compilar con una nueva versión del compilador (de interrupción), donde podría comportarse de forma diferente[35].

Con Solidity que es el lenguaje base para la creación de contratos inteligentes maneja una similitud a todos los lenguajes de programación orientados a objetos, con esto se puede visualizar que para comenzar con el contrato es necesaria la palabra reservada **contract,** seguida del nombre que le demos al contrato en este caso **Registro** como se puede apreciar en la Fig. 20.

Dependiendo del contexto, hay una opción por defecto de donde se almacenará una variable, pero éste puede ser sobrescrito mediante el atributo **memory** o **storage**. El almacenamiento

por defecto para parámetros de funciones (incluyendo parámetros de retorno) es memory, y storage es el lugar por defecto para variables locales. Para variables de estado es forzosamente storage como es evidente ya que el estado de nuestro contrato necesita de esa persistencia[72]. Además, en el caso de los contratos inteligentes también hay que buscar que estas variables sean baratas las que se almacenan en memory son perfectas para aplicar.

Struct es la forma que nos ofrece Solidity para crear nuevos tipos de datos como agregación de tipos ya existentes. Los tipos creados con **struct** pueden ser usados en mappings y arrays (tipos que veremos a continuación en la Fig. 20 y nuestras estructuras pueden contener a su vez mappings, arrays o cualquier otro tipo básico[72].

Las funciones que se pueden usar dentro de Solidity son las que permiten poder interactuar con el compilador y las tareas a hacer.

6.6.5.1. Estructura, variables y funciones del contrato

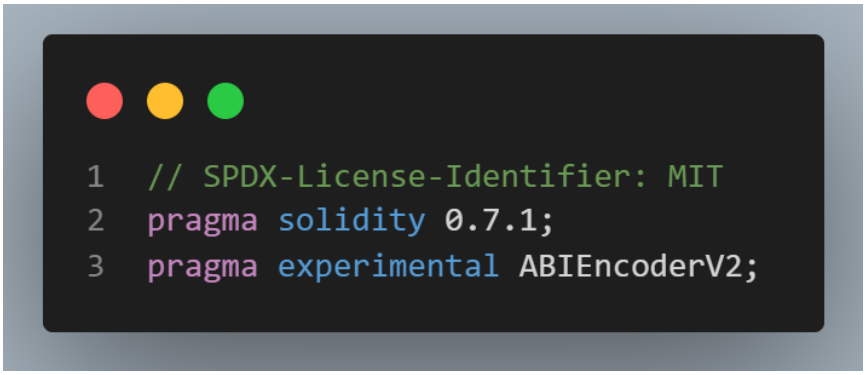


Fig. 19 Identificador de licencia para código abierto y versión del compilador Solidity.

Identificador de licencia SPDX: MIT

La confianza en los contratos inteligentes puede establecerse mejor si su código fuente está disponible. Dado que poner a disposición el código fuente siempre plantea problemas legales en relación con los derechos de autor, el compilador de Solidity fomenta el uso de identificadores de licencia **SPDX** legibles por máquina. Cada archivo fuente debe comenzar con un comentario que indique su licencia[35].

En base a la especificación de licencias el código del contrato es abierto y con la primera línea el compilador no validará que sea una licencia valida dentro de la lista de **SPDX**, pero incluirá la cadena.

Pragma experimental

El segundo pragma es el experimental. Se puede utilizar para activar características del compilador o del lenguaje que aún no están activadas por defecto.

Los siguientes pragmas experimentales están actualmente soportados [35]:

ABIEncoderV2

El nuevo codificador ABI es capaz de codificar y decodificar arrays y structs anidados de forma arbitraria. Puede producir un código menos óptimo y no ha recibido tantas pruebas como el antiguo codificador, pero se considera no experimental a partir de Solidity 0.6.0. Todavía tiene que activarlo explícitamente usando el pragma experimental ABIEncoderV2; se mantuvo el mismo pragma, aunque ya no se considera experimental.

Declaración del contrato y struct

En la Fig. 20 se presenta la estructura que se usó en este contrato inteligente y las variables a emplearse dentro de la misma.

```
1  contract Registro{
2      struct register {
3          uint256 idUser;
4
5          uint cedula;
6          string nombreUser;
7          string apellidoUser;
8          string curso;
9          string Tarjeta;
10         address count;
11         string clave;
12
13     }
```

Fig. 20 Estructura usada dentro del contrato.

Solidity para los contratos inteligentes manejan una estructura similar a las clases de otros lenguajes de programación como ya se mencionó; se definirá la palabra reservada struct

seguida del nombre para identificar esta estructura la cual definirá el tipo de datos y cuáles serán los que se recibirán para que la transacción pueda proceder en caso de no ser así existirá un error.

La estructura de datos empleada llamada **register** contiene 8 campos los cuales con:

- Un id de tipo uint256 (dato numérico de 256 bits)
- Un entero que almacena la cedula
- Un string que almacena los nombres
- Un string que almacena los apellidos
- Un string que guarda el curso
- El número de tarjeta ase almacena en un string
- Una variable de tipo **address** que es propio de solidity para la cuenta de del estudiante.
- La clave será de tipo string

Una **address** es un tipo de datos de 20 bytes. Está diseñado específicamente para contener direcciones de cuentas en Ethereum, que tienen un tamaño de 160 bits o 20 bytes. Puede contener direcciones de cuentas de contrato, así como direcciones de cuentas de propiedad externa [36].

Declaración de variables de estado

Las variables de estado en un contrato inteligente serán los valores que permanecerán almacenados permanentemente o como ya explicamos los datos guardados como **storage,** en caso de existir alguna llamada a una función del contrato donde no sea la instancia definida en estas variables no permitirán el acceso y se revertirá la transacción.

Las variables de estado usadas en este contrato son las que se presentan en la Fig. 21.

Fig. 21 Variables locales

La primera variable que tenemos es de tipo **address** la cual servirá para asignar la dirección de la cuenta de quien despliego el contrato inteligente dentro de la red de prueba **Ropsten**, la variable de tipo **mapping** facilitará la relación de la lista de datos de la estructura **register** con una dirección para proceder a almacenarlos.

Constructor del contrato inteligente

Los constructores son opcionales en Solidity y el compilador induce un constructor por defecto cuando no se define explícitamente un constructor. El constructor en Solidity, se ejecuta un constructor cuando se despliega el EVM. Los constructores deben utilizarse para inicializar las variables de estado y, en general, debe evitarse escribir un código Solidity extenso. El código del constructor es el primer conjunto de código que se ejecuta para un contrato. A diferencia de los constructores de otros lenguajes de programación, en un contrato sólo puede haber un constructor [36].

El constructor del contrato inteligente es aquella función que se ejecutará una sola vez, esto pasa cuando se realiza el despliegue del contrato y la forma de declararlo depende del compilador de Solidity en este caso es la **0.7.1** como se muestra en la Fig. 22.

Fig. 22 Función del constructor

Además se puede observar dentro del constructor la declaración de la variable usuario a la cual se le esta asignando mediante el método **msg,sender** el valor de la dirección de quien creo el contrato.

Funciones Set y Get del contrato

Para poder realizar interacción con el contrato inteligente se lo logra mediante llamadas a las funciones presentes. En la Fig. 23 se puede observar que se inicia por la palabra reservada **function** seguida del nombre que se le dé, posterior a esto se coloca entre paréntesis los datos que admite y especificando el tipo al que pertenecen, además el tipo de almacenamiento y el nombre de cada parámetro. Fuera de los paréntesis se coloca su visibilidad a que pertenece puede ser **public, private, internal o external** en esta parte se puede agregar los modificadores de ser necesarios, y por último se colocan los valores que retornara.

```
1
2     function addNewUser(
3
4         uint _cedula,
5         string memory _nombreUser,
6         string memory _apellidoUser,
7         string memory _curso,
8         string memory _tarjeta,
9         address _count,
10        string memory _clave
11
12        )public {
13
14        id = list.length;
15
16        estudiantes[_count].idUser = id;
17        estudiantes[_count].cedula = _cedula;
18        estudiantes[_count].nombreUser = _nombreUser;
19        estudiantes[_count].apellidoUser = _apellidoUser;
20        estudiantes[_count].curso = _curso;
21        estudiantes[_count].Tarjeta = _tarjeta;
22        estudiantes[_count].count = _count;
23        estudiantes[_count].clave = _clave;
24    }
25
```

Fig. 23 Función para registrar usuarios

La función principal **addNewUser** es publica, esta nos permite almacenar información de la estructura **register**, recibe como parámetros la cedula que es de tipo entero, los nombres, apellidos, el curso, el registro de la tarjeta, y la clave son de tipo string y por último de tipo **address** la cuenta, posterior aplicamos él envió de información al **mapping** estudiantes el cual guardara los datos teniendo como referencia la cuenta registrada.

Fig. 24 Función para obtener información de los estudiantes

La función **getInfoUser** de la Fig. 24 es donde se obtiene la información que se encuentra registrada con la cuenta del estudiante que se desee encontrar, la cual fue previamente fue agregada mediante una transacción.

6.6.6. Instalación de complementos y despliegue del contrato inteligente

Para el despliegue de contratos inteligentes existen varias herramientas que nos ayudan, las que se aplicaron aquí son las determinadas en base a nuestro estudio y para proceder debemos instalar los complementos necesarios en el despliegue dentro de la red **Ropsten** la cual es una blockchain de simulación para los contratos inteligentes, además esta nos permite agregar **Éther** de prueba que es de donde se descontaran para realizar las transacciones.

6.6.6.1. Instalación de complementos para desplegar el contrato inteligente en la red de prueba Ropsten

A continuación, se mostrará el proceso para instalar MetaMask y crear cuentas, además a cada cuenta se le agregará **Éther** de prueba para los costos de ejecutar en la cadena de bloques.

Instalar extensión MetaMask y obtener Éther de prueba para Ropsten

MetaMask está disponible para cualquier navegador como (Chrome, Microsoft Edge, Opera, etc.), es el plugin de Ethereum para interactuar con la red Ropsten de bloques y nos permite agregar **Éther** de prueba.

Al crear nuestra cuenta se otorga una semilla que es un conjunto de 12 palabras se deben guardar, las cuentas den la Blockchain se crean aleatoriamente. En esta se pueden crear tantas cuentas de pruebas como deseemos crearemos 8 en este caso.

Seleccionamos en la parte superior donde mostrara todas las redes que existen y procederemos a escoger Ropsten como en la Fig. 25.

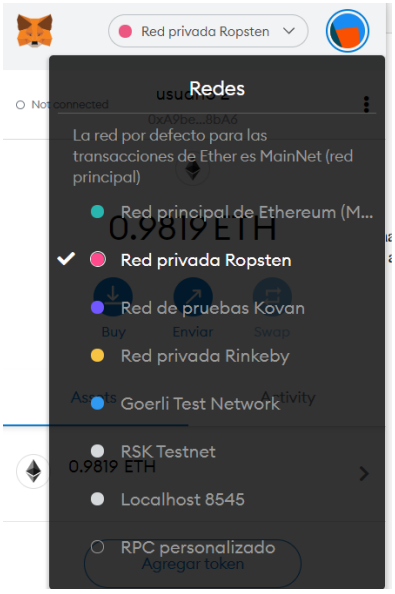


Fig. 25 Selección de red en MetaMask

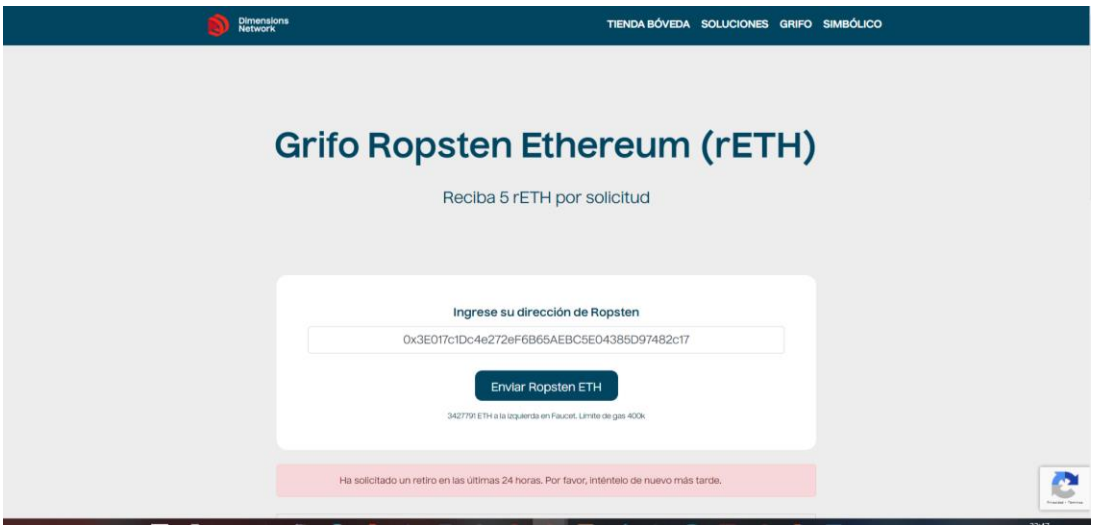


Fig. 26 Obtener Éther

Una vez creadas las cuentas, se procede a obtener **Éther** del sitio de Ropsten (https://faucet.dimensions.network/), para poder pagar nuestra creación de contrato y las transacciones. En la Fig. 26 se muestra cómo se realiza el proceso.

Luego en la Fig. 27 podemos observar algunas de las cuentas con **Éther** ya cargado del grifo para poder realizar las transacciones y pruebas del funcionamiento.

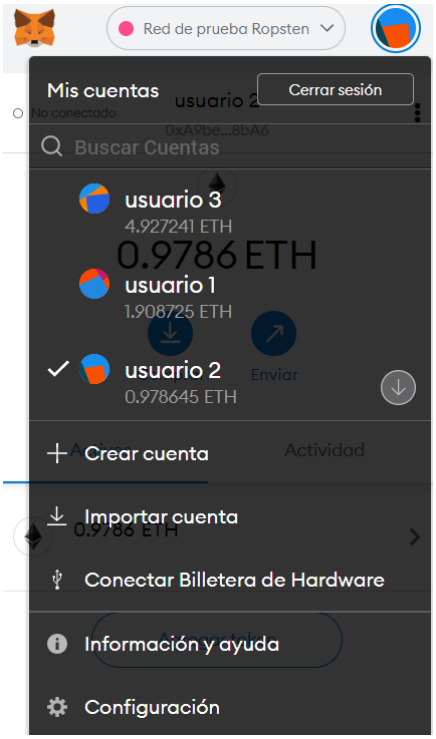


Fig. 27 Cuentas y Éther disponible

6.6.6.2. Despliegue del contrato inteligente en la red blockchain Ropsten con Remix IDE

Una vez que se realizó todo el proceso con MetaMask y la preparación de la red prueba y el contrato inteligente ya programado en Remix IDE se procederá al despliegue del contrato en la blockchain para posteriormente interactuar con este.

Existe un procedimiento para crear e implementar un contrato inteligente, según [73] lo describe de la siguiente manera:

- Programar el contrato inteligente (Code), utilizando un lenguaje de programación para codificar lo que las partes desean hacer.
- Cuanto más complejas son las condiciones, menor es la capacidad del contrato para realizarlas (conforme la tecnología va madurando).

- Luego de que el contrato este escrito, a continuación, es publicarlo en la blockchain sea privada o pública a esto se le conoce como **deploy.** El contrato está encriptado por lo que publicarlo no implica que éste pueda ser leído por terceros, una vez publicado y almacenado en la Blockchain, el contrato puede ser ejecutado (call).

- Una vez que el contrato se encuentra en la blockchain, es ejecutado por los nodos y es necesario que se llegue a un consenso sobre el resultado, según lo definido en el contrato, puede ser necesario actualizar la cadena.

Compilador de Solidity en Remix IDE

Una vez que el contrato se encuentra escrito, se procede a compilarlo, esto gracias a un plugin que admite Remix IDE para Solidity [74], el cual se lo explica a continuación en base a la documentación de la herramienta:

Existen dos formas de compilar el contrato cuando se lo está trabajando, una manera es haciendo clic en el botón compilar y que lleva el nombre del contrato (punto D.) o si desea que el archivo se esté compilando cada vez que se guarda se activaría la casilla de compilación automática (punto E.) de la Fig. 28.

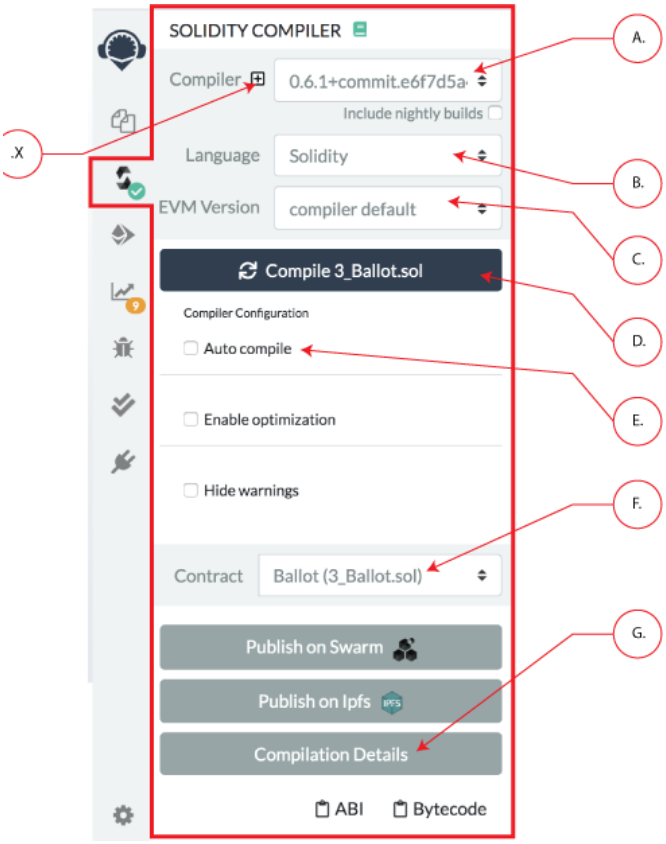


Fig. 28 Compilador de Solidity en Remix IDE [74]

Dependiendo de la versión que estemos utilizando del compilador se puede cambiar de ser el caso (punto A.) y también cuando aquellos que tienen su propio compilador de Solidity personalizado pueden importarlo (punto X.). Aquí mismo se puede seleccionar el lenguaje para la compilación actualmente de Remix existe para Solidity y Yul (punto B.), la versión del EVM de Ethereum por lo general siempre se lo configura por default y selecciona el mejor para el contrato inteligente (punto C.)

Por último, si se tiene varios contratos creados en el (Punto F.) se puede buscar los que están disponibles para compilación y para ver todos los detalles de la compilación que se realice se lo puede visualizar en el (punto G.), además aparecerán como información adicional una vez compilado el ABI y el Bytecode necesarios para establecer una conexión con el contrato desde la interfaz del sistema.

Compilación del contrato Registro.sol

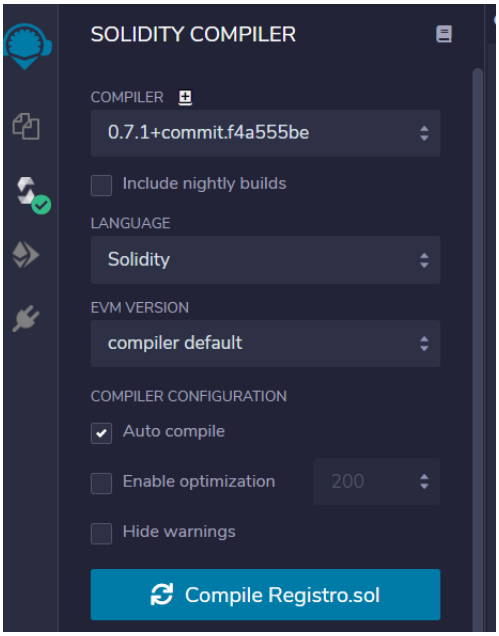


Fig. 29 Configuración del compilador aplicado

En el caso del contrato inteligente que se aplica para el registro de usuarios definimos la versión del compilador en **0.7.1** (ver Fig. 29), y tenemos seleccionado el lenguaje Solidity y activado la compilación automática y como se puede observar que no existe ningún error por el visto en color verde que marca el compilador con esto nos permite proceder a paso del **deploy** del contrato.

Despliegue de contratos en Remix IDE

Para poder acceder a este módulo debe haber previamente compilado un contrato si esto no se ha realizado esta opción no se activará hasta que se agregue uno.

Existen tres formas de desplegar los contratos inteligentes en base a [74] son las siguientes:

- **JavaScript VM:** Todas las transacciones se ejecutarán en una cadena de bloques sandbox en el navegador. Esto significa que no se conservará nada al volver a cargar la página. La JsVM es su propia cadena de bloques y en cada recarga comenzará una nueva cadena de bloques, la antigua no se guardará.

- **Injected Provider:** Remix se conectará a un proveedor web3 inyectado. es un ejemplo de un proveedor que inyecta web3.MetaMask

- **Web3 Provider:** Remix se conectará a un nodo remoto. Deberá proporcionar la dirección URL al proveedor seleccionado: geth, parity o cualquier cliente Ethereum.

También en este módulo contamos con 3 características más antes de desplegar el contrato como se puede observar en la Fig. 30, y según la documentación de Remix [74] las explica a continuación:

Cuenta: La lista de cuentas asociadas al entorno actual (y sus saldos asociados). En la JsVM, usted tiene una opción de 5 cuentas. Si utiliza Injected Web3 con MetaMask, debe cambiar la cuenta en MetaMask.

Límite de gas: Esto establece la cantidad máxima de gas que se permitirá para todas las transacciones creadas en Remix.

Valor: Esto establece el importe de ETH, WEI, GWEI, etc. que se envía a un contrato o a una función pagadera. El valor siempre se restablece a 0 después de cada ejecución de transacción). El campo Valor **NO** es para gas.

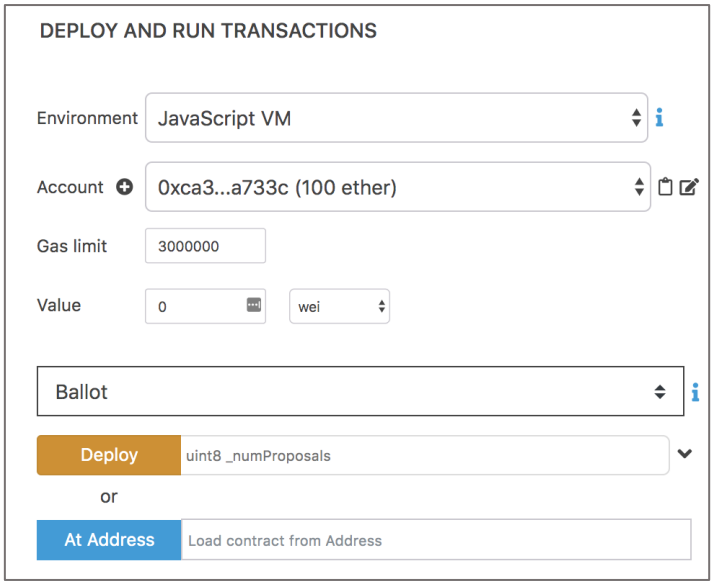


Fig. 30 modulo Deploy de Remix IDE [74]

Despliegue del contrato Registro.sol

Antes del despliegue procedemos a conectar Remix con la cuenta de quién creara el contrato inteligente hospedada en Ropsten como se muestra en la Fig. 31, seleccionamos la opción inyectar Web3 la cual nos mostrara que existen 3 cuentas en la red, en la parte de la cuenta mostrara el número de esta y el saldo que posee y los demás valores quedarán igual para proceder a hacer el deploy.

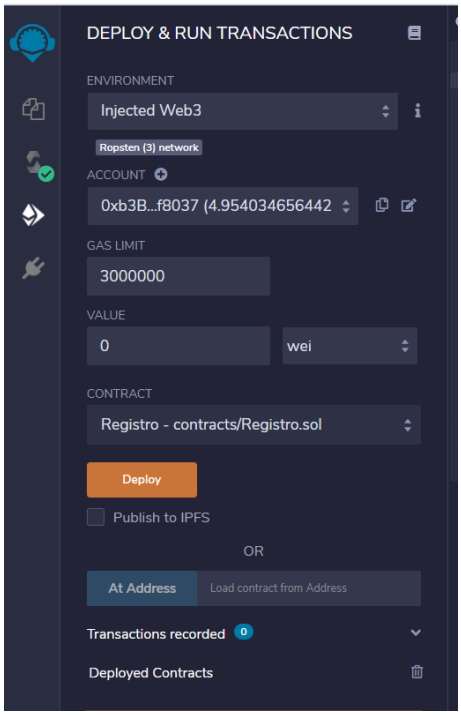


Fig. 31 Configuración para el despliegue del contrato Registro.sol

Al desplegar el contrato se enlazará a la red Ropsten y lo agregará al contrato, asignándole una dirección única, en la Fig. 32 tenemos la salida en Remix del contrato, donde podemos observar las funciones para agregar y obtener la información, aquí se puede interactuar con el contrato, además, para agregar un usuario tenemos todos los campos que se programaron en esta función.

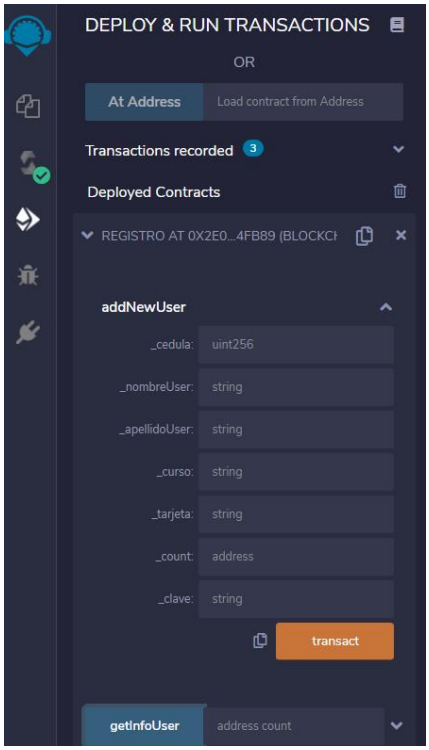


Fig. 32 Funciones del despliegue del contrato en Remix IDE

Así mismo, la consola de Remix tendrá una salida de información donde mostrará el proceso mientras agrega el contrato a la red Ropsten y el resultado con la información de la creación, en caso de ser exitoso como se observa en la Fig. 33.

```
creation of Registro pending...
https://ropsten.etherscan.io/tx/0xfcd551d58162d49e0b2c6b961a26392384ac02b349b2b77f8a2c4a2f9b4f67e2

 [block:10121998 txIndex:9] from: 0xb3B...f8037to: Registro.(constructor)value: 0 weidata: 0x608...10033logs: 0hash: 0xfcd...f67e

 status true Transaction mined and execution succeed

 transaction hash       0xfcd551d58162d49e0b2c6b961a26392384ac02b349b2b77f8a2c4a2f9b4f67e2

 from   0xb3B2b5cfBE9DFfDfA5F768CD47D11077d9df8037

 to     Registro.(constructor)

 gas    715179 gas

 transaction cost       715179 gas

 hash   0xfcd551d58162d49e0b2c6b961a26392384ac02b349b2b77f8a2c4a2f9b4f67e2

 input  0x608...10033

 decoded input  {}

 decoded output  -

 logs   []

 value  0 wei
```

Fig. 33 Salida de consola del despliegue exitoso del contrato inteligente

Una vez completo el proceso en consola, nos brindará directamente un link a la blockchain de Ropsten para la introducción del contrato, conjuntamente se va observando cómo se van minando las transacciones, la hora que fueron agregadas el hash de la transacción, el estado, el número de bloque en el que se agregó, quien realizó la transacción, a quien fue dirigida la transacción, los valores de costo, el precio del gas y la información codificada. Se puede acceder ingresando directamente o también colocando la dirección del contrato en https://ropsten.etherscan.io (ver Fig. 34).

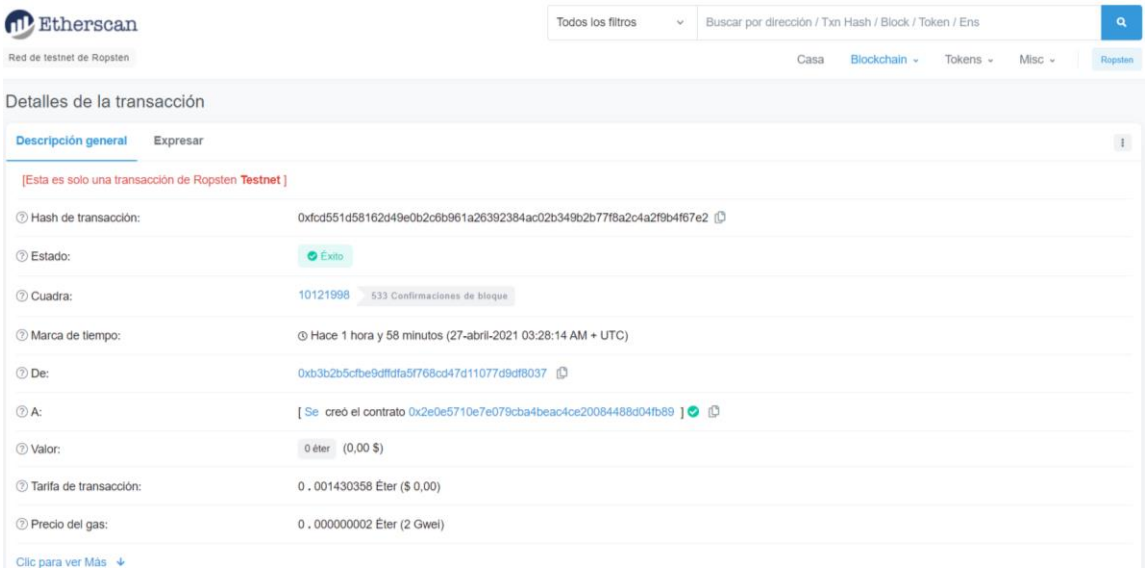


Fig. 34 Vista de la transacción de creación del contrato en la red Ropsten con Etherscan

Objetivo N.º 3: Elaborar un entorno de Pruebas e interacción para medir la usabilidad y el funcionamiento del Contrato Inteligente dentro del ambiente local (EVM).

6.7. Preparación del ambiente de pruebas

Posterior al despliegue del contrato inteligente el ambiente de pruebas será la red de pruebas Ropsten alojada en **Etherscan,** la cual permite visualizar de manera general como se procesan las transacciones en el contrato inteligente, además se muestra la dirección del contrato ("0x2e0E5710E7e079cBa4bEac4Ce20084488d04Fb89") con la cual podremos acceder al mismo, un balance de cuanto Éther se ha manejado, también se puede observar la cuenta que creo el contrato, y en la parte inferior un listado de las transacciones que se van agregando especificando cual fue la razón de esta y otros valores más sobresalientes (ver Fig. 35).

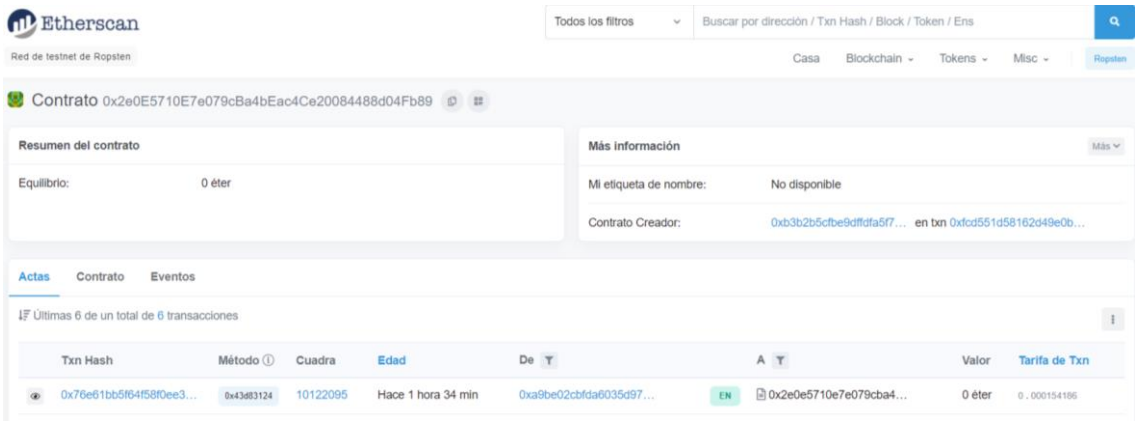


Fig. 35 Visualización del listado de transacciones del contrato en la red Ropsten con Etherscan

6.7.1. Conexión del microservicio con Ethereum

Node.js

Node.js es un entorno de tiempo de ejecución de JavaScript (de ahí su terminación en .js haciendo alusión al lenguaje JavaScript). Este entorno de tiempo de ejecución en tiempo real incluye todo lo que se necesita para ejecutar un programa escrito en JavaScript. Node.js utiliza un modelo de entrada y salida sin bloqueo controlado por eventos que lo hace ligero y eficiente (con entrada nos referimos a solicitudes y con salida a respuestas). Puede referirse a cualquier operación, desde leer o escribir archivos de cualquier tipo hasta hacer una solicitud HTTP [75].

Web3.js

Web3.js es una colección de bibliotecas que le permiten interactuar con un nodo Ethereum local o remoto mediante HTTP, IPC o WebSocket [76].

Web3, en el contexto de Ethereum, se refiere a aplicaciones descentralizadas que se ejecutan en blockchain. Estas son aplicaciones que permiten que cualquier persona participe sin monetizar sus datos personales [77].

Los beneficios que ofrece el uso de web3 en los proyectos según [77] son:

- Cualquiera que esté en la red tiene permiso para usar el servicio, o en otras palabras, no se requiere permiso.
- Nadie puede bloquearlo o negarle el acceso al servicio.
- Los pagos se integran a través del token nativo, Éther (ETH).
- Ethereum es turing-complete, lo que significa que puedes programar prácticamente cualquier cosa.

6.7.2. Conexión con el proveedor web3 y la Wallet

Infura

Infura es una plataforma que proporciona un conjunto de herramientas e infraestructuras que permiten a los desarrolladores llevar fácilmente su aplicación blockchain de la prueba, a la implementación a escala, con acceso simple y confiable a Ethereum e IPFS [78].

Se especificará el uso de Web3 dentro del proyecto mediante su librería y se establecerá una conexión con infura el cual será el proveedor del nodo para interactuar en la red Ethereum usando web3 (ver Fig. 36); el método **HttpProvider()** perteneciente a web3 permite establecer un proveedor de conexión y un nodo para poder interactuar con contratos inteligentes en la blockchain..

```
1  const Web3 = require("web3");
2  let web3 = new Web3(new Web3.providers.HttpProvider('https://ropsten.infura.io/v3/6cb59ec50ca34dc4b40a60e63e62b5b0'));
3
```

Fig. 36 Uso de web3 y infura como proveedor

Una vez que se define el proveedor para el nodo de web3, se procede a guardar la dirección del contrato inteligente que fue desplegado esta será única y el ABI resultante de la compilación que serán necesarios para poder acceder al contrato y poder registrar la

información de los estudiantes (ver Fig. 37); el ABI se lo puede obtener directamente de Remix IDE el cual permite copiarlo para facilitar el uso de contratos inteligentes.

ABI

La Interfaz Binaria de Aplicación de Contratos (ABI) es la forma estándar de interactuar con los contratos en el ecosistema Ethereum, tanto desde fuera de la cadena de bloques como para la interacción entre contratos. Los datos se codifican según su tipo, como se describe en esta especificación. La codificación no es autodescriptiva y, por lo tanto, requiere un esquema para descodificarla [35].

```
let contractAddress = "0x2e0E5710E7e079cBa4bEac4Ce20084488d04Fb89";
  const abi = [
    {
      "inputs": [],
      "stateMutability": "nonpayable",
      "type": "constructor"
    },
    {
      "inputs": [
        {
          "internalType": "uint256",
          "name": "_cedula",
          "type": "uint256"
        },
        {
          "internalType": "string",
          "name": "_nombreUser",
          "type": "string"
        },
        {
```

Fig. 37 Declaración de la dirección del contrato y el ABI.

Y para poder establecer la conexión con la cuenta del estudiante que se va a registrar activamos la Wallet en este caso MetaMask enviando la clave privada y el numero de la cuenta que se confirmaran para el registro (ver Fig. 38); el método **add()** permite que se active una cuenta en la cual se va realizar una transacción recibiendo de parámetro los campos que llene el estudiante en el correo de confirmación.

Fig. 38 Activación de la billetera.

6.8. Interacción con el contrato inteligente

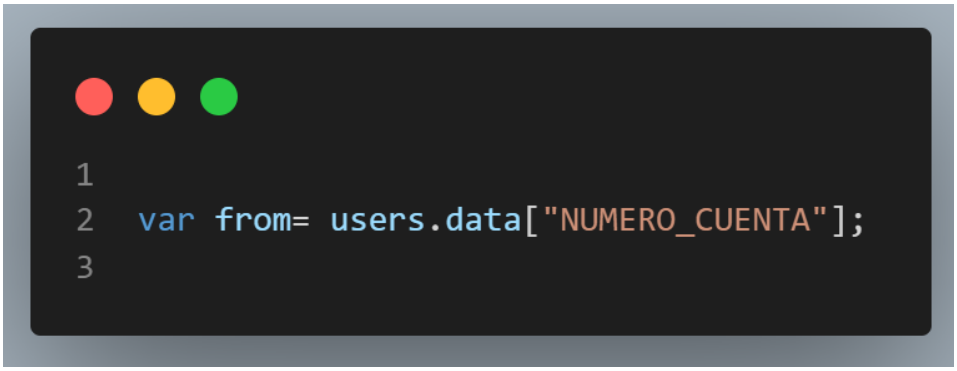
Para la interacción con el contrato inteligente se especificó el proceso de cómo se registrará los estudiantes en la cadena de bloques, luego se mostrará el proceso de la transacción que se realizara al momento de agregar información y posterior poder observar si esta se guardó correctamente.

6.8.1. Creación de objetos para el llamado de funciones del contrato inteligente

Para poder realizar la interacción entre el contrato inteligente, el nodo y la aplicación web se declara una variable **myContract**. Dentro de esta se usará el método **web3.eth.Contract**, el cual se encarga de realizar una instancia del contrato desplegado, usando los datos que declaramos en las variables anteriores del abi y el contractAddress la declaración se puede observar en la Fig. 39.

Fig. 39 Instancia del contrato

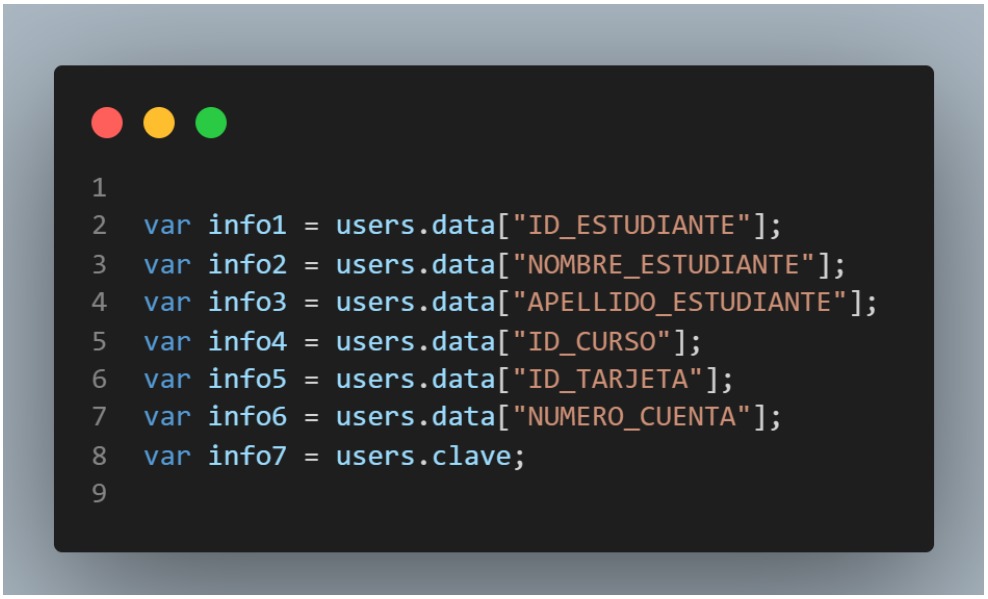
Para poder realizar la transacción se necesita de alguna dirección de una cuenta de la billetera entonces se obtendrá el número de la cuenta que fue confirmada mediante el correo electrónico de la data que se envió al servidor node.js y eso lo guardamos en una variable para completar la información al llamar las funciones del contrato inteligente (ver Fig. 40).



```
1
2  var from= users.data["NUMERO_CUENTA"];
3
```

Fig. 40 Número de cuenta a registrar.

Para poder proceder al llamado de las funciones que se crearon en el contrato inteligente se necesita obtener todos los datos del estudiante que se registrarán en la blockchain y guardarlos en variables para poder realizar el envío mediante la transacción, **users.data** es donde se obtiene toda esta información y la vamos separando en cada variable como se puede ver en la Fig. 41.



```
1
2  var info1 = users.data["ID_ESTUDIANTE"];
3  var info2 = users.data["NOMBRE_ESTUDIANTE"];
4  var info3 = users.data["APELLIDO_ESTUDIANTE"];
5  var info4 = users.data["ID_CURSO"];
6  var info5 = users.data["ID_TARJETA"];
7  var info6 = users.data["NUMERO_CUENTA"];
8  var info7 = users.clave;
9
```

Fig. 41 Variables para guardar la información del formulario

6.8.2. Pruebas de funcionamiento del contrato inteligente

6.8.2.1. Pruebas unitarias a las funciones del contrato inteligente

Para realizar las pruebas unitarias a los contratos inteligentes Remix IDE cuenta con un plugin el cual facilita ciertas pruebas que son básicas para el funcionamiento de los contratos, dentro del desarrollo del contrato para registro de estudiantes tenemos 2 funciones que son las de agregar los datos y obtenerlos a estas se les ejecuto dos pruebas unitarias el resultado de la ejecución en el entorno de desarrollo se muestra en la Fig. 42.

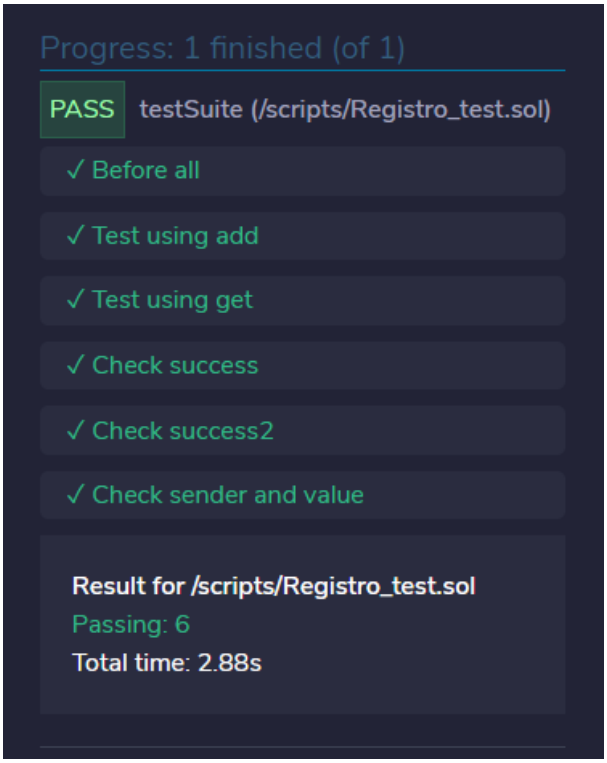


Fig. 42 Resultado de ejecutar las pruebas unitarias del contrato Registro.sol en Remix IDE

El entorno de desarrollo crea un archivo derivado del nombre del contrato que se va realizar el test, además en el informe presenta el estado de las pruebas que pasaron y el tiempo que se tomó ejecutarlas. La segunda y tercera prueba fueron agregadas para verificar la funcionalidad de esos procesos del contrato enviando información previamente agregada a las funciones.

6.8.2.2. Llamada de las funciones del contrato inteligente

Para realizar las llamadas a las funciones del contrato inteligente se debe tener la instancia del mismo, la cual ya la declaramos con la dirección y el ABI del contrato.

Una vez que tenemos acceso al contrato podemos enviar información mediante las funciones para este caso se realizarán las llamadas en orden, primero para agregar la información del estudiante y luego obtendremos esa información de la blockchain.

Fig. 43 Salida por consola en Visual Studio Code al llamar la función de registro de estudiantes (estudiante 1), (addNewUser).

Una vez que el estudiante confirmó su cuenta mediante el correo enviado la transacción se realiza enviando la información a la blockchain y retorna la dirección del bloque en la que fue minada y además de otros valores e información como se puede ver en la Fig. 43 y 44.

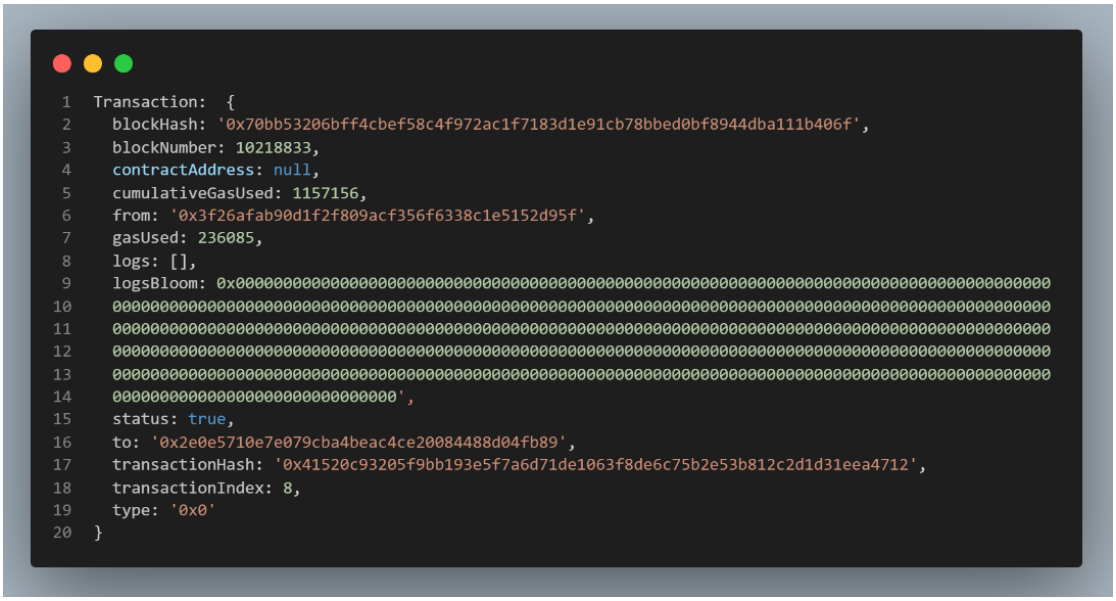


Fig. 44 Salida por consola en Visual Studio Code al llamar la función de registro de estudiantes (estudiante 2), (addNewUser).

Cuando las transacciones son realizadas con éxito, se puede acceder a la información desde sistema para visualizar los datos guardados en la blockchain para esto se accederá a la función get() del contrato inteligente, esta obtendrá la información de un estudiante registrado mediante su cuenta de wallet con la que confirmo el correo electrónico el proceso de tanto para registro, la confirmación y ver la información se encuentran en el Anexo 2, la salida por consola de la transacción para la función get() se pueden visualizar en la Fig. 45 y 46.

Fig. 45 Salida por consola en Visual Studio Code al llamar la función para visualizar los datos del estudiante (estudiante 1), (getInfoUser)

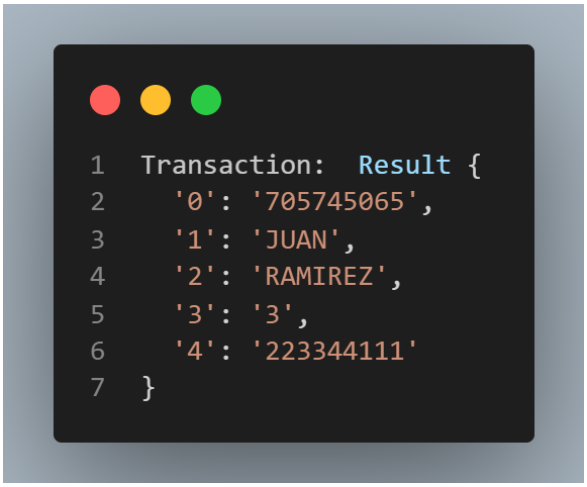


Fig. 46 Salida por consola en Visual Studio Code al llamar la función para visualizar los datos del estudiante (estudiante 2), (getInfoUser)

En todas las salidas de las transacciones, se puede observar los datos como el hash de transacción, el emisor de la transacción, la dirección del contrato, el número de bloque en el que se minó la transacción, el hash del bloque entre otros datos, esta información es importante al momento de comprobar que un registro fue exitoso y se introdujo a la blockchain, todos estos datos se los puede revisar en la interfaz de Etherscan de la red Ropsten el cual presenta todas las transacciones realizadas en el contrato inteligente como se puede ver en la Fig. 47.

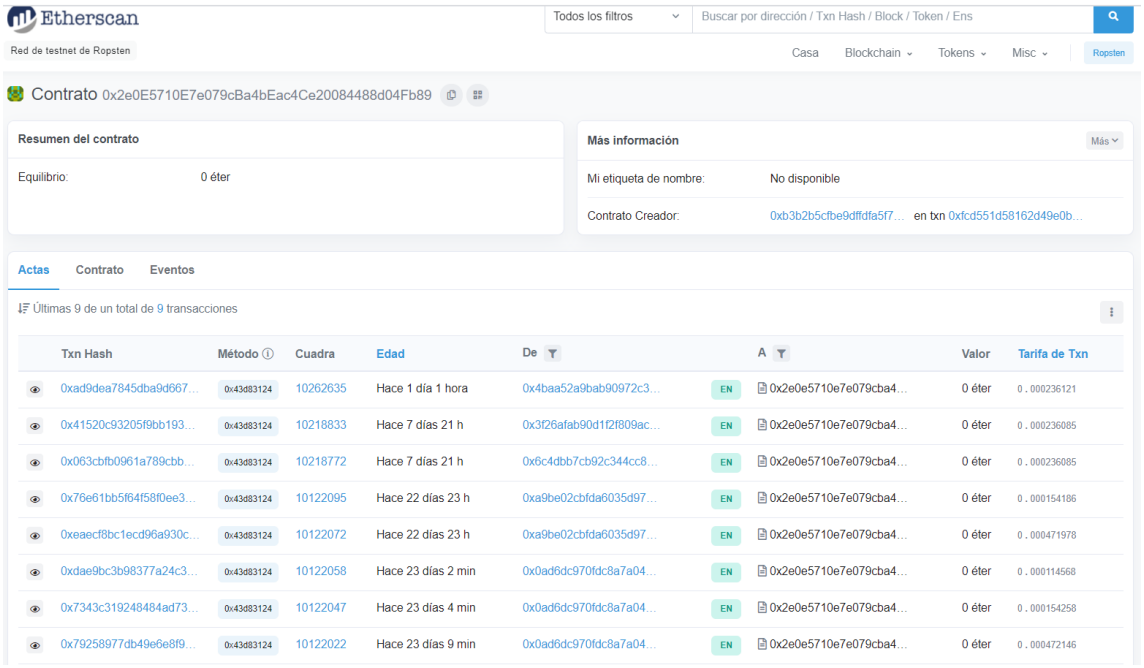


Fig. 47 Listado de las transacciones registradas en la cadena de bloques respecto al contrato inteligente y sus interacciones.

6.9. Comparativa y Análisis de Seguridad y Rendimiento

Para determinar si existe la optimización de seguridad al implementar las tecnologías de blockchain y contratos inteligentes, así como para verificar el rendimiento que estas ofrecen, se realizará una comparativa ante los resultados obtenidos.

6.9.1. Seguridad

A nivel de la seguridad de los datos de un sistema existen tres parámetros que se consideran importantes para determinar si una aplicación es segura o no, a continuación, se examina el valor de seguridad que brinda la aplicación de blockchain a un control de acceso.

TABLA XIV

ANÁLISIS DE SEGURIDAD

	Blockchain	Base de Datos
Confidencialidad	• Seguridad de la infraestructura clave pública (KPI) que identifica a cada usuario. • La encriptación completa de los bloques de datos.	• No existe encriptación directamente, pero se puede aplicar por el administrador. • Sistema centralizado
Integridad	• Inmutabilidad y transparencia. • Firmada digitalmente y con marca de tiempo	• Los datos pueden ser modificados por el administrador o decide quien accede a ellos.
Disponibilidad	• Red distribuida no interrumpe el servicio, todos los nodos comparten una copia	• Al funcionar (cliente-servidor), existen fallas.

Se puede determinar que a nivel de seguridad de la información la tecnología blockchain proporciona mejor infraestructura para cuidar los datos que sean entregados para ser almacenados; la confidencialidad, la integridad y la disponibilidad son los tres parámetros fundamentas en la seguridad, (KPI) esta aplicada en toda la infraestructura de la cadena de bloques la cual es un conjunto de políticas y procedimientos que ayudan a autentificar y autorizar a las partes de un contrato a realizar una transacción en contraste la base de datos conservan su información de forma simple aunque se puede aplicar la encriptación de los datos esto queda al parecer del encargado o administrador. En este punto blockchain encripta todos los datos que se le brindaron en la transacción y además cada bloque donde está el contrato inteligente también el encriptado por una clave hash ofreciendo inmutabilidad de la información previamente registrada, en cambio las bases de todos son manejadas por terceros los cuales pueden decidir si alterar información a conveniencia o determinar quién accede. Y como las BD se manejan de forma cliente-servidor si existe una falla y no se han

hecho respaldos no se podrá acceder al servicio mientras que la blockchain si existe la caída o fallo en algún nodo el servicio no se interrumpe ya que existen las copias en cada nodo de dicha red.

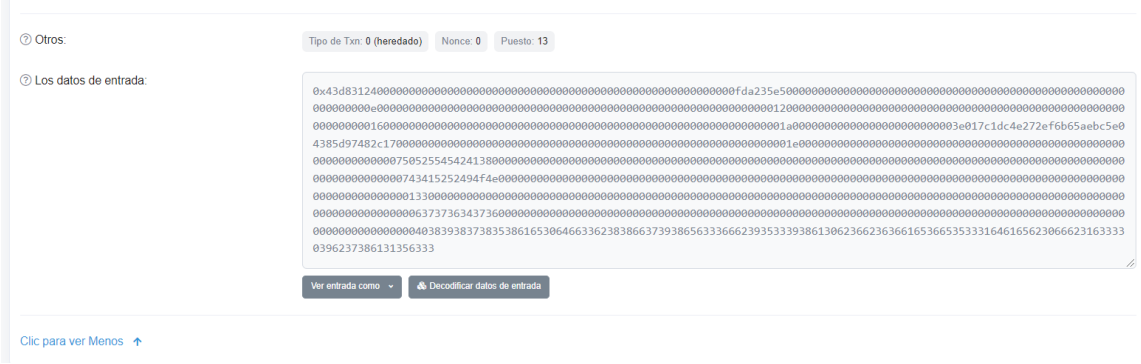


Fig. 48 Datos encriptados que forman parte de una transacción.

+ Opciones

				ID_ESTUDIANTE	NOMBRE_ESTUDIANTE	APELLIDO_ESTUDIANTE	ID_CURSO	ID_TARJETA	ESTADO
☐	Editar	Copiar	Borrar	1104161763	DANNY	JARAMILLO	5	B5 02 25 C3	ACTIVADO
☐	Editar	Copiar	Borrar	1104787021	GABRIEL	VITERI	5	80	DESACTIVADO
☐	Editar	Copiar	Borrar	1105209397	KAREN	NAGUA	5	26	DESACTIVADO
☐	Editar	Copiar	Borrar	1105381014	OMAR	SANMARTIN	5	11	DESACTIVADO
☐	Editar	Copiar	Borrar	1105537664	EVELYN	QUEVEDO	5	48	DESACTIVADO
☐	Editar	Copiar	Borrar	1105642076	ALEX	NOLE	5	108	DESACTIVADO

Fig. 49 Información existente en la base de datos del sistema.

Como resultado se puede determinar el mejoramiento en seguridad de los datos en la Fig. 48 se observa los datos enviados de una transacción conservados encriptados y en la Fig. 49 como se guarda la información del registro sin aplicar la blockchain.

6.9.2. Rendimiento

Un buen rendimiento es indispensable para una aplicación, para eso se examinará el funcionamiento de la blockchain y su capacidad para el registro de usuarios en su red.

- **Ingreso de información**

TABLA XV

TIEMPOS DE REGISTROS

Ingresos de datos en la Blockchain	
	Tiempo de ejecución
Estudiante 1	6.16 s
Estudiante 2	5.78 s
Estudiante 3	7.75 s
Total	19.69 s
Ingreso de datos en Base de datos	
Estudiante 1	198 ms
Estudiante 2	167 ms
Estudiante 3	175 ms
Total	540 ms

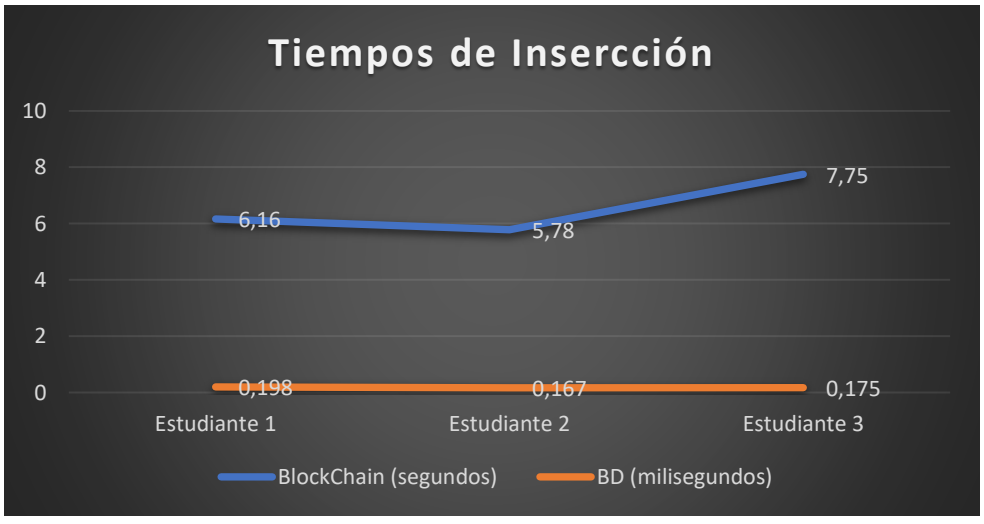


Fig. 50 Tiempos en la inserción de información tanto a la blockchain como a la BD.

Como se puede observar en TABLA XV y en la Fig. 50, los tiempos para registrar la información son mucho mayor en la blockchain con respecto a la BD ya que es un sistema centralizado y mucho mas directo, en cambio el sistema descentralizado que tiene la infraestrctura de la cadena de bloques replica la informacion y la encripta tomando mas tiempo, pero como son procesos de resgistros y no se esperan respuestas rapidas para ningun otro proceso siendo aceptables para el mejoramiento de un sistema.

- **Lectura de información**

TABLA XVI

TIEMPOS DE LECTURA

Lectura de datos en la Blockchain	
	Tiempo de ejecución
Estudiante 1	227 ms
Estudiante 2	200 ms
Estudiante 3	188 ms
Total	615 ms
Lectura en la Base de datos	
Estudiante 1	81 ms
Estudiante 2	74 ms
Estudiante 3	94 ms
Total	249 ms

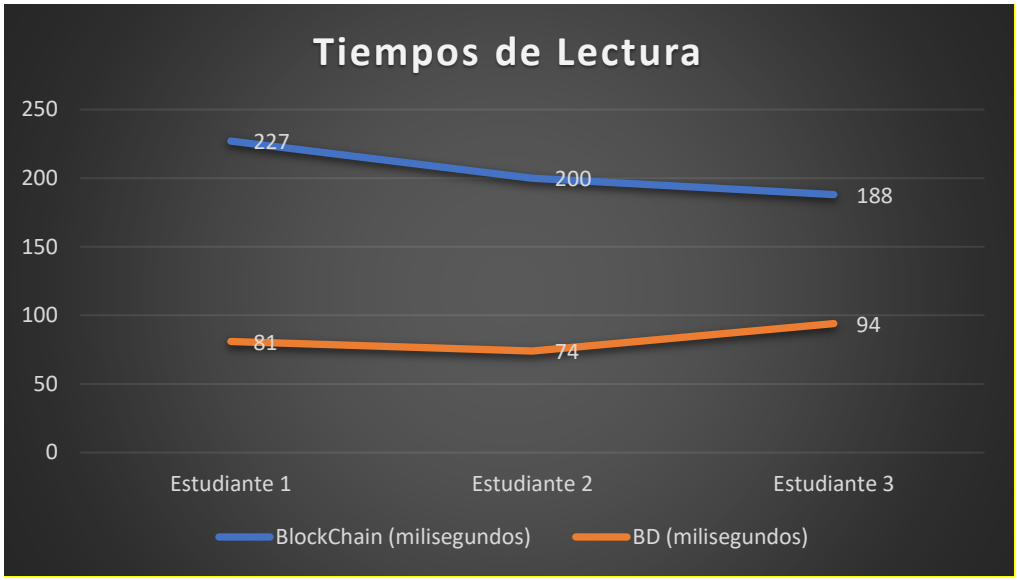


Fig. 51 Tiempos de lectura de la información.

En la obtencion de la información tanto de la cadena de bloques como de la base de datos se puede observar tanto en la TABLA XVI como en la Fig. 51, que existe un poco menos de diferencia de teimpo, pero sigue necesitando mas tiempo de ejeución la blockchain ya que para acceder dicha infirmacion necesita verificar la cuenta de quien envio la transacción, por lo contrario la base de datos lo hace mucho mas rapido.

La diferencia de tiempos que existen con la blockchain se debe a su arquitectura y a la seguridad que aplica a los datos en este caso para el registro de usuarios para diferncia y enteder esto se explica en la Fig. 52 donde cada transaccion se almacenara en un bloque que tendra un id unico para poder acceder a esos y se tendra que ir siguiendo toda la cadena hasta llegar al punto de regsitro o de comsulta. Por lo contrario para almacenar información en la base de datos se lo resliza de forma directa como en la Fig.53 sin inspeccionar nada antes o encriptar por eso alcanza esos tiempos.

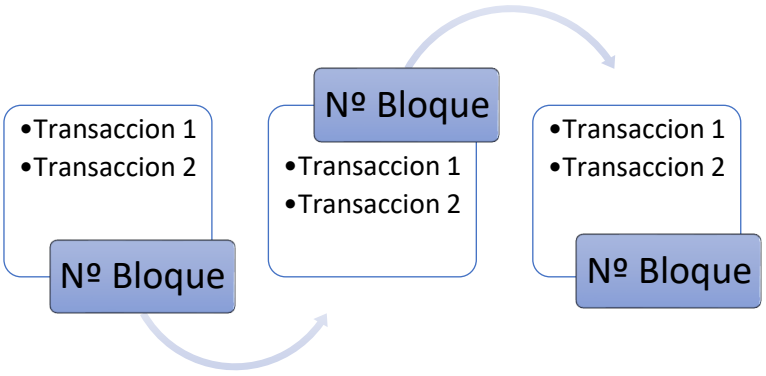


Fig. 52 Registro de información en la blockchain.

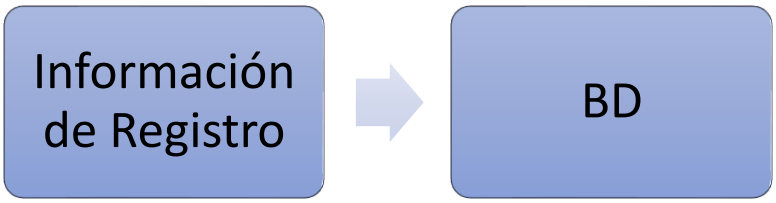


Fig. 53 Registro de información en la BD.

En la Fig. 54 se puede observar como fueron obtenidos los tiempos de cada transacción asi como el resgitro de los datos mediante el uso del navegador y las herramientas que el mismo brinda.

Fig. 54 Obtención de tiempos mediante la herramienta del navegador.

7. Discusión.

De acuerdo con lo realizado durante la investigación, queda en evidencia que la integración de blockchain y contratos inteligentes tiene aún un gran camino por recorrer, pero también, se puede deducir que la combinación de estas tecnologías genera enfoques completamente nuevos adaptables a sistemas y aplicaciones que se basan en paradigmas centralizados para el manejo de sus datos y gestionar sus procesos.

En el presente trabajo se presentó un procedimiento para implementar la tecnología de los contratos inteligentes y la blockchain enfocada a optimizar el proceso de registro de estudiantes del tema de titulación **"DESARROLLO DE UN PROTOTIPO WEB PARA EL CONTROL DE ACCESO A LOS CENTROS DE CÓMPUTO DE LA CARRERA DE INGENIERÍA EN SISTEMAS DE LA UNIVERSIDAD NACIONAL DE LOJA"** [79] por medio del uso de la plataforma Ethereum; dando como resultado la propuesta del modelo de solución, a pesar de esto, los campos de aplicación pueden ser infinitos en cualquier ámbito de la vida cotidiana.

Objetivo N°1: Realizar el análisis de los requerimientos necesarios para el desarrollo de un Contrato Inteligente y determinar las herramientas y plataforma a usar.

Mediante aplicación de la revisión sistemática de literatura perteneciente a la metodología de Barbara Kitchenham con cada una de las fases principales y sus etapas, se pudo determinar los trabajos que guardan una relación al tema propuesto y así encontrar los requerimientos necesarios para desarrollar un contrato inteligente en Ethereum. Estos trabajos seleccionados permitieron definir un proceso para el desarrollo del contrato que son: programarlo, compilarlo, desplegarlo y probarlo para lo cual intervinieron lenguajes de programación como Solidity, librerías para ayudar en la conexión con la blockchain.

Además de los estudios obtenidos se precisó las herramientas que deben intervenir para establecer un adecuado funcionamiento entre la blockchain y los contratos como: Remix-IDE la cual fue usada en este proyecto debido a que aportaba con todas las características para el desarrollo de contratos que ya fueron explicadas y que además la plataforma Ethereum la recomienda, MetaMask la billetera para el manejo de la transacciones esencial para el uso de blockchain y Ropsten (tesnet) una blockchain de pruebas para aplicaciones descentralizadas y que es totalmente aparecida a la red de Ethereum.

Con la aplicación de esta revisión sistemática de literatura y mediante sus resultados se pretende establecer una ayuda para aplicar contratos inteligentes en sistemas de registros,

así mismo asentar bases de las herramientas que se necesitan para establecer un funcionamiento entre contratos inteligentes y la blockchain.

Objetivo N.º 2: Desplegar el Contrato Inteligente y el módulo de registro de usuarios.

Para realizar el despliegue del contrato se siguió todo el proceso previo que fue: programar o crear el contrato inteligente, la compilación para errores, el despliegue en la red de pruebas Ropsten.

Para escribir el contrato inteligente se utilizó la versión más actual y estable en ese momento del lenguaje Solidity 0.7.1 cabe mencionar que de los trabajos relacionados no establecen ni describen la versión aplicada es el caso de M. García [17] u optan por utilizar versiones experimentales para evitar errores como es R. Eduardo [7], esto conlleva a que el código tenga fallas de compilación ya que no son versiones estables y no servirán cundo exista una actualización. Además en este trabajo semejante se registra información y lo realiza con herramientas que crean un blockchain pequeña para realizar todo el proceso del contrato inteligente ya explicado anteriormente de forma local, en contraste el presente trabajo se utilizó remix-IDE que determina los errores ya sean de programación o de fugas de información antes de ser desplegado en la red Ropsten la cual es una cadena de bloques igual a la de Ethereum para obtener mejor resultados de funcionamiento semejantes a la realidad y la cual permite observar como se adicionan las transacciones, como se guarda la información y consultarla de forma ágil.

Como un aporte más de este proyecto se presenta pasos a seguir para adicionar los complementos de la configuración de MetaMask para poder transaccionar e interactuar el contrato y el usuario que se va a registrar, y también la serie de pasos para realizar el despliegue del contrato inteligente en la blockchain usando remix-IDE.

Objetivo N.º 3: Elaborar un entorno de Pruebas e interacción para medir la usabilidad y el funcionamiento del Contrato Inteligente dentro del ambiente local (EVM).

Finalizado el despliegue del contrato inteligente y se configuraron los componentes necesarios para la interacción, se utilizó etherscan que analiza los bloques de una cadena, así como las transacciones de un contrato que conste en dicha red, además se configuro el nodo intermediario entre el sistema y la blockchain para realizar las pruebas e interacciones de funcionamiento del contrato inteligente. Luego de visualizar los resultados de las pruebas funcionales se evidenció efectos positivos a nivel de seguridad de la información, donde los datos que llevan para el registro tiene mayor fianza cumpliendo con tres de las características importantes que son la confidencialidad, integridad y disponibilidad esto es gracias a las

características de la blockchain y su arquitectura encriptando los datos de registro en cada bloque de la cadena siendo su información inmutable y no dependiente de terceras personas.

Por lo contrario se contrasta a nivel del funcionamiento con los tiempos de ejecución el sistema que maneja base de datos tiene tiempos muchos menores tanto en la inserción de datos con un total de 540 milisegundos en los tres registros hechos, y para la lectura de esos datos se obtuvo un tiempo de 249 milisegundos, por lo contrario al trabajar con la blockchain se visualizó un tiempo de 19,69 segundos en el ingreso de datos y al momento de extraerlos se observó un tiempo de 615 milisegundos esto se debe a la forma de guardar la información ya que tiene que agregarse a un bloque y crear un id de transacción donde la seguridad de toda esta cadena causa que se extienda los tiempos de estos procesos, tomando en cuenta que es un sistema de control de acceso no amerita respuestas demasiado rápidas siendo factible, aplicable y sobre todo optimizando seguridad aunque exista un poco más de demora en él envió y obtención de la información de la cadena de bloques.

8. Conclusiones.

- En el presente trabajo se puede concluir que la intervención de la tecnología blockchain y su descentralización de sistemas realiza un gran aporte para actualizar sistemas centralizados ofreciendo tres características fundamentales en la seguridad que se evidenciaron al aplicar la Blockchain en el registro de estudiantes derivando una comparativa, como son la confidencialidad de los datos que no estén en manos de terceras personas, la integridad que no puedan ser modificados a favor de ninguna de las partes y la disponibilidad que el servicio siempre esté disponible al usuario. A pesar de que esta seguridad afectara en la rapidez de respuesta con tiempos de19,69 segundo y 615 milisegundos en comparación a los tiempos de una Base de datos de 540 milisegundos y 249 milisegundos existe una gran diferencia lo cual afecta a sistemas donde las respuestas deben ser rápidas, pero a pesar de ello se pueden vincular en sistemas en los cuales no sea una necesidad obtener un funcionamiento veloz.

- A través de la revisión bibliográfica aplicada se pudo identificar que la información existente para el desarrollo y aplicación de contratos inteligentes en la blockchain es muy escasa o no es concreta para ayudar a los desarrolladores a implementar estas tecnologías en los sistemas de Ecuador, eso quiere decir que de la información obtenida se habla parcialmente del desarrollo de los contratos y se centran en el funcionamiento del sistema en general, aunque se nombre los contratos inteligentes, no especifican las herramientas y librerías que usan para integrarlos a la blockchain y la aplicación, pero también, se menciona a la plataforma Ethereum principalmente en trabajos similares o que tienen una aplicación en sistemas de registro denominándola una tecnología que apunta hacia el futuro; a pesar de esto se logró determinar las herramientas adecuadas para este trabajo y mejorar el proceso de registro de estudiantes, obteniendo resultados positivos.

- En revisión bibliográfica se obtuvo como resultado el uso de Remix IDE para programar el contrato, realizar las pruebas, la depuración y el despliegue del mismo en la red de pruebas Ropsten, también que el uso de la librería web3.js ayuda a establecer la conexión entre la blockchain, el contrato inteligente y la interfaz del usuario, y por último se estableció que el uso de Ethereum para la implementación de contratos inteligentes dentro de aplicaciones y sistemas como la plataforma más usada.

- El proceso de despliegue el contrato inteligente en este proyecto se lo realizó mediante Remix IDE, donde el contrato conservará los datos de registró de los estudiantes, esta información permanecerá almacenada mediante una transacción en un bloque de la red Ropsten conservando su integridad, además con el uso de MetaMask se firmará las transacciones que se envíen de cada cuenta permitiendo optimizar la seguridad al encriptar los datos de los estudiantes y que la interacción sea personal.

- Realizar el proceso de registro de los estudiantes del sistema con la intervención de contratos inteligentes combinados con la tecnología Blockchain genera un gran impacto a nivel de la educación superior del país, permitiendo conservar de forma confiable el registro de los datos que brinda un usuario y que a su vez estén disponibles para su uso sin temor a ser modificados o que se les niegue el acceso, esto significa que el cumplimiento de los tres parámetros de la seguridad de la información se cumple afectando a la rapidez de contestación del sistema.

9. Recomendaciones.

De acuerdo al Trabajo de Titulación realizado, se puede presentar las siguientes recomendaciones:

- Usar la plataforma Ethereum para el manejo de los contratos inteligentes y las cadenas de bloques públicas, debido a que la combinación y su funcionamiento permiten accesos rápidos, reducir costos, mejorar procesos, generar más seguridad, descentralizar sistemas, mejorar el enfoque de aplicaciones, etc.
- Utilizar el entorno de desarrollo integrado Remix para el desarrollo de los contratos inteligentes ya que cuenta con lo necesario para programar, compilar, desplegar y realizar pruebas en los contratos, además que cuenta con herramientas diseñadas para acoplarse entre sí sin mayores configuraciones y que estas facilitan la interacción de los usuarios en esos sistemas.
- Mayor uso de la librería web3.js debido a su alta facilidad de integración con herramientas relacionadas al manejo de contratos inteligentes, además esta librería es una de las principales que trabaja en conjunto con MetaMask facilitando al usuario mediante el manejo de una billetera firmar transacciones personales que trabajan con los contratos.
- Hacer uso del lenguaje Solidity para programar los contratos inteligentes debido a que es orientado a contratos y su sintaxis es similar a la de JavaScript, además Ethereum lo recomienda directamente.
- Utilizar la red de pruebas Ropsten ya que gracias a su cadena de bloques con contratos funcionando brinda una mayor similitud a la red Ethereum real, sin olvidar que es muy sencillo obtener el Éther para realizar las transacciones al hacer las pruebas de funcionamiento.

Trabajos futuros

- Implementar una solución para el sistema web actual migrando de php a node.js basándose en que este es un entorno de ejecución en tiempo real, esto quiere decir que incluye lo necesario ejecutar un programa con un servidor incluido como si se tratara de aplicaciones independientes convirtiéndolo en ligero y eficiente, mediante esta investigación se usó de node.js para establecer la conexión al contrato inteligente y observar las respuestas en tiempos de conexión ya que fueron mucho más rápidos, además el uso de librerías en node.js para la implementación de contratos brinda facilidades para el programador.

- Se puede adoptar en el actual sistema web de control de acceso la aplicación del Blockchain con sus características siendo una de estas los contratos inteligentes para mejorar la productividad y optimizar todos los procesos, destacando el uso de la plataforma Ethereum, para permitir generar procesos más rápidos y con mayor seguridad proporcionando una comparación con los resultados obtenidos en este trabajo de investigación y mejorar la implementación de las nuevas tecnologías en el país.
- Recabar nueva información para mejorar la revisión bibliográfica con nuevas herramientas y librerías con sus respectivas versiones para que facilite la implementación de la blockchain y contratos inteligentes dentro de aplicaciones existentes o para los futuros proyectos de innovación.

10. Bibliografía.

[1] R. P. Godinho, “¿Por qué el Blockchain es una tecnología disruptiva?,” 2019. [Online]. Available: https://www.projectco3.eu/es/2019/04/08/por-que-el-blockchain-es-una-tecnologia-disruptiva/.

[2] MINTEL, *Libro Blanco de la Sociedad de la Información y del Conocimiento*, vol. 1. 2018.

[3] Buterin and Vitalik, “Ethereum White Paper: A Next Generation Smart Contract {\&} Decentralized Application Platform,” *Etherum*, no. January, pp. 1–36, 2014.

[4] B. Kitchenham, “Procedures for Performing Systematic Literature Reviews,” *Jt. Tech. Report, Keele Univ. TR/SE-0401 NICTA TR-0400011T.1*, vol. 33, p. 33, 2004.

[5] R. Hans, H. Zuber, A. Rizk, and R. Steinmetz, “Blockchain and smart contracts: Disruptive technologies for the insurance market,” *AMCIS 2017 - Am. Conf. Inf. Syst. A Tradit. Innov.*, vol. 2017-Augus, no. August, pp. 1–10, 2017.

[6] J. F. Galvez, J. C. Mejuto, and J. Simal-Gandara, “Future challenges on the use of blockchain for food traceability analysis,” *TrAC - Trends Anal. Chem.*, vol. 107, pp. 222–232, 2018.

[7] R. C. L. Eduardo, “Propuesta de una aplicación basada en la tecnología blockchain para el registro de títulos académicos,” UNIVERSIDAD CENTRAL DEL ECUADOR, 2019.

[8] M. Alharby and A. van Moorsel, “The Impact of Profit Uncertainty on Miner Decisions in Blockchain Systems,” *Electron. Notes Theor. Comput. Sci.*, vol. 340, pp. 151–167, 2018.

[9] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, “On blockchain and its integration with IoT. Challenges and opportunities,” *Futur. Gener. Comput. Syst.*, vol. 88, no. 2018, pp. 173–190, 2018.

[10] Y. Huang, Y. Bian, R. Li, J. L. Zhao, and P. Shi, “Smart Contract Security: A Software Lifecycle Perspective,” *IEEE Access*, vol. 7, pp. 150184–150202, 2019.

[11] M. Singh and S. Kim, “Branch based blockchain technology in intelligent vehicle,” *Comput. Networks*, vol. 145, pp. 219–231, 2018.

[12] I. Makhdoom, M. Abolhasan, H. Abbas, and W. Ni, “Blockchain’s adoption in IoT: The challenges, and a way forward,” *J. Netw. Comput. Appl.*, vol. 125, pp. 251–279, 2019.

[13] V. M. S. Grewal-Carr, “Blockchain Opportunity Contents,” *Deloitte.*, p. 27, 2016.

[14] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends,” *Proc. - 2017 IEEE 6th Int. Congr. Big Data, BigData Congr. 2017*, pp. 557–564, 2017.

[15] M. E. Peck, “¿Necesitas una Blockchain? - Espectro IEEE,” 2017. [Online]. Available: https://spectrum.ieee.org/computing/networks/do-you-need-a-blockchain. [Accessed: 28-Feb-2020].

[16] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, “A survey on privacy protection in blockchain system,” *J. Netw. Comput. Appl.*, vol. 126, pp. 45–58, 2019.

[17] M. Garcia, “Modelo de solucion mediante el uso de Smart Contracts para el registro de matrículas de estudiantes en la UCE,” UNIVERSIDAD CENTRAL DEL ECUADOR, 2019.

[18] N. RODRIGUEZ, “Algoritmos de consenso: la raíz de la tecnología blockchain,” *101 Blockchains*, 2018. [Online]. Available: https://101blockchains.com/es/algoritmos-de-consenso-blockchain/#5ç. [Accessed: 19-Feb-2020].

[19] F. Tschorsch and B. Scheuermann, “Bitcoin and beyond: A technical survey on decentralized digital currencies,” *IEEE Commun. Surv. Tutorials*, vol. 18, no. 3, pp. 2084–2123, 2016.

[20] R. álvarez, A. Andrade, and A. Zamora, “Optimizing a password hashing function with hardware-accelerated symmetric encryption,” *Symmetry (Basel).*, vol. 10, no. 12, pp. 1–11, 2018.

[21] S. Kornmesser, “Theoretizität im logischen empirismus und im strukturalismus - Erläutert am fallbeispiel des neurobiologischen konstruktivismus,” *J. Gen. Philos. Sci.*, vol. 39, no. 1, pp. 53–67, 2008.

[22] J. Garcia-Alfaro, G. Navarro-Arribas, H. Hartenstein, and J. Herrera-Joancomartí, “Data Privacy Management, Cryptocurrencies and Blockchain Technology,” *Proceedings*, pp. 297–315, 2017.

[23] L. Luu, D. H. Chu, H. Olickel, P. Saxena, and A. Hobor, “Making smart contracts

smarter," *Proc. ACM Conf. Comput. Commun. Secur.*, vol. 24-28-Octo, pp. 254–269, 2016.

[24] W. Dingman *et al.*, "Defects and vulnerabilities in smart contracts, a classification using the NIST bugs framework," *Int. J. Networked Distrib. Comput.*, vol. 7, no. 3, pp. 121–132, 2019.

[25] A. Pinna, S. Ibba, G. Baralla, R. Tonelli, and M. Marchesi, "A Massive Analysis of Ethereum Smart Contracts Empirical Study and Code Metrics," *IEEE Access*, vol. 7, pp. 78194–78213, 2019.

[26] Miethereum, "¿Qué son los Smart Contracts o Contratos Inteligentes?" [Online]. Available: https://www.miethereum.com/smart-contracts/#toc2. [Accessed: 04-Feb-2020].

[27] T. N. Szabo, Satoshi Nakamoto Institutue, ""The idea of smart contracts,"" *[Online]*, 1997.

[28] H. Min, "Blockchain technology for enhancing supply chain resilience," *Bus. Horiz.*, vol. 62, no. 1, pp. 35–45, 2019.

[29] A. Rosic, "¿Qué son los contratos inteligentes? [Guía definitiva para principiantes sobre contratos inteligentes]," 2017. [Online]. Available: https://blockgeeks.com/guides/smart-contracts/. [Accessed: 23-Feb-2020].

[30] M. Alharby and A. van Moorsel, "Blockchain Based Smart Contracts : A Systematic Mapping Study," pp. 125–140, 2017.

[31] Buterin and Vitalik, "Ethereum White Paper: A Next Generation Smart Contract & Decentralized Application Platform," *Etherum*, no. January, pp. 1–36, 2014.

[32] Solidity, "Introducción a los Contratos Inteligentes — documentación de Solidity," 2019. [Online]. Available: https://solidity-es.readthedocs.io/es/latest/introduction-to-smart-contracts.html. [Accessed: 29-Jan-2020].

[33] JP Buntinx, "¿Qué es la máquina virtual Ethereum? ," 2017. [Online]. Available: https://themerkle.com/what-is-the-ethereum-virtual-machine/. [Accessed: 18-Mar-2020].

[34] Preethi Kasireddy, "¿Cómo funciona Ethereum, de todos modos? - Preethi Kasireddy - Medio," 2017. [Online]. Available: https://medium.com/@preethikasireddy/how-does-

ethereum-work-anyway-22d1df506369. [Accessed: 18-Mar-2020].

[35] Ethereum, “Solidity Documentation,” *Ethereum Found.*, vol. 1, no. 1, 2020.

[36] R. Modi, *Solidity programming essentials_ a beginner’s guide*. 2018.

[37] M. E. SÁENZ, “Contratos Electronicos Autoejecutables (Smart Contract) Y Pagos Con Tecnología Blockcahin,” *Rev. Estud. Eur.*, pp. 69–95, 2017.

[38] K. Carrión, “Análisis de la utilización de la tecnología Blockcahin para la gestión de la información en sisteas de alarmas residenciales.,” p. 156, 2018.

[39] C. Castaño and M. Quecedo, “Introducción a la metodología de investigación cualitativa,” *Rev. psicodidáctica*, vol. 14, no. 14, pp. 5–40, 2002.

[40] D. R., “¿Qué es el método científico experimental? Ejemplos y pasos.” [Online]. Available: https://investigacioncientifica.org/que-es-el-metodo-cientifico-experimental/. [Accessed: 25-May-2021].

[41] J. P. Cruz, Y. Kaji, and N. Yanai, “RBAC-SC: Role-based access control using smart contract,” *IEEE Access*, vol. 6, pp. 12240–12251, 2018.

[42] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, “Smart contract-based access control for the internet of things,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1594–1605, 2019.

[43] Mark Petticrew y Helen Roberts, *Systematic Reviews in the Social Sciences*. 2006.

[44] S. Wang, X. Wang, and Y. Zhang, “A Secure Cloud Storage Framework With Access Control Based on Blockchain,” *IEEE Access*, vol. 7, pp. 112713–112725, 2019.

[45] S. Wang, R. Pei, and Y. Zhang, “EIDM: A Ethereum-Based Cloud User Identity Management Protocol,” *IEEE Access*, vol. 7, pp. 115281–115291, 2019.

[46] H. R. Hasan and K. Salah, “Combating Deepfake Videos Using Blockchain and Smart Contracts,” *IEEE Access*, vol. 7, no. c, pp. 41596–41606, 2019.

[47] H. R. Hasan and K. Salah, “Blockchain-based Proof of Delivery of Physical Assets with Single and Multiple Transporters,” *IEEE Access*, vol. PP, no. 8, p. 1, 2018.

[48] H. R. Hasan and K. Salah, “Proof of Delivery of Digital Assets Using Blockchain and Smart Contracts,” *IEEE Access*, vol. 6, no. 8, pp. 65439–65448, 2018.

[49] W. E. I. Xiong and L. I. Xiong, “Smart Contract Based Data Trading Mode Using Blockchain and Machine Learning,” *IEEE Access*, vol. PP, p. 1, 2019.

[50] C. Xu, Y. Fang, and Y. Ma, “ScienceDirect Integrated Integrated Application Application of of Blockchain Blockchain in in the the Electric Electric Information Information Management System Management System,” *Procedia Comput. Sci.*, vol. 162, no. Itqm 2019, pp. 88–93, 2020.

[51] M. Debe, K. Salah, M. Habib, and U. R. Rehman, “Monetization of Services Provided by Public Fog Nodes Using Blockchain and Smart Contracts,” vol. 8, 2020.

[52] S. Pal, T. Rabehaja, M. Hitchens, V. Varadharajan, and A. Hill, “On the Design of a Flexible Delegation Model for the Internet of Things Using Blockchain,” *IEEE Trans. Ind. Informatics*, vol. 16, no. 5, pp. 3521–3530, 2020.

[53] G. Gürsoy, C. M. Brannon, and M. Gerstein, “Using Ethereum blockchain to store and query pharmacogenomics data via smart contracts,” *BMC Med. Genomics*, vol. 13, no. 1, pp. 1–11, 2020.

[54] M. Debe, K. Salah, M. Habib, and U. R. Rehman, “IoT Public Fog Nodes Reputation System : A Decentralized Solution Using Ethereum Blockchain,” *IEEE Access*, vol. 7, pp. 178082–178093, 2019.

[55] S. Wang, Y. Zhang, and Y. Zhang, “A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems,” *IEEE Access*, vol. 6, pp. 38437–38450, 2018.

[56] J. S. Park, T. Y. Youn, H. Bin Kim, K. H. Rhee, and S. U. Shin, “Smart contract-based review system for an IoT data marketplace,” *Sensors (Switzerland)*, vol. 18, no. 10, pp. 1–16, 2018.

[57] O. López-pintado, L. García-bañuelos, I. Weber, and A. Ponomarev, “Caterpillar : A business process execution engine on the Ethereum blockchain,” no. April, pp. 1162–1193, 2019.

[58] D. C. Nguyen, P. N. Pathirana, and S. Member, “Blockchain for Secure EHRs Sharing of Mobile Cloud Based E-Health Systems,” *IEEE Access*, vol. 7, pp. 66792–66806, 2019.

[59] M. Debe, K. Salah, M. Habib, and U. R. Rehman, “Blockchain-Based Decentralized

Reverse Bidding in Fog Computing," vol. 4, 2020.

[60] Bit2Me Academy, "Qué es MetaMask - La forma más fácil de usar DApps." .

[61] WeTrust Blog, "WeTrust Community Update — September 12 2017 | by WeTrustLeonD," 2017. .

[62] E.-I. Remix, "Welcome to Remix documentation! — Remix, Ethereum-IDE 1 documentation." .

[63] E.-I. Remix, "Remix-IDE Layout — Remix, Ethereum-IDE 1 documentation." .

[64] M. Neto, "Get Ropsten Ethereum — The Easy Way. | by Moritz Neto | bitfwd | Medium," 2018. .

[65] Truffle, "Truffle | Truffle and MetaMask | Documentation | Truffle Suite." .

[66] "Documentación para Visual Studio Code." [Online]. Available: https://code.visualstudio.com/docs. [Accessed: 22-Mar-2021].

[67] "Visual Studio Code - Code Editing. Redefined." [Online]. Available: https://code.visualstudio.com/. [Accessed: 22-Mar-2021].

[68] P. Ordoñez *et al.*, "Ambiente Inteligente para el macro laboratorio de formación conjunta en la Facultad de Energía de la Universidad Nacional de Loja. SmartLab," *Univ. Nac. Loja*, p. 19, 2019.

[69] Y. Makarova and R. Langmann, "Prototype of the modern hands-on smart lab for automation engineering," *Proc. 2016 13th Int. Conf. Remote Eng. Virtual Instrumentation, REV 2016*, no. February, pp. 254–259, 2016.

[70] J. P. Quiroga, "Requerimientos Funcionales y No Funcionales," pp. 1–27, 2014.

[71] "Requerimientos Funcionales y No Funcionales, ejemplos y tips | by Requeridos Blog | Medium." [Online]. Available: https://medium.com/@requeridosblog/requerimientos-funcionales-y-no-funcionales-ejemplos-y-tips-aa31cb59b22a. [Accessed: 27-Aug-2020].

[72] Á. Suárez, "Tipos de datos III – APRENDE BLOCKCHAIN." [Online]. Available: https://aprendeblockchain.wordpress.com/desarrollo-en-ethereum/tipos-de-datos-iii/. [Accessed: 25-Mar-2021].

[73] D. D. G. e I. M. Alexander Preukschat, Carlos Kuchkovsky, Gonzalo Gómez Lardies, “Blockchain._La_revolucion_industrial_de_Internet,” *Blockchain. La Revoluc. Ind. internet*, vol. 1, p. 397, 2017.

[74] Remix, “Remix documentation,” 2018.

[75] J. Lucas, “Qué es NodeJS y para qué sirve | OpenWebinars,” 2019. [Online]. Available: https://openwebinars.net/blog/que-es-nodejs/. [Accessed: 02-May-2021].

[76] F. Vogelsteller, M. Kotewicz, J. Wilcke, and M. Oance, “web3.js Documentation,” 2021.

[77] Elmorg *et al.*, “Web2 vs Web3 | ethereum.org,” 2021. [Online]. Available: https://ethereum.org/en/developers/docs/web2-vs-web3/. [Accessed: 02-May-2021].

[78] U. de Alcalá, “Infura - Master Ingeniería Blockchain,” 2020. [Online]. Available: https://masterblockchain.net/infura-master-blockchain-online/. [Accessed: 02-May-2021].

[79] B. A. A. Alvarado, “CARRERA DE INGENIERIA EN SISTEMAS “ DESARROLLO DE UN PROTOTIPO WEB PARA EL CONTROL DE ACCESO A LOS CENTROS DE CÓMPUTO Certificación del director,” 2020.

11. Anexos

Anexo 1: Tablas de resumen

A continuación, desde la TABLA XIV hasta la TABLA XXXI, se presenta la información de los estudios relacionados al trabajo de investigación donde se puede observar el título del artículo, el resumen, la información más relevante y la conclusión más relevante, además se asignó un código para poder diferenciarlos en el documento.

TABLA XVII

DESCRIPCIÓN DEL ARTÍCULO D05

Código	**D05**	**Referencia**	[44]	**Año**	**2019**
Título	Un marco de almacenamiento seguro en la nube con control de acceso basado en Blockchain				
Resumen del articulo	Nuestro sistema tiene tres características principales. En primer lugar, como se utiliza la tecnología de la cadena de bloqueo del Etéreo, el propietario puede almacenar el texto cifrado de los datos a través de contratos inteligentes en una red de cadenas de bloques. En segundo lugar, el propietario de los datos puede establecer un período de acceso válido para el usuario de los datos, de modo que el texto cifrado sólo pueda descifrarse durante períodos de acceso válidos. Por último, como la creación e invocación de cada contrato inteligente puede almacenarse en la cadena de bloques, se logra la función de rastreo. El análisis de la seguridad y el experimento muestran que nuestro esquema es factible.				
Información relevante	No hay ninguna solución específica para realizar la integración de la idea de descentralización de la tecnología y el acceso a las cadenas de bloques tecnología de control. Todavía hay mucho que hacer en esta área.				
Conclusiones relevantes	Se propone un marco basado en la cadena de bloques. El tradicional algoritmo de encriptación basado en atributos de texto cifrado es transformado por la introducción del contrato inteligente de Ethereum la tecnología. Para evitar que la autoridad del centro siga siendo atacada, la clave de distribución ya no depende de la misma.				

TABLA XVIII

DESCRIPCIÓN DEL ARTÍCULO D13

Código	**D13**	**Referencia**	[55]	**Año**	**2018**
Título	Un marco basado en blockchain para compartir datos con control de acceso de grano fino en sistemas de almacenamiento descentralizados				
Resumen del articulo	Estudiamos el esquema de almacenamiento e intercambio de datos para sistemas de almacenamiento descentralizados, y proponer un marco que combine el sistema de almacenamiento descentralizado IPFS, la tecnología de encriptación basada en atributos y cadenas de bloqueo Ethereum (ABE). En este marco, el propietario de los datos tiene la capacidad de distribuir la clave secreta para los usuarios de los datos, y de encriptar los datos que compartidos especificando el acceso y el plan logra un control de acceso muy preciso sobre los datos. Al mismo tiempo, basado en un contrato inteligente en la cadena de bloqueo Ethereum, la función de búsqueda de palabras clave en el texto cifrado de la de almacenamiento, lo que resuelve el problema de que el servidor de la nube no pueda devolver todos los resultados buscados o devuelven resultados erróneos en los sistemas tradicionales de almacenamiento en la nube.				
Información relevante	Los usuarios no tienen que preocuparse de no poder acceder a sus propios datos, debido a que la disponibilidad de los datos puede ser garantizados por contratos inteligentes desplegados en él y sólo tienen que pagar una cuota regular por los datos que han almacenado.				
Conclusiones relevantes	En la actualidad, el almacenamiento tradicional en nubes puede hacer que los datos de los usuarios no estén disponible debido a factores de fuerza mayor (como desastres naturales, censores del gobierno, etc.). La tecnología ABE y la tecnología de cifrado con capacidad de búsqueda en el texto cifrado son tecnologías importantes para resolver la privacidad de los datos y las problemas de control de acceso.				

TABLA XIX

DESCRIPCIÓN DEL ARTÍCULO D17

Código	**D17**	**Referencia**	[58]	**Año**	**2019**
Título	Cadena de bloqueo para compartir de forma segura los EHR de los móviles. Sistemas de salud electrónica basados en la nube				
Resumen del articulo	En este documento, proponemos un novedoso marco de intercambio de EHR que combina la cadena de bloques y el sistema de le interplanetario descentralizado (IPFS) en una plataforma de nubes móvil. En particular, diseñamos un mecanismo fiable de control de acceso utilizando contratos inteligentes para lograr un intercambio seguro de EHR entre diferentes pacientes y proveedores médicos. Presentamos un prototipo de implementación usando la cadena de bloqueo Ethereum en un escenario de intercambio de datos reales en una aplicación móvil con la computación en nube de Amazon.				
Información relevante	El control de acceso basado en cadenas de bloques proporciona varios nuevas funciones de seguridad para la salud electrónica. Primero, la cadena de bloques construye libros de cuentas inmutables de transacciones para el sistema de intercambio de datos. En segundo lugar, el control mediante cadenas de bloqueo puede lograr la propiedad de transparencia con la capacidad de resolver eficazmente la cuestión de fugas de datos que pueden ser causadas por servidores curiosos.				
Conclusiones relevantes	Los resultados de la aplicación muestran que nuestro marco puede permitir a los usuarios médicos compartir datos médicos sobre los entornos de nubes móviles de una manera fiable y rápida, en comparación con los esquemas convencionales. En particular, nuestro acceso de control puede identificar e impedir eficazmente el acceso no autorizado al sistema de salud electrónica, con el objetivo de lograr un nivel deseado de privacidad del paciente y seguridad de la red.				

TABLA XX

DESCRIPCIÓN DEL ARTÍCULO D18

Código	**D18**	**Referencia**	[59]	**Año**	**2020**
Título	Licitación inversa en la computación de la niebla descentralizado basado en cadenas de bloques				
Resumen del articulo	El esquema propuesto asegura que todos los nodos de niebla de la red pueden hacer por igual y de manera justa ofertas para ganar la licitación. El proceso de licitación incorpora los pagos automatizados al final del servicio. Nuestra propuesta se implementa mediante contratos inteligentes Ethereum. También integra un sistema de reputación para los nodos de niebla e impone una pena por el mal comportamiento de los nodos. Nuestra solución es totalmente descentralizada y proporciona un alto nivel de confianza, transparencia y seguridad. En el documento, presentamos la arquitectura del sistema, los detalles de implementación, y mostrar la correcta funcionalidad de la solución global propuesta.				
Información relevante	Los sistemas de computación de niebla permiten servicios de computación, comunicación y almacenamiento a través de los nodos de niebla (también conocidos como servidores Edge) cerca de los datos como los sensores a bordo de los móviles de punto final y Dispositivos de IO.				
Conclusiones relevantes	Nuestra solución basada en la cadena de bloqueo aborda el punto final de mal comportamiento y los proveedores de servicios de niebla en los que los deshonestos se penaliza el comportamiento reduciendo la devolución de los depósitos fondos.				

TABLA XXI

DESCRIPCIÓN DEL ARTÍCULO D02

Código	**D02**	**Referencia**	[47]	**Año**	**2018**
Título	Prueba de entrega de activos físicos basada en la cadena de bloques con uno o varios transportistas				
Resumen del articulo	En este documento, presentamos una solución y un marco general utilizando el popular Ethereum sin permiso para crear un sistema descentralizado y confiable de PoD que asegura la responsabilidad, la auditabilidad y la integridad. La solución utiliza contratos inteligentes de Ethereum para probar la entrega de un envío entre un vendedor y un comprador, independientemente del número de se necesitan transportadores intermedios. En nuestra solución propuesta, todas las entidades participantes se ven incentivadas a actuar con honestidad mediante el uso de un doble depósito de garantía. El pago automatizado en el Éter es un parte integral de la solución para asegurar que cada entidad obtenga su la parte prevista del Éter al ser entregado con éxito.				
Información relevante	Los sistemas actuales están en su mayor parte centralizados y se basan sobre terceros de confianza (TTP) para completar la entrega entre un vendedor y un comprador. Estos sistemas son difíciles de manejar y son costosos ya que involucran TTPs. No sólo esto, los TTP pueden ser un único punto de fracaso, y están sujetos a la piratería informática, la evasión de la privacidad y el compromiso.				
Conclusiones relevantes	Nuestra solución descentralizada de PoD utiliza una cadena de contratos, sin dependencias cíclicas, para satisfacer la necesidad de entregar entre múltiples transportadores. Probamos las funcionalidades clave, y demostró el comportamiento y los resultados correctos considerando múltiples escenarios de casos de prueba.				

TABLA XXII

DESCRIPCIÓN DEL ARTÍCULO D15

Código	**D15**	**Referencia**	[57]	**Año**	**2019**
Título	Caterpillar: un motor de ejecución de procesos empresariales en Blockchain de Ethereum				
Resumen del articulo	El artículo presenta un BPMN basado en una cadena de bloques motor de ejecución, llamado Caterpillar. Como cualquier motor de ejecución BPMN, Caterpillar apoya la creación de instancias de un modelo de proceso y permite a los usuarios para supervisar el estado de las instancias del proceso y ejecutar las tareas del mismo. El documento describe la arquitectura de Caterpillar y las interfaces que proporciona para apoyar la vigilancia de instancias del proceso, la asignación y ejecución de los elementos de trabajo, y la ejecución de las tareas de servicio.				
Información relevante	Las aplicaciones de cadenas de bloques existentes implementan procesos comerciales que involucran a múltiples participantes independientes, tales como procesos de gestión de la cadena de suministro. Sin embargo, la aplicación de los procesos comerciales utilizando las primitivas de bajo nivel proporcionadas por plataformas de cadenas de bloques es engorroso, propenso a errores y requiere habilidades especializadas.				
Conclusiones relevantes	En este artículo se presentó el diseño y la aplicación del sistema Caterpillar para la ejecución en cadena de la colaboración procesos de negocio capturados en la notación BPMN. Hasta donde sabemos, Caterpillar es el primer motor de ejecución de procesos basado en cadenas de bloques capaz de manejar modelos de procesos con subprocesos, así como BPMN construye como eventos límite y actividades de múltiples instancias.				

TABLA XXIII

DESCRIPCIÓN DEL ARTÍCULO D01

Código	**D01**	**Referencia**	[46]	**Año**	**2019**
Título	Combatir los videos de Deepfake usando la cadena de bloques y contratos inteligentes				
Resumen del articulo	Ofrecemos una solución y un marco general utilizando Ethereum contratos inteligentes para rastrear y seguir la procedencia y historia del contenido digital a su fuente original, incluso si el contenido se copia varias veces. El contrato inteligente utiliza hashes del Sistema de Archivos Interplanetarios (IPFS) utilizado para almacenar el contenido digital y sus metadatos. Nuestra solución se centra en el vídeo contenido, pero el marco de solución proporcionado en este documento es suficientemente genérico y puede aplicarse a cualquier otra forma de contenido.				
Información relevante	Los usuarios deben tener acceso a una procedencia de datos fiable del contenido digital, y poder rastrear un artículo en la historia para probar su originalidad y autenticidad. Este mecanismo puede ayudar a los usuarios de siendo engañados o atraídos a creer en un contenido digital falso. Las soluciones actuales están disponibles para probar la autenticidad de de arte físico (y no digital).				
Conclusiones relevantes	Las cadenas de bloques prueban la autenticidad de los videos digitales en los que se rastreabilidad confiable hasta el creador o la fuente del video original puede se establezcan, de manera descentralizada. Nuestra solución hace que uso de un sistema de almacenamiento descentralizado IPFS, nombre de Ethereum y un sistema de reputación descentralizado.				

TABLA XXIV

DESCRIPCIÓN DEL ARTÍCULO D08

Código	**D08**	**Referencia**	[45]	**Año**	**2019**
Título	EIDM: Un usuario de la nube basado en el etéreo. Protocolo de gestión de la identidad				
Resumen del articulo	Se propuso un protocolo de gestión de la identidad basado en la cadena de bloqueo del etéreo, seguido de un establecimiento de un simple marco de sistema de gestión de crédito. El nuevo protocolo es una versión mejorada del CIDM (Gestión de Identidad Consolidada), denominado protocolo EIDM (Gestión de Identidad Basada en el Etéreo). En el protocolo mejorado, JWT (JSON Web Token) en OAuth 2.0 fue usado para introducir contratos inteligentes en el protocolo EIDM, y el sistema de gestión de crédito se añadió al sistema para que pueda proporcionar un protocolo de autenticación de identidades creíbles para los usuarios y proveedores de servicios de la nube				
Información relevante	La gestión de la identidad utiliza generalmente uno de los tres tipos para gestionar la información personal de un usuario: el primero es una pieza de información que tanto el usuario como el servicio proveedor saben, como establecer una contraseña; el segundo es información que el usuario entiende y que identifica de la gestión. El tercero es para las características de identidad del usuario, como la huella dactilar del usuario, el iris, etc.				
Conclusiones relevantes	Nuestra solución es dejar que los usuarios de la nube se den cuenta su gestión de identidad a través de la tecnología Ethereum. La razón por la que la cadena de bloqueo del Etéreo se utiliza para reemplazar el anterior sistema de IDMs es porque nuestro sistema tiene el registro de la cadena de bloqueo de Ethereum de hacer que los resultados de nuestras operaciones transparentes, y a través de JWT y criptografía.				

TABLA XXV

DESCRIPCIÓN DEL ARTÍCULO D06

Código	**D06**	**Referencia**	[50]	**Año**	**2019**
Título	Aplicación integrada de la cadena de bloques en la información eléctrica. Sistema de gestión				
Resumen del articulo	En este documento, describimos las aplicaciones de la tecnología de cadenas de bloques en el Sistema de Información de Gestión Eléctrica. En primer lugar, se introducen los componentes y la estructura del marco de la cadena de bloques. Luego, la autenticación basada en la cadena de bloques, se estudia la aplicación para integrarla en la infraestructura informática existente. Por último, el beneficio y las limitaciones de la propuesta de integración en el marco se analizan como aplicación industrial.				
Información relevante	Desde 2008, junto con el desarrollo del BTC, la cadena de bloques ha atraído cada vez más atención de diferentes aspectos. Se ha aplicado en muchos campos de aplicación, desde la IO hasta la trazabilidad de los alimentos. Puede proporcionar una solución descentralizada marco para registrar los datos de manera verificable e innegable, lo cual es necesario para muchas aplicaciones en el EMIS, como la autenticación del usuario, la transmisión de datos, etc.				
Conclusiones relevantes	En este documento, creamos un servicio de cadena de bloques integrado para apoyar diferentes tipos de aplicaciones en el EMIS. El marco propuesto se implementa en una red de tres nodos e incluye diez unidades de despliegue que trabajan juntas para suministrar un servicio de cadena de bloques unificado desde la autenticación hasta la transmisión de datos.				

TABLA XXVI

DESCRIPCIÓN DEL ARTÍCULO D12

Código	**D12**	**Referencia**	[54]	**Año**	**2019**
Título	Sistema de reputación de los nodos de la niebla pública de IoT. Una solución descentralizada usando Ethereum Blockchain				
Resumen del articulo	Este documento propone un modelo de confianza descentralizado para mantener la reputación de los nodos de niebla disponibles. La reputación se mantiene teniendo en cuenta las opiniones de los usuarios sobre sus interacciones anteriores con los nodos de niebla públicos. El modelo de confianza propuesto está diseñado usando la cadena de bloqueo del Etéreo público y contratar tecnologías a fin de permitir el suministro descentralizado de servicios fiables entre los dispositivos de IO y nodos de niebla públicos. El enfoque propuesto se pone a prueba y se evalúa en términos de seguridad, rendimiento y costo. Los resultados muestran que el uso de cadenas de bloqueo para la gestión descentralizada de la reputación podría convertirse en más ventajoso cuando se compara con los modelos de confianza centralizados existentes.				
Información relevante	La gran cantidad de datos generados por la IO los nodos de fog aseguran un mejor rendimiento reduciendo latencia y optimizando la conexión entre los clientes de IO y sus proveedores de servicios. Los nodos de niebla suelen residir muy cerca de los dispositivos de IO (típicamente en los routers cercanos o interruptores). Proporcionan una conexión fiable para el proceso, lter y almacenar los flujos de datos de IOT antes de enviarlos a los proveedores de servicios en la nube.				
Conclusiones relevantes	La solución ha sido optimizada para asegurar un costo mínimo, y fue probado en el IDE Remix usando la solidez. El diagrama de relación de entidades, diagrama de secuencia y los algoritmos se proporcionaron si se necesitaban para manipulación para aplicar la solución en otras aplicaciones. La cadena de bloqueo del etéreo apoya la descentralización y asegura nuestra solución está validada, inmutable y segura.				

TABLA XXVII

DESCRIPCIÓN DEL ARTÍCULO D07

Código	**D07**	**Referencia**	[51]	**Año**	**2020**
Título	Monetización de los servicios prestados por Public Fog. Nodos que utilizan blockchain y contratos inteligentes				
Resumen del articulo	Este documento presenta un novedoso esquema para permitir la monetización basada en cadenas de bloques y el pago automatizado en criptografía de los servicios prestados por los nodos de niebla públicos. El esquema propuesto es descentralizado, fiable, automatizada, y con ciertas garantías de calidad de servicio, satisfacción del cliente, y resolución de disputas a través de un sistema de reputación. La solución propuesta utiliza la cadena de bloqueo Ethereum y sus características nativas de contrato inteligente para gobernar las interacciones entre los dispositivos y los nodos de niebla. La solución propuesta se aplica, se prueba y se evaluado para mostrar el comportamiento y la funcionalidad correctos. También proveemos análisis de costo y seguridad y mostramos que nuestra solución es resistente a los principales ataques de seguridad.				
Información relevante	La computación de la niebla extiende los servicios de computación en la nube de infraestructuras de red central a las instalaciones de los clientes utilizando variedad de nodos de niebla, a saber, interruptores inteligentes, micro centros de datos, nubes, y servidores de borde móvil proximal, por nombrar unos cuantos.				
Conclusiones relevantes	Usamos contratos inteligentes para automatizar los pagos, mantener la reputación de la niebla y para permitir que los clientes seleccionen sus nodos preferidos. Los contratos inteligentes aseguraron la transparencia y mayor nivel de confianza entre los proveedores de nodos de niebla públicos y sus clientes.				

TABLA XXVIII

DESCRIPCIÓN DEL ARTÍCULO D09

Código	**D09**	**Referencia**	[52]	**Año**	**2020**
Título	Sobre el diseño de un modelo de delegación flexible para el Internet de las cosas que usan Blockchain				
Resumen del articulo	En este documento, proponemos una delegación sin identidad, asincrónica y descentralizada modelo para la IO basado en la tecnología de cadenas de bloques. Nosotros describir los componentes del sistema, la arquitectura y los aspectos clave relacionados con la seguridad del sistema. Utilizamos atributos para validar una entidad en lugar de depender de identidades únicas. Demostramos la viabilidad de nuestro modelo a través de ejemplos de casos de uso y analizar la actuación con una prueba de implementación del banco de pruebas de conceptos utilizando el Etéreo.				
Información relevante	La transformación está muy extendida tanto en los dominios de aplicación y este paradigma es comúnmente conocido como el Internet de las cosas (IoT). Se predice que habrá 50 mil millones de dispositivos para el año 2020, y esto significa el crecimiento del uso de sistemas de IO y sus posibles dominios de aplicación a escala. Aunque tal convergencia del mundo digital y físico puede mejorar la experiencia del usuario de muchas maneras (por ejemplo, el transporte inteligente, la salud, ciudad inteligente, etc.)				
Conclusiones relevantes	Hemos proporcionado un prototipo de implementación del sistema, se discutió detalladamente componentes arquitectónicos y exploró la comunicación. Usamos la cadena de bloqueo privada Ethereum para demostrar la viabilidad de nuestro modelo.				

TABLA XXIX

DESCRIPCIÓN DEL ARTÍCULO D03

Código	**D03**	**Referencia**	[48]	**Año**	**2018**
Título	Prueba de entrega de activos digitales utilizando cadena de bloques y contratos inteligentes				
Resumen del articulo	En este documento, proponemos una solución basada en una cadena de bloques para la prueba de entrega de los activos digitales, incluyendo videos, fotos, libros electrónicos, artículos, cualquier medio digital, etc. El enfoque principal de nuestra solución es eliminar la necesidad de un tercero de confianza y para erradicar la cuestión de la confianza de los actuales sistemas de PoD.				
Información relevante	La nueva tecnología digital se encuentra ahora en mercados que ofrecen productos digitales a consumidores que se interesan por el contenido entregado por otros proveedores digitales. De acuerdo con las estadísticas, los ingresos de los EE.UU. de los medios digitales por sí solos son 43.200 millones de dólares. Con el fin de cumplir el aumento de la demanda, los mercados digitales ofrecen un espacio para tanto a los proveedores como a los consumidores para que se conecten entre sí.				
Conclusiones relevantes	En este trabajo, hemos presentado una solución basada en cadenas de bloques para el PoD de los activos digitales. Nuestra solución proporciona la manipulación registros de prueba que simplifican el seguimiento de eventos y tareas. La solución utiliza un token único para que cada cliente acceda al contenido del servidor.				

TABLA XXX

DESCRIPCIÓN DEL ARTÍCULO D16

Código	**D16**	**Referencia**	[41]	**Año**	**2018**
Título	RBAC-SC: Control de acceso basado en roles usando Smart Contract				
Resumen del articulo	En este documento, presentamos un control de acceso basado en roles utilizando contrato (RBAC-SC), una plataforma que hace uso de la tecnología de contrato inteligente de Ethereum para realizar una utilización de los roles. El etéreo es una plataforma de cadena de bloqueo abierta que está diseñada para ser segura, adaptable y flexible. Es pionera en contratos inteligentes, que son aplicaciones descentralizadas que sirven como "agentes autónomos" que funcionan exactamente como se ha programado y se despliegan en una cadena de bloques. El RBAC-SC utiliza contratos inteligentes y tecnología de cadena de bloques como infraestructuras versátiles para representar la confianza y el respaldo relación que son esenciales en RBAC y para realizar un protocolo de autenticación de desafío-respuesta que verifica la propiedad de los roles de un usuario. Describimos el marco RBAC-SC, que está compuesto de dos partes, a saber, el contrato inteligente y el protocolo de desafío-respuesta, y presentar un análisis de rendimiento.				
Información relevante	En el RBAC, los usuarios están asociados con roles, y los papeles están asociados a los servicios. Muchas organizaciones y las empresas utilizan ese marco en sus sistemas informáticos para aplicar sus requisitos de control de acceso interno. Por ejemplo, los programadores de una empresa tienen acceso tanto a los códigos fuente del backend y del frontend, mientras que la garantía de calidad en el personal sólo tiene acceso a los códigos fuente del frontend.				
Conclusiones relevantes	El RBAC-SC proporciona un mecanismo seguro y eficiente para la creación de asignaciones de funciones de usuario y para la verificación de la propiedad de un usuario de una función. Además, muchos de los usuarios colaborativos gestión de los derechos, como la personalización y el respaldo, están naturalmente incluidos.				

TABLA XXXI

DESCRIPCIÓN DEL ARTÍCULO D04

Código	**D04**	**Referencia**	[49]	**Año**	**2019**
Título	Modo de comercio de datos basado en contratos inteligentes usando Blockchain y aprendizaje automático				
Resumen del articulo	En este documento, proponemos una solución del modo de comercio de datos basado en el contrato inteligente usando cadena de bloqueo y aprendizaje de la máquina. El diseño e implementación del contrato inteligente de comercio de datos logró con éxito el objetivo de eliminar el tercero de confianza en el comercio de datos, por lo que el problema de que el centro de comercio de datos tiene la capacidad de los datos en el proceso de la negociación de datos se resuelve. Este documento presenta el conjunto proceso de contrato inteligente desde el diseño, la implementación hasta la finalización de la prueba, y proporciona el análisis de seguridad y evaluación del desempeño.				
Información relevante	Con el fin de satisfacer el crecimiento de las demandas de datos, los centros de comercio de datos proporcionan los propietarios de los datos y los compradores de datos con el espacio interconectado. Sin embargo, los datos tienen su particularidad, es decir, no hay unicidad, no hay una clara propiedad de limitaciones; una vez visto, está la propiedad; la replicación de los datos es completamente indiferenciado				
Conclusiones relevantes	Nuestra solución puede ser usada para resolver el problema de que el centro de comercio de datos tiene la capacidad de retener los datos en los datos tradicionales, modo de comercio, a fin de proteger los derechos y intereses del propietario de los datos y promover el desarrollo del comercio de datos.				

TABLA XXXII

DESCRIPCIÓN DEL ARTÍCULO D10

Código	**D10**	**Referencia**	[42]	**Año**	**2019**
Título	Control de acceso basado en contratos inteligentes para el Internet de las cosas				
Resumen del articulo	Este documento investiga un tema crítico de control de acceso en el Internet de las cosas (IoT). En particular, proponemos un marco inteligente basado en contratos, que consiste en múltiples contratos de control de acceso (ACC), contrato de un juez (JC) y un contrato de registro (RC), para lograr una distribución y confianza de control de acceso a los sistemas de IO. Cada ACC proporciona un método de control para un par sujeto-objeto, e implementa ambos la validación del derecho de acceso estático basado en políticas predefinidas y validación dinámica de los derechos de acceso comprobando el comportamiento del sujeto.				
Información relevante	Los esquemas tradicionales de control de acceso a la IO se basan principalmente en la parte superior de los modelos de control de acceso conocidos, incluyendo el modelo de control de acceso basado en roles (RBAC), el modelo de control de acceso basado en atributos, el modelo de control de acceso (ABAC) y el modelo de modelo de control de acceso (CapBAC).				
Conclusiones relevantes	Este trabajo investigó el tema del control de acceso en la IO, para lo cual propusimos un marco inteligente basado en contratos para implementar un control de acceso distribuido y confiable. El marco incluye contratos de control de acceso múltiple (ACC) para el control de acceso de múltiples pares sujeto-objeto en el sistema, un contrato de un juez (JC) para juzgar el mal comportamiento de los sujetos durante el control de acceso, y un registro contrato (RC) para la gestión de los ACC y JC.				

TABLA XXXIII

DESCRIPCIÓN DEL ARTÍCULO D14

Código	**D14**	**Referencia**	[56]	**Año**	**2018**
Título	Sistema inteligente de revisión basado en contratos para una IoT. Mercado de datos				
Resumen del articulo	Proponemos un sistema de revisión que puede confirmar la reputación de un dato o los datos comercializados en el mercado de datos P2P. Los sistemas tradicionales de revisión servidor-cliente tienen muchos inconvenientes, como la vulnerabilidad de seguridad o el comportamiento malicioso del administrador del servidor. Sin embargo, el sistema de revisión desarrollado en este estudio se basa en contratos inteligentes Ethereum; por lo tanto, este sistema funciona en la red P2P y es más flexible para el problema de la red. Además, la integridad e inmutabilidad de las revisiones registradas están aseguradas debido al público de la cadena de bloqueo libro de cuentas.				
Información relevante	Los dispositivos basados en la Internet de las Cosas (IO), especialmente los utilizados para la automatización del hogar, consisten en sus propios sensores y generan muchos registros durante un proceso. Las empresas que producen dispositivos de IO convertir estos datos de registro en datos más útiles mediante un procesamiento secundario; por lo tanto, requieren datos de los usuarios del dispositivo.				
Conclusiones relevantes	El mercado de datos de IO es una plataforma para compartir datos de IO en la que se realizan transacciones entre datos los consumidores, que necesitan datos, y los propietarios de datos, que quieren vender sus datos. En los últimos años, muchos estudios han considerado un modelo de mercado P2P en lugar del tradicional mercado orientado a los servidores que tiene muchos inconvenientes.				

TABLA XXXIV

DESCRIPCIÓN DEL ARTÍCULO D11

Código	**D11**	**Referencia**	[53]	**Año**	**2020**
Título	Usando la cadena de bloqueo Ethereum para almacenar y consultar los datos farmacogenómicos a través de la tecnología de contratos inteligente				
Resumen del articulo	Diseñamos un contrato inteligente específico para almacenar y consultar las interacciones gen-droga en el Etéreo basado en un índice y un enfoque de mapas múltiples. Nuestro contrato almacena cada observación farmacogenómica, una droga de variante genética trillizo con resultado, en un mapa que se puede buscar por un identificador único, lo que permite un almacenamiento eficiente en tiempo y espacio y consulta. Esta solución se clasificó entre las tres primeras en la competencia IDASH 2019.				
Información relevante	Es necesario adoptar protocolos de almacenamiento e intercambio de datos. Una opción prometedora para el almacenamiento seguro y de alta integridad y compartir es un contrato inteligente de Ethereum. El Etéreo es una plataforma de cadena de bloques, y los contratos inteligentes son piezas inmutables de código que se ejecuta en máquinas virtuales en esta plataforma y que puede ser invocado por un usuario u otro contrato (en la cadena de bloqueo de la red).				
Conclusiones relevantes	Demostramos que los datos farmacogenómicos pueden ser almacenados y consultados eficientemente usando Ethereum cadena de bloqueo. Nuestras soluciones podrían utilizarse para almacenar una serie de datos clínicos y extenderse a otros campos que requieren un almacenamiento de datos de alta integridad y un acceso eficiente.				

Anexo 2: Proceso de registro en el Smart Contract

A continuación, se muestra el proceso para agregar estudiantes utilizando el contrato inteligente, la blockchain, y la interfaz, además se muestra el resultado de las transacciones.

Estudiante 1

Proceso Administrador

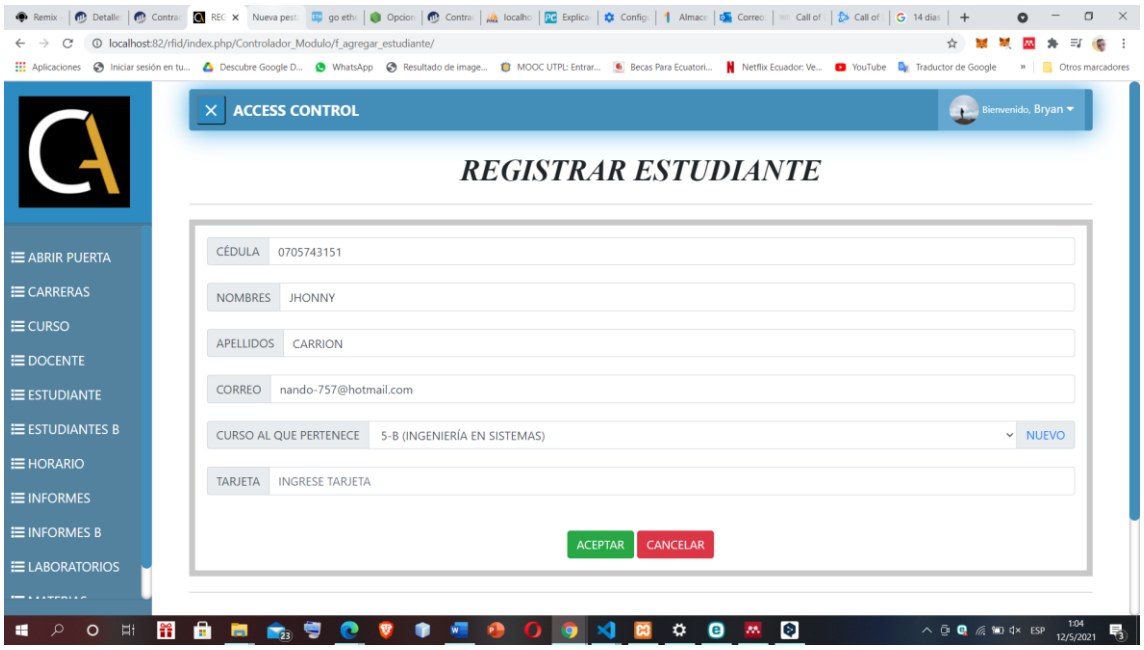


Fig. 55 formulario del sistema web para registrar estudiantes lo realiza el administrador (estudiante 1).

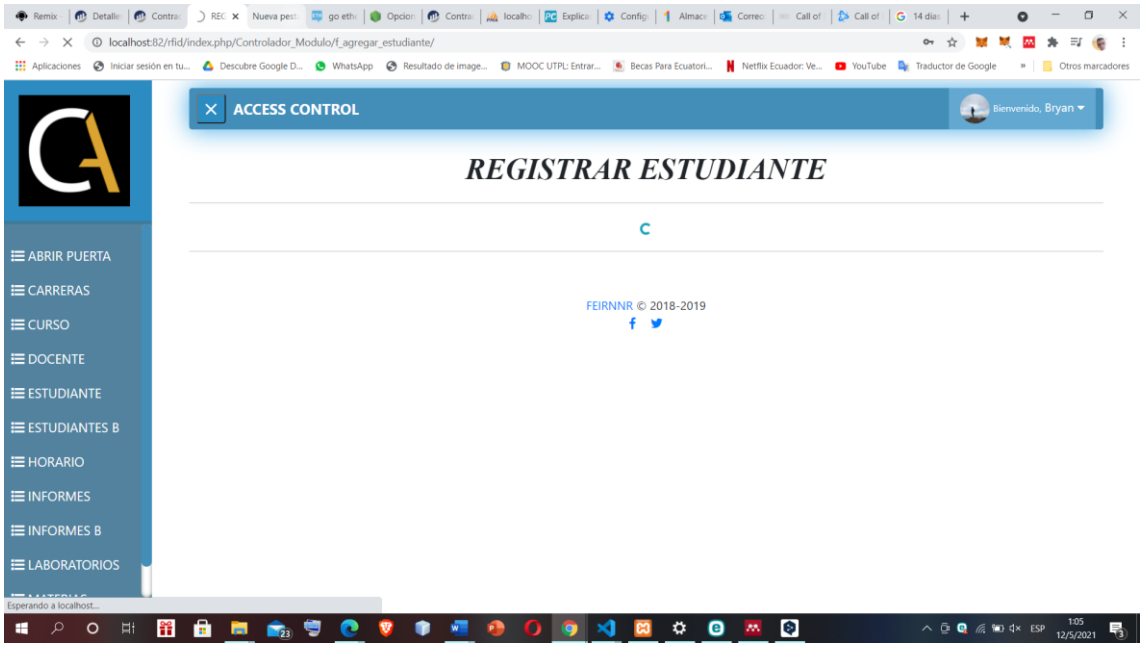


Fig. 56 Estado de cargando al presionar “Aceptar” el registro del estudiante mientras se envía la información (estudiante 1).

Fig. 57 Estado de estudiante mientras se espera la confirmación por correo electrónico guardado temporal (estudiante 1).

Proceso Estudiante

Fig. 58 Ventana de confirmación del estudiante para ingresar la cuenta de la billetera y la clave privada (estudiante 1).

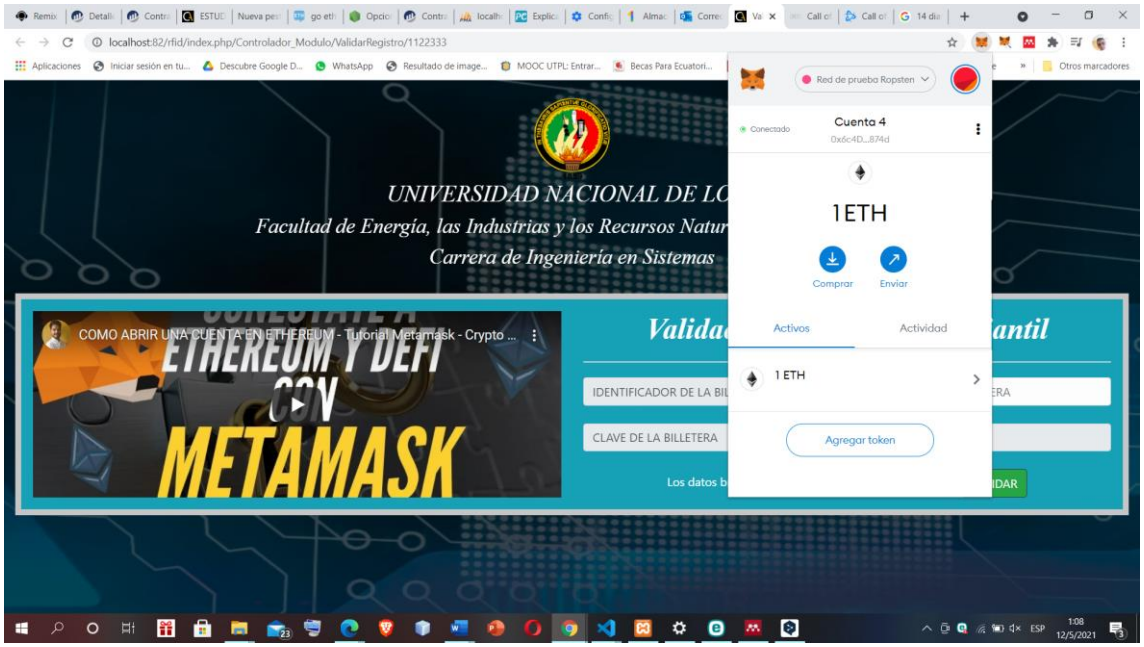


Fig. 59 Uso de MetaMask para obtener los datos personales necesarios (estudiante 1).

Fig. 60 Campos llenos con la información de la cuenta para enviar a validar (estudiante 1).

Fig. 61 Envió de la información de la cuenta para terminar el registro (estudiante 1).

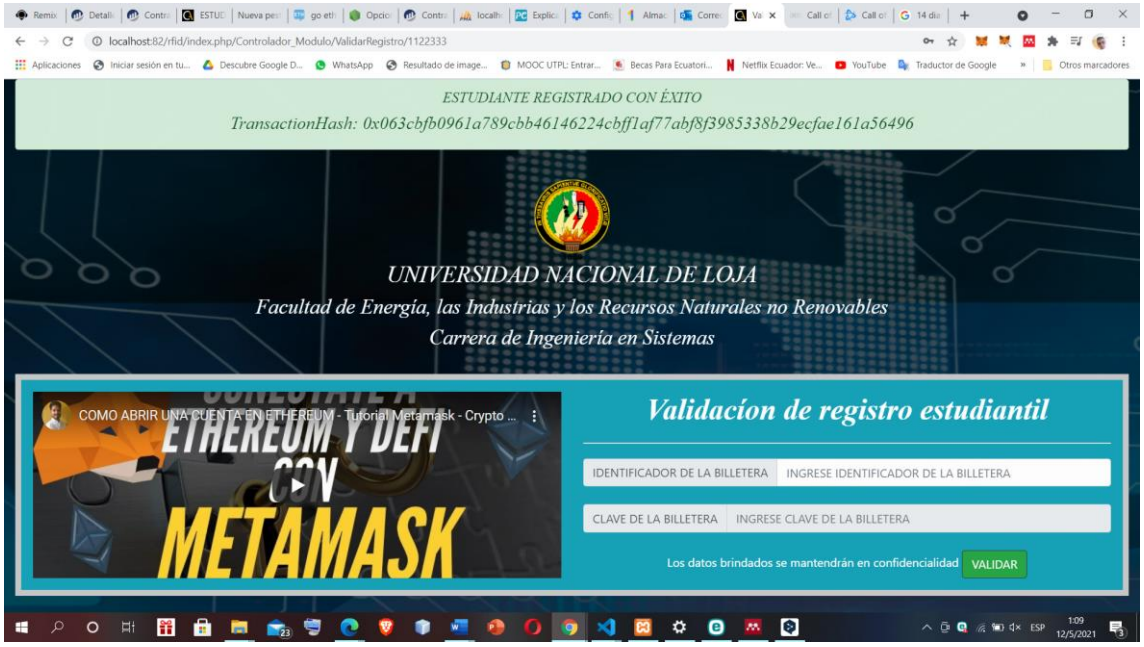


Fig. 62 Transacción realizada con éxito muestra el hash de transacción en caso de visualizar el proceso (estudiante 1).

Proceso Administrador

Fig. 63 Información en modulo que se muestra después de la confirmación (estudiante 1).

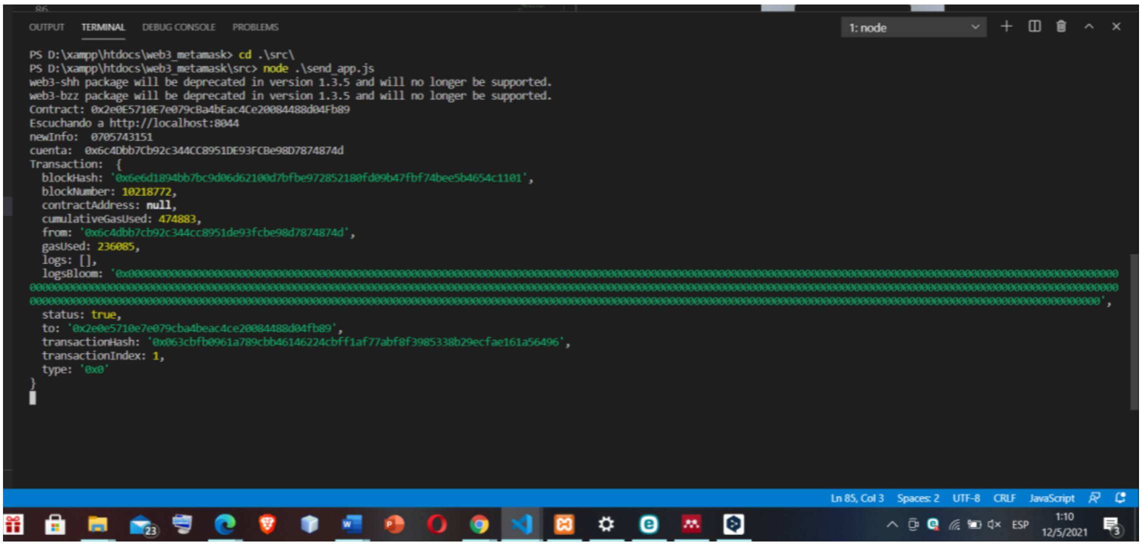


Fig. 64 Información de la transacción enviada a la Blockchain y bloque en el que fue minada (estudiante 1).

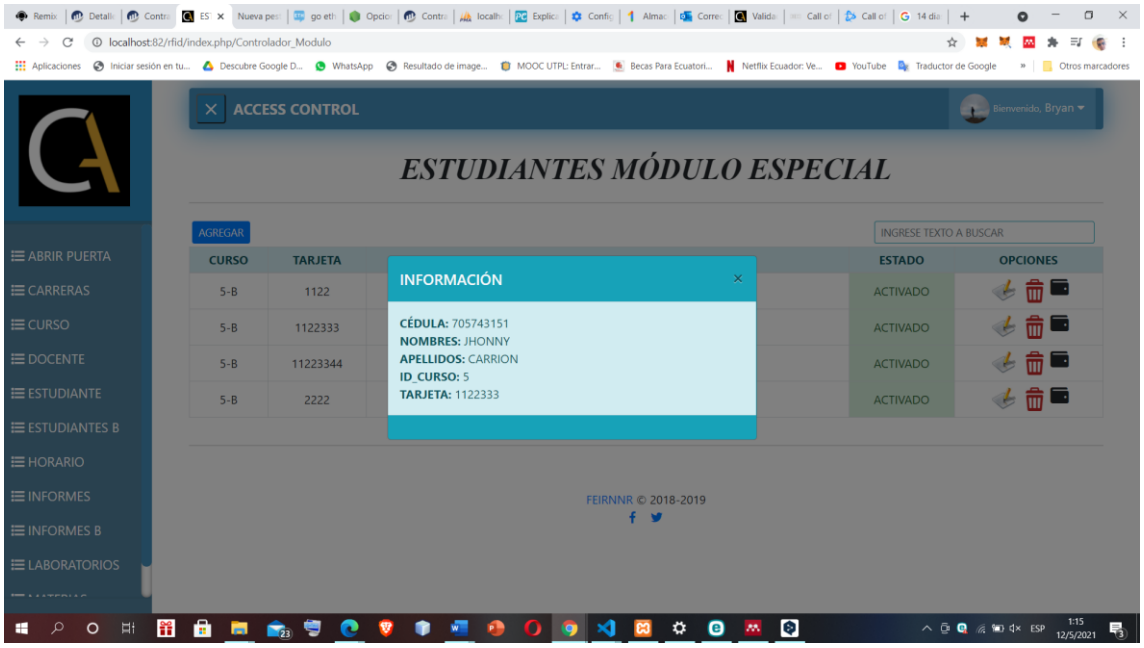


Fig. 65 Se muestra la información del estudiante al presionar el botón “BILLETERA” extraída de la Blockchain (estudiante 1).

Estudiante 2

Proceso Administrador

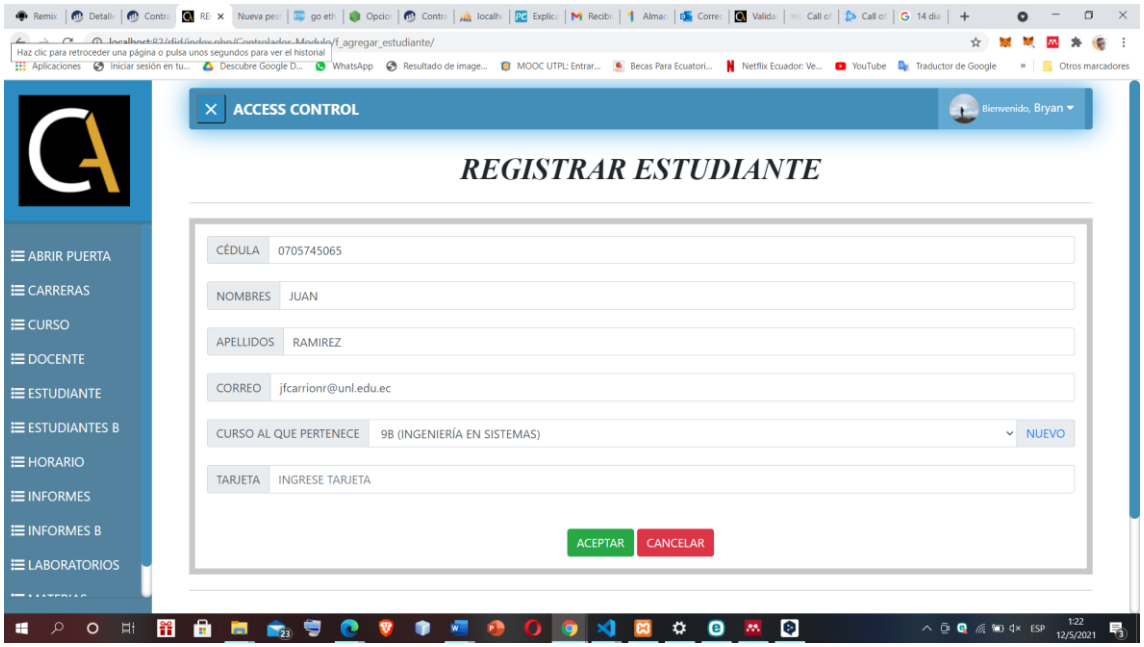


Fig. 66 formulario del sistema web para registrar estudiantes lo realiza el administrador (estudiante 2).

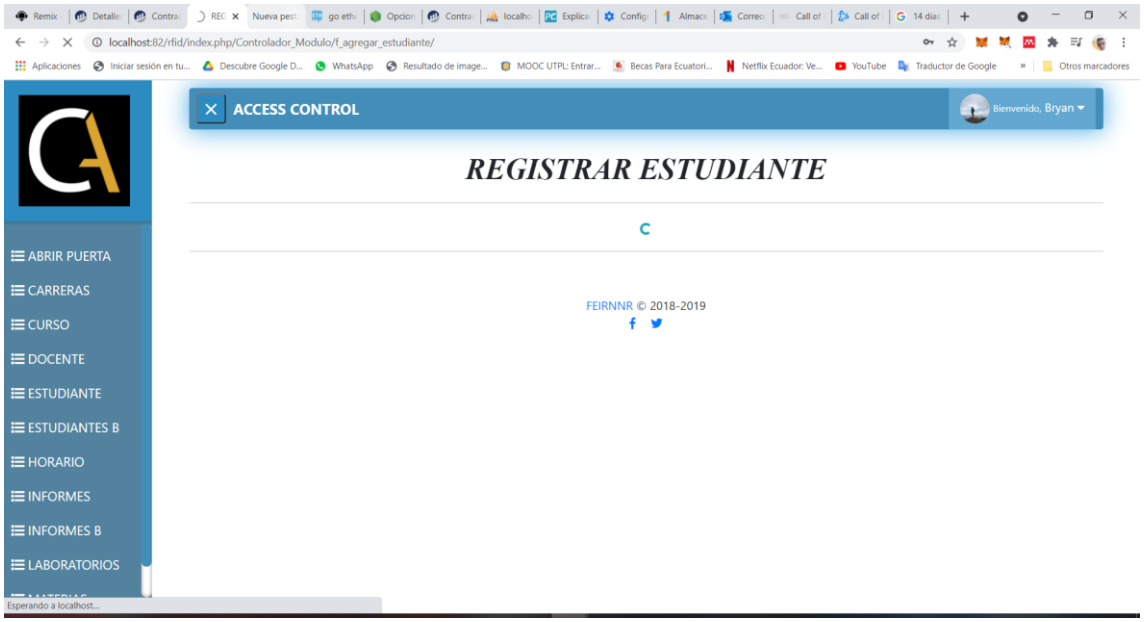


Fig. 67 Estado de cargando al presionar “Aceptar” el registro del estudiante mientras se envía la información (estudiante 2).

Fig. 68 Estado de estudiante mientras se espera la confirmación por correo electrónico guardado temporal (estudiante 2).

Proceso Estudiante

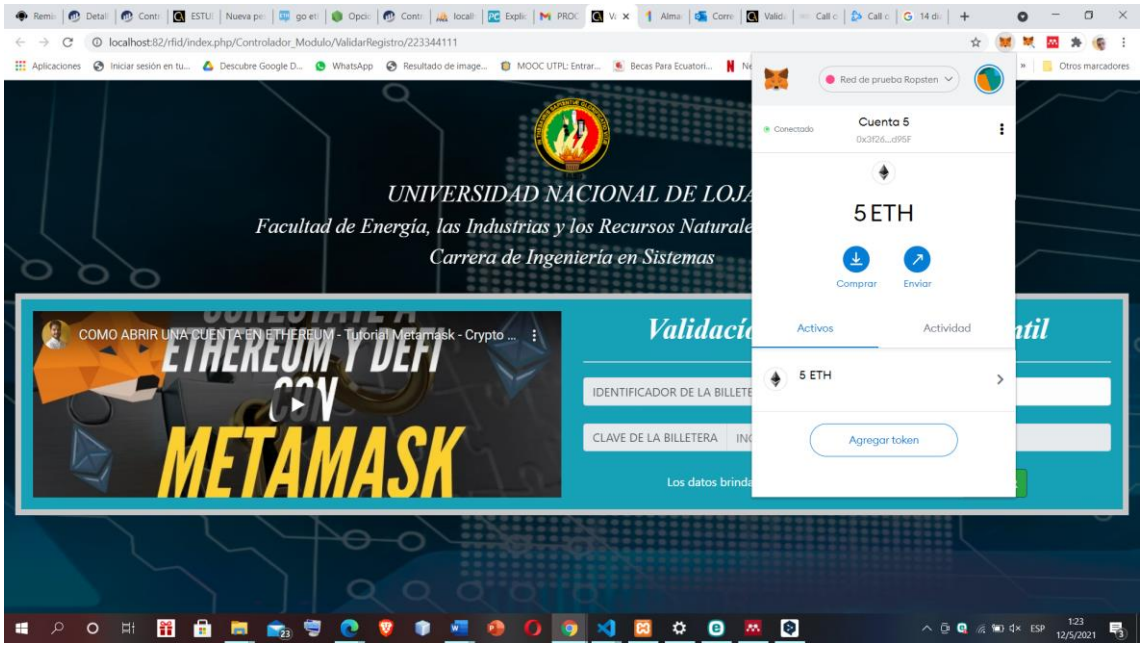


Fig. 69 Uso de MetaMask para obtener los datos personales necesarios (estudiante 2).

Fig. 70 Campos llenos con la información de la cuenta para enviar a validar (estudiante 2).

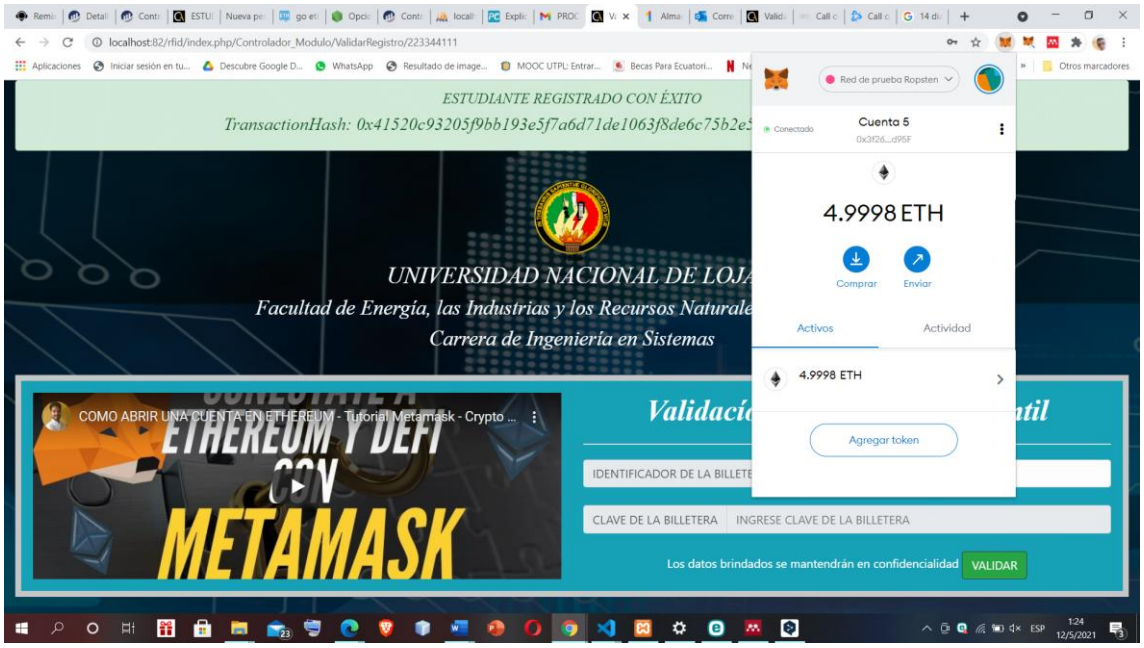


Fig. 71 Transacción realizada con éxito muestra el hash de transacción en caso de visualizar el proceso y el débito en la cuenta (estudiante 2).

Proceso Administrador

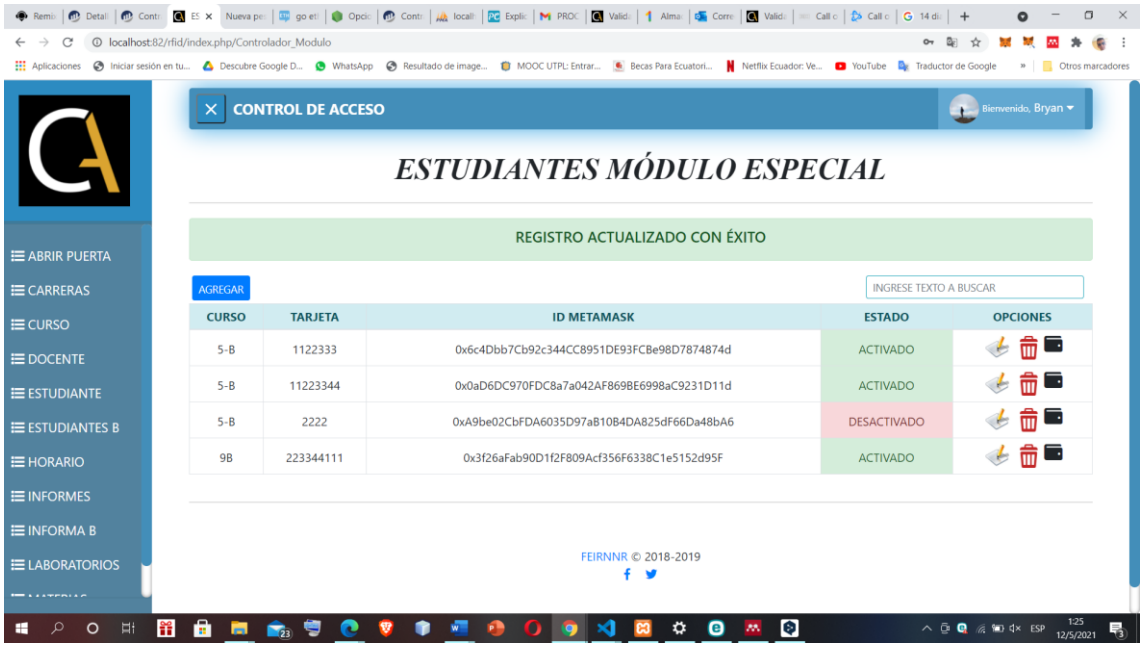


Fig. 72 Información en modulo que se muestra después de la confirmación (estudiante 2).

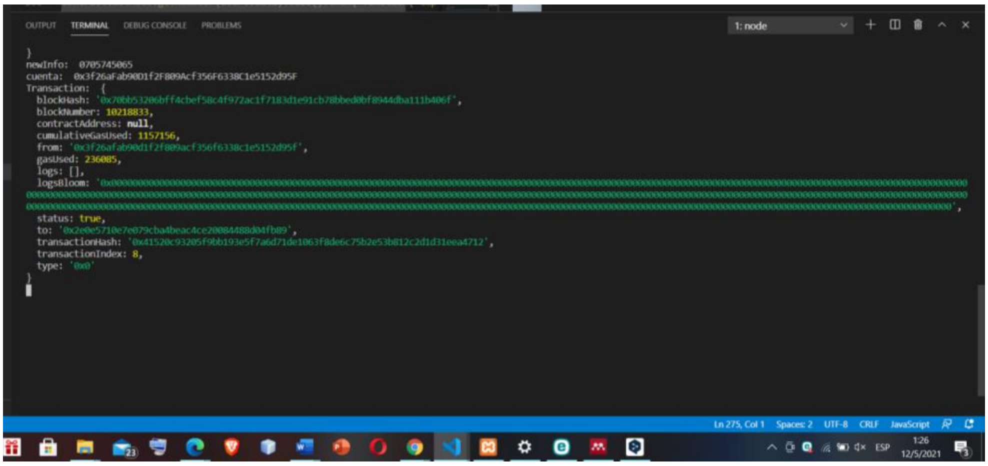


Fig. 73 Información de la transacción enviada a la Blockchain y bloque en el que fue minada (estudiante 2).

Excepciones

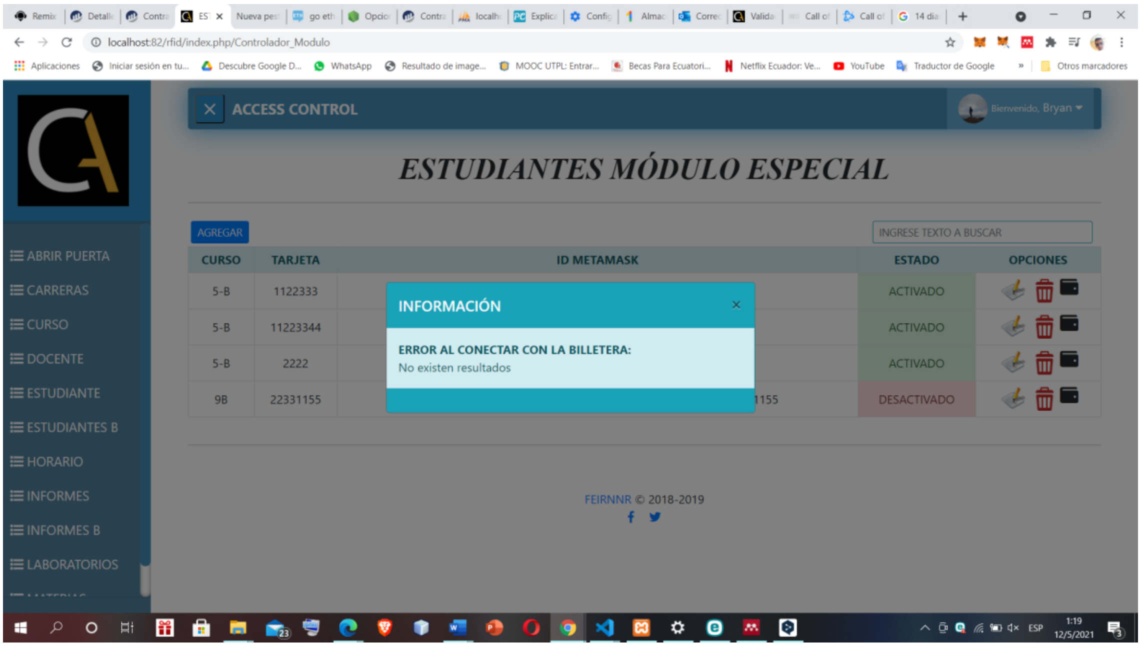


Fig. 74 Mensaje de error al ver la información de un estudiante que aún no ha confirmado su registro.

Anexo 3: Revisión sistemática

A continuación, se presenta el artículo de la aplicación de la revisión sistemática.

Revisión Sistemática de Literatura: Análisis de Módulos para Registro de Usuarios aplicando Smart Contracts

Carrión-Ramírez Jhonny Fernando, Cristian Narváez.
Carrera de Ingeniería en Sistemas
Universidad Nacional de Loja, Loja-Ecuador
jfcarrionr@unl.edu.ec, Cristian.narvaez@unl.edu.ec

***Resumen*— La seguridad y aplicación de nuevas tecnologías para no siempre será fácil integrarlas, debido a diferentes características en especial que beneficios me ofrecerá que costo tendrá entre otras, debido a esto la búsqueda de seguridad siempre será un punto difícil de alcanzar.**

En la presente Revisión Sistemática de Literatura (RSL), se busca la forma de desarrollar contratos inteligentes dentro del modulo de registro de usuarios de un sistema, además de detectar cuales son las herramientas que brindaran una mejor aplicación de los contratos. Se aplico la metodología de Bárbara Kitchenham con todas sus fases y etapas, lo cual definió como una de las herramientas mas usadas para el desarrollo, las pruebas y depuración de contratos inteligentes a Remix IDE, además esta revisión nos brindó pautas para el desarrollo de los contratos en diferentes ámbitos como en la nube y niebla y en sistemas de almacenamiento.

***Abstract*– The security and application of new technologies will not always be easy to integrate them, due to different characteristics, especially the benefits they will offer and their cost, among others, due to this, the search for security will always be a difficult point to achieve.**

In the present Systematic Literature Review (SLR), the way to develop smart contracts within the user registration module of a system is sought, in addition to detecting which are the tools that will provide a better application of the contracts. Barbara Kitchenham methodology was applied with all its phases and stages, which defined Remix IDE as one of the most used tools for the development, testing and debugging of smart contracts. This review also provided us with guidelines for the development of contracts in different environments such as cloud, fog and storage systems.

I. Introducción

En la actualidad la seguridad de los sistemas informáticos es de mucha importancia , y la implementación de nuevas tecnologías lo hace cada día más eficiente. Los contratos inteligentes[1] en conjunto con la tecnología Blockchain brindan una nueva de propuesta de seguridad dentro de estos sistemas.
Mediante el manejo de sistemas seguros y descentralizados aplicando el uso de la tecnología emergente blockchain, se puede ofrecer al propietario de una cuenta, seguridad al registrar y almacenar sus datos privados de conocimiento único, previniendo el robo de información, uso de servicios y el acceso ilegal a los recursos existentes dentro de una cuenta por terceros[1].

Dentro de la presente RSL, se aplico las fases que presenta la metodología de Bárbara Kitchenham se obtuvo los estudios y el conocimiento necesario para determinar el desarrollo de contratos inteligentes, así como de las herramientas y lenguaje de programación a usar.

El estudio a continuación se realizó por secciones que son: el Alcance, que es el que determina razón de realizar la RSL, la Metodología donde se definió el proceso establecido por Bárbara Kitchenham dentro de las revisiones, los Resultados donde se encuentra aplicada la RSL en base a las fases de la metodología y finalmente las Conclusiones en base a la investigación realizada.

II. Alcance

En la presente Revisión Sistemática de Literatura tiene como fin investigar y analizar los estudios que se relacionan al desarrollo e implementación de contratos inteligentes dentro del modulo de registro de usuarios, para así poder identificar las herramientas y pautas necesarias para integrar estos contratos a el registro de usuarios de un sistema.

III. Metodología

Para el desarrollo y realización de la revisión sistemática de literatura se sustentó en la metodología de Bárbara Kitchenham [2], la cual describe su proceso en tres fases que son: Planificación de la revisión, Realización de la revisión y Presentación de informes como se lo presenta en la TABLA I a continuación:

[1]Contratos inteligentes: Según [22], es un protocolo de computadora destinado a facilitar, verificar o hacer cumplir digitalmente la negociación o ejecución de un contrato

TABLA I
PROCESO DE REVISIÓN SISTEMÁTICA DE LITERATURA

Fases	Etapas
Planificación de la Revisión	1) Identificación de la necesidad de una revisión.
	2) Formulación de las preguntas de investigación.
	3) Desarrollo de un protocolo de revisión.
Realización de la Revisión	1) Identificación de la investigación.
	2) Selección de estudios primarios.
	3) Evaluación de calidad del estudio.
	4) Extracción de datos y monitoreo.
	5) Síntesis de datos.
Presentación de informe	1) Discusión y análisis

IV. RESULTADOS

A. Planificación de la revisión

1) Identificación de la necesidad de una revisión

La implementación de una revisión sistemática de literatura, permite analizar conocimiento que abarca a un cierto tema de investigación, dentro de nuestro caso de estudio permitió elegir todos los requerimientos necesarios para desarrollar Smart contracts, las herramientas a implementar, el lenguaje de programación y librerías para interactuar con los Smart contracts dentro del manejo de registro de usuarios de un sistema.

2) Preguntas de investigación

Para el cumplimiento de lo anteriormente propuesto y en base a la temática en la TABLA II se presentan dos preguntas de investigación que limitan el curso de la revisión.

TABLA II
PREGUNTAS DE INVESTIGACIÓN

Preguntas de investigación	
P1	¿Qué estudios discuten acerca de los componentes para el desarrollo de contratos inteligentes en el registro y control de usuarios en EVM?
P2	¿Qué estudios presentan herramientas para el desarrollo de contratos inteligentes basados Ethereum para el registro y control de usuarios?

3) Desarrollo de un protocolo de revisión

El protocolo a implementar es el propuesto por Mark Petticrew y Helen Roberts[3], PICOC (Población, Intervención, Comparación, Resultado y Contexto), el cual ayuda a definir la estructura de las cadenas de búsquedas, además con la ayuda de la herramienta Parsifal se puede proceder a la revisión con mayor facilidad.

a) *Fuentes Bibliográficas*

Para realizar búsqueda y selección de los artículos, se procedió a usar bases de datos científicas las cuales son:

- IEEE Digital Library (https://ieeexplore.ieee.org/)
- Science@Direct (https://www.sciencedirect.com/)
- Scopus (http://www.scopus.com)

b) *Selección de Palabras Clave*

Se determina las palabras claves en base a algunos artículos científicos que se detallan en la TABLA III.

TABLA III
ARTÍCULOS Y PALABRAS CLAVES

Artículos	Palabras claves
RBAC-SC: Role-Based Access Control Using Smart Contract[4]	Blockchain technology, role-based access control, smart contracts.
A Secure Cloud Storage Framework With Access Control Based on Blockchain[5]	Cloud storage, access control, Ethereum, blockchain, smart contract.
EIDM: A Ethereum-Based Cloud User Identity Management Protocol[6]	Cloud computing, identity management, blockchain, reputation, smart contract.
Smart Contract-Based Access Control for the Internet of Things[1]	Internet of Things, access control, blockchain, smart contract.

c) *Cadenas de Búsqueda*

Cadenas de búsqueda formadas y especificas para cada biblioteca virtual se las presenta en la TABLA IV.

TABLA IV
CADENAS DE BÚSQUEDA

Bibliotecas	Cadenas
IEEE Digital Library	(((((((("All Metadata":smart contract) OR "All Metadata":contratos inteligentes) OR "All Metadata":SmartContract) OR "All Metadata":smart contracts) AND "All Metadata":login) OR "All Metadata":register users) OR "All Metadata":ethereum) NOT "All Metadata":bitcoin)
Science@Direct	(("Smart Contract" OR "Pactos Inteligentes" OR "SmartContract") AND ("ethereum" OR "EVM") AND (register% OR user%) NOT(BITCOIN))inteligentes basados Ethereum para el registro y control de usuarios?

Scopus	TITLE-ABS-KEY (("Smart Contract" OR "Pactos Inteligentes" OR "SmartContract") AND ("ethereum" OR "EVM") AND (register% OR user%) AND NOT (bitcoin)) AND (LIMIT-TO (DOCTYPE , "ar")) AND (LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016))

d) *Criterios de Inclusión*

Para que los estudios sean incluidos se tomaran en cuenta los siguientes criterios:

- Estudios publicados entre el 2016 y 2020.
- El idioma ingles será el más relevante en los documentos y también se aceptará en español.
- En el resumen deberán contener información sobre la aplicación de contratos inteligentes y el uso de herramientas dentro del mismo.
- Artículos que sean clasificados como revistas.

e) *Criterios de exclusión*

Se procedió a excluir los estudios que no cumplan con las siguientes condiciones:

- Estudios que no cumplan con los criterios anteriores
- Artículos que se clasifican como conferencias.
- Publicaciones que estén relacionadas en mayor parte con criptomonedas y bitcoin.
- Estudios parciales o que estén en desarrollo

B. Realización de la revisión

1) Identificación de la investigación

La finalidad de la revisión sistemática de literatura, es responder a las preguntas planteadas a través del análisis de los estudios que se obtengan y contribuyan con información confiable y verídica.

2) Selección de estudios primarios

En la TABLA V, se presenta los estudios que se obtuvo luego de aplicar las cadenas de búsqueda y los criterios de inclusión y exclusión en su respectiva biblioteca virtual.

TABLA V
ESTUDIOS PRIMARIOS

Bibliotecas	Encontrados	Seleccionados
IEEE Digital Library	87	52
Science@Direct	51	16
Scopus	89	58

3) Evaluación de calidad de los estudios

Para definir la calidad de cada artículo se definió las siguientes preguntas:

- **PE1:** ¿El artículo se enfoca en el desarrollo de contratos inteligentes en el módulo de registro de usuarios?
- **PE2:** ¿Describe los componentes del contrato inteligente?
- **PE3:** ¿Se identifica alguna herramienta para el desarrollo de contratos inteligentes?

4) Extracción de datos y monitoreo

Luego de todo el proceso se aceptaron 18 estudios que contienen información relacionada y se los detalla en la TABLA VI.

TABLA VI
ESTUDIOS ACEPTADOS

N.º	Titulo	Año
D01	Combating Deepfake Videos Using Blockchain and Smart Contracts[7].	2019
D02	Blockchain-Based Proof of Delivery of Physical Assets With Single and Multiple Transporters[8].	2018
D03	Proof of Delivery of Digital Assets using Blockchain and Smart Contracts[9].	2018
D04	Smart Contract Based Data Trading Mode Using Blockchain and Machine Learning[10].	2019
D05	A Secure Cloud Storage Framework With Access Control Based on Blockchain[11].	2019
D06	Integrated Application of Blockchain in the Electric Information Management System[12].	2019
D07	Monetization of Services Provided by Public Fog Nodes Using Blockchain and Smart Contracts[13].	2020
D08	EIDM: A Ethereum-Based Cloud User Identity Management Protocol[6].	2019
D09	On the Design of a Flexible Delegation Model for the Internet of Things Using Blockchain[14].	2020
D10	Smart contract-based access control for the internet of things[1].	2019
D11	Using Ethereum blockchain to store and query pharmacogenomics data via smart contracts[15].	2020
D12	IoT Public Fog Nodes Reputation System: A Decentralized Solution Using Ethereum Blockchain[16].	2019
D13	A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems[17].	2018
D14	Smart Contract-Based Review System for an IoT Data Marketplace[18].	2018
D15	Caterpillar: A business process execution engine on the Ethereum blockchain[19].	2019

D16	RBAC-SC: Role-based access control using smart contract[4].	2018
D17	Blockchain for Secure EHRs Sharing of Mobile Cloud Based E-Health Systems[20].	2019
D18	Blockchain-based decentralized reverse bidding in fog computing[21].	2020

5) Síntesis de datos

De los estudios que se aceptaron se analizo cada uno para identificar el aporte más relevante al tema y obtener los componentes para el desarrollo de contratos inteligentes dentro del modulo de registro de usuarios dependiendo del tipo de sistema.

a) ¿Qué estudios discuten acerca de los componentes para el desarrollo de contratos inteligentes en el registro y control de usuarios en EVM?

La aplicación de los contratos inteligentes tiene una gran amplitud, dado esto en la TABLA VII se presentan los estudios que dependiendo del sistema desarrollan los contratos en diferentes ámbitos dentro de Ethereum.

TABLA VII
CONTRATOS INTELIGENTES DESARROLLADOS N DIFERENTES SISTEMAS

Sistema	Estudios
Basados en la nube y niebla	D05, D12, D17
Basados en presentación de información	D06, D10, D16
Basados en almacenamiento	D13

b) ¿Qué estudios presentan herramientas para el desarrollo de contratos inteligentes basados Ethereum para el registro y control de usuarios?

Existen muchas herramientas para vincular los contratos inteligentes a sistemas actuales con módulos de registro de usuarios, las cuales tienen diferentes funcionalidades tanto para seguridad como para calidad, es así las podemos observar en la TABLA VIII detalladas por estudios.

TABLA VIII
HERRAMIENTAS PARA EL DESARROLLO DE CONTRATOS INTELIGENTES EN ETHEREUM

Funcionalidad	Estudios	Herramientas
Desarrollo, Pruebas y Depuración	D01, D02, D03, D04, D05, D07, D08, D10, D12, D14, D18	Remix IDE
Despliegue e interacciones	D05, D09, D10, D17	Ethereum Geth Client
Acceso a la red y Pruebas	D11, D12	Truffle
Compilación	D09, D15	Solc-js compilador estándar de Solidity

C. Presentación de informe

1) Discusión y análisis

Durante la etapa de revisión se observa que los diferentes artículos analizados presentan diversidad de herramientas para las diferentes etapas como son: desarrollo, despliegue, compilación y las pruebas de los contratos inteligentes dentro de la plataforma de Ethereum la cual mediante un lenguaje de programación le permite crear software para gestionar las transacciones y automatizar resaltados, estas herramientas permiten observar la simulación y el comportamiento en la red blockchain, así mismo como los costos que se efectuaran debido al consumo de gas por transacciones, además se puede destacar que una de las herramientas más usadas que trabaja en base a un navegador es Remix IDE la cual brinda una interfaz e implementa la programación con Solidity que es el lenguaje base y reconocido por Ethereum para el desarrollo de contratos inteligentes, en contraste se puede usar librerías dentro de IDEs Y otros lenguajes para realizar las algunas de las funciones o conexiones a la red blockchain.

En los estudios D01, DO2, DO3, D04, D05, D07, DO8, D10, D12, D14, D18 presentan una potente herramienta muy utilizada Remix IDE en la actualidad sirve para el desarrollo, depuración, pruebas e implementación de contratos inteligentes con base en el lenguaje solidity, al tener todas estas capacidades la convierte en ideal para implementarla en sistemas que trabajen con la red blockchain de Ethereum, además los estudios D05, D07, D08, D12, D18 están orientados a sistemas implementados en la nube(cloud) y en la niebla(fog) que utilizan esta herramienta para el desarrollo de los contratos inteligentes, estos intervienen principalmente para almacenar y recuperar datos, monitorear y medir sus servicios, además la interacción entre nodos de la blockchain forman esquemas de control de acceso distribuidos. Dentro de los estudios D04, D14 hablan sobre el comercio de datos donde debe existir seguridad una parte muy importante, ahí es donde intervienen los contratos inteligentes para verificar al propietario de estos datos y su calidad evitando la intervención de terceros, en base a lo observado, el uso de remix dentro de la plataforma Ethereum es esencial pese a que se encontró muy

poca especificación de la herramienta en los documentos, se desconoce las versiones con las que se trabajó y las especificaciones en las que intervino, aun así la capacidad de Remix es robusta para implementar en todo tipo de sistema e importante para el desarrollo de este trabajo.

En los estudios D11 y D12 emplean la herramienta Truffle muy conocida e implementada que sirve como Frond-End, además de que abarca el marco de pruebas y canalización de activos de la blockchain, en el documento D12 realizo la combinación del funcionamiento de Remix y Truffle para abarcar el desarrollo y las pruebas dentro de IOT dentro de una arquitectura den la niebla(Fog), con esto se pretende ver la amplitud de herramientas que trabajan en conjunto con Ethereum y definirlas para ser aplicadas en el trabajo, pese a que Truffle tiene muchas capacidades Remix demuestra ser más robusta y abarcar muchas fases que los contratos inteligentes conllevan y eso se demuestra en la cantidad de estudios en la que interviene .

La metodología para implementar contratos inteligentes dentro de la plataforma Ethereum y su red blockchain está definida por su documentación, se trata de un diseño por contrato, la cual toma a los elementos del diseño como participantes de una relación similar al contrato de negocios. Aunque en los estudios acoplan sus sistemas a sus propias metodologías para la arquitectura, la escritura y diseño de los contratos inteligentes está dirigida por esta metodología de diseño por contrato que es la que procederemos a aplicar para programarlo.

V. Conclusiones

En base a todo el análisis realizado de la RSL se puede concluir lo siguiente:

- En muchos de los trabajos relacionados los autores hablan sobre la aplicación de contratos inteligentes a partes del sistema, lo cual no se tiene una referencia de la implicación a un sistema completo que funcione con ellos.
- La metodología para implementar los contratos inteligentes no se encuentra explicada en ninguno de los trabajos, pero en base a la plataforma Ethereum y su documentación, se conoce como Diseño por Contrato, la cual toma a los elementos del diseño como participantes de una relación similar al contrato de negocios. Aunque en los estudios acoplan sus sistemas a sus propias metodologías para la arquitectura, la escritura y diseño de los contratos inteligentes está dirigida por esta metodología de diseño por contrato que es la que procederemos a aplicar para programarlo.
- Se ha concluido que dentro de las herramientas mas usadas para la programación de contratos inteligentes se encuentran Remix IDE y Truffle donde las funcionalidades que nos permiten abarcan las necesidades para cumplir los objetivos de implementar contratos inteligentes en sistemas actuales.
- Para la interacción y pruebas de los contratos inteligentes hacen falta mayor información, ya que para realizarlo necesitamos la intervención de una herramienta mas llamada MetaMask la cual no se presenta en los estudios realizados.

VI. Referencias

[1] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the internet of things," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1594–1605, 2019.

[2] B. Kitchenham, "Procedures for Performing Systematic Literature Reviews," *Jt. Tech. Report, Keele Univ. TR/SE-0401 NICTA TR-0400011T.1*, vol. 33, p. 33, 2004.

[3] *Systematic Reviews in the Social Sciences*. 2006.

[4] J. P. Cruz, Y. Kaji, and N. Yanai, "RBAC-SC : Role-based Access Control using Smart Contract," vol. 3536, no. c, 2018.

[5] S. Wang, X. Wang, and Y. Zhang, "A Secure Cloud Storage Framework With Access Control Based on Blockchain," *IEEE Access*, vol. 7, pp. 112713–112725, 2019.

[6] S. Wang, R. Pei, and Y. Zhang, "EIDM: A Ethereum-Based Cloud User Identity Management Protocol," *IEEE Access*, vol. 7, pp. 115281–115291, 2019.

[7] H. R. Hasan and K. Salah, "Combating Deepfake Videos Using Blockchain and Smart Contracts," *IEEE Access*, vol. 7, no. c, pp. 41596–41606, 2019.

[8] H. R. Hasan and K. Salah, "Blockchain-based Proof of Delivery of Physical Assets with Single and Multiple Transporters," *IEEE Access*, vol. PP, no. 8, p. 1, 2018.

[9] H. R. Hasan and K. Salah, "Proof of Delivery of Digital Assets Using Blockchain and Smart Contracts," *IEEE Access*, vol. 6, no. 8, pp. 65439–65448, 2018.

[10] W. E. I. Xiong and L. I. Xiong, "Smart Contract Based Data Trading Mode Using Blockchain and Machine Learning," *IEEE Access*, vol. PP, p. 1, 2019.

[11] S. Wang, X. U. Wang, and Y. Zhang, "A Secure Cloud Storage Framework with Access Control based on Blockchain," *IEEE Access*, vol. PP, p. 1, 2019.

[12] C. Xu, Y. Fang, and Y. Ma, "ScienceDirect Integrated Integrated Application Application of of Blockchain Blockchain in in the the Electric Electric Information Information Management System Management System," *Procedia Comput. Sci.*, vol. 162, no. Itqm 2019, pp. 88–93, 2020.

[13] M. Debe, K. Salah, M. Habib, and U. R. Rehman, "Monetization of Services Provided by Public Fog Nodes Using Blockchain and Smart Contracts," vol. 8, 2020.

[14] S. Pal, T. Rabehaja, M. Hitchens, V. Varadharajan, and A. Hill, "On the Design of a Flexible Delegation Model for the Internet of Things Using Blockchain," *IEEE Trans. Ind. Informatics*, vol. 16, no. 5, pp. 3521–3530, 2020.

[15] G. Gürsoy, C. M. Brannon, and M. Gerstein, "Using Ethereum blockchain to store and query pharmacogenomics data via smart contracts," *BMC Med. Genomics*, vol. 13, no. 1, pp. 1–11, 2020.

[16] M. Debe, K. Salah, M. Habib, and U. R. Rehman, "IoT Public Fog Nodes Reputation System : A Decentralized Solution Using Ethereum Blockchain," *IEEE Access*, vol. 7, pp. 178082–178093, 2019.

[17] S. Wang, Y. Zhang, and Y. Zhang, "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems," *IEEE Access*, vol. 6, pp. 38437–38450, 2018.

[18] J. S. Park, T. Y. Youn, H. Bin Kim, K. H. Rhee, and S. U. Shin, "Smart contract-based review system for an IoT data marketplace," *Sensors (Switzerland)*, vol. 18, no. 10, pp. 1–16, 2018.

[19] O. López-pintado, L. García-bañuelos, I. Weber, and A. Ponomarev, "Caterpillar : A business process execution engine on the Ethereum blockchain," no. April, pp. 1162–1193, 2019.

[20] D. C. Nguyen, P. N. Pathirana, and S. Member, "Blockchain for Secure EHRs Sharing of Mobile Cloud Based E-Health Systems," *IEEE Access*, vol. 7, pp. 66792–66806, 2019.

[21] M. Debe, K. Salah, M. Habib, and U. R. Rehman, "Blockchain-Based Decentralized Reverse Bidding in Fog Computing," vol. 4, 2020.

[22] A. Rosic, "¿Qué son los contratos inteligentes? [Guía definitiva para principiantes sobre contratos inteligentes]," 2017. [Online]. Available: https://blockgeeks.com/guides/smart-contracts/. [Accessed: 23-Feb-2020].

Anexo 4: Certificado de intervención.

A continuación, se adjunta el certificado de la intervención en las conferencias CITIS de la Universidad Politécnica Salesiana para la publicación del artículo de la revisión sistemática.

UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR

CITIS

Universidad Politécnica Salesiana
grants the present Certificate to:

Jhonny Carrión Ramírez

For having presented the full paper:

"Smart contracts for user registration on ethereum technology: Systematic literature review"

by the authors: Ramiro Cristian Narváez, Jhonny Carrión Ramírez, Leonardo Benavides Maldonado and María del Cisne Ruilova, at the:

VII International Conference on Science, Technology and Innovation for Society **CITIS 2021**

Awarded in Guayaquil, Ecuador, May 26-28, 2021.

PhD. Juan Cárdenas Tapia. sdb.
Rector

PhD. Juan Pablo Salgado
Vice-rector of Research

Dra. Ana Reino Molina
General Secretary

Registro interno
Certificado No.: 32575

Anexo 5: Postulación a intervención BCCA2021

A continuación, se adjunta el correo de envió del artículo a la conferencia sobre Computación y Aplicaciones de Blockchain (BCCA2021).

BCCA2021 submission 21

jue., 1 jul. 2021 01:30

Dear authors,

We received your submission to BCCA2021 (The Third International Conference on Blockchain Computing and Applications):

Authors : Cristian Ramiro Narváez Guillén, Jhonny Fernando Carrión Ramirez, Pablo Fernando Ordoñez Ordoñez and Ruperto Alexander López Lapo
Title : Creation and deployment of a Smart Contract on the EVM Blockchain network for a user registration application.
Number : 21

Anexo 6: Articulo de la conferencia BCCA2021

A continuación, se presenta el artículo para la conferencia sobre Computación y Aplicaciones de Blockchain (BCCA2021).

Creation and deployment of a Smart Contract on the EVM Blockchain network for a user registration application.

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

Jhonny Fernando Carrión Ramírez
Facultad de Energía, CIS
Universidad Nacional de Loja
Loja, Ecuador
jfcarrionr@unl.edu.ec
0000-0003-2087-4517

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

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

***Abstract*—The integration of technologies such as blockchain and smart contracts have a great capacity to be integrated into systems such as the registration of users of a Smart Lab, in addition to the advantages offered as decentralization makes it one of the most appropriate to provide data security.**

In the present work, it is addressed the integration of Ethereum smart contracts with a system of control and access of students to a Smart Lab, changing the technology currently used which was a simple database and a centralized system, to a more optimal system using blockchain mining and decentralized, using smart contracts as an intermediary between the system that interacts with the user and the security of keeping their data.

***Index Terms*—Blockchain, Smart Contract, Smart Lab, decentralization, centralization, Remix IDE, Ethereum, Ropsten**

I. Introduction

The introduction of new technologies to improve the functioning of Smart Labs[1] in National University of Loja (UNL) is essential since they could be more efficient and secures. Blockchain technology can offer these features using Smart Contracts inside a system to help obtain a decentralised system and much more secure.

The smart contract are of the contractual type, once deployed, is immutable, providing individual security to the user and its information. The present work demonstrates that ones can be integrated and show better functionality for the entire system. The use of tools that facilitate the coding, compilation, deployment and interaction of an one with a system is of great importance and even more important to know about them to facilitate integrating these technologies to programmers or system owners who wish to update or improve them.

[1]Local research at UNL

In consequence, the tools that allow us to reinforce this type of smart contract are MetaMask [2] as a wallet for interactive with the blockchain platform. While Web3.js [3] library helps to communicate with the smart contract through an API Json, this one is provided by an Ethereum node.

The present work is break down in diverse sections that is explained to below: Related Works is mentioned the ones that are served to background to carry out this work. Materials and Methods, it is described the methods that are used to achieve the implementation of smart contracts, as well as, the tools necessary to create and deploy the ones.

Finally, Conclusions section where is determined if it is successful or not the implementation of the smart contract. Future Works section is describe the element that could improve in the system implemented as well as adding new features.

II. Related Works

There exist diverse types of system proposed to the implementation of blockchain technology using smart contract as it is details to below:

A. Application to registry the academic degrees

In [1] is mentioned about the combination of the Smart Contract with Blockchain technology, where it offers an application with the idea to assign academic degrees to the students. Moreover, it could be considered an alternative and leading innovator application, opening a wide range of possibilities in contrast to those based on monolithic stereotypes. In the first instance, this type of technology generates distrust since it is

[2]https://metamask.io/
[3]https://web3js.readthedocs.io/en/v1.3.4/

a new technology that offers enormous potential, generating a great impact in the superior education of Ecuador, thanks to the trustworthy creation, decentralised and available on the moment that it is accomplished the restrictions.

B. Application to registry enrollments of the UCE Students

According to [2], the project is created through using Smart Contract, offering the students the integrity of the data with the goal of implement the decentralised information based on Ethereum Blockchain. With the implementation of Blockchain could notice a system with a better result in terms of efficiency, low costs and democracy promotion in the creation of Smart Contract. It is noteworthy to mention: the gas in the Smart Contract transactions; due to there is a relation among significant use of gas, the transaction is processed faster.

C. Auto-Runnable Electronic Contract and payment using Blockchain technology

As mentioned in [3], it is shown to the deployment of Smart Contract to carry out electronic payments, looking for the way of once making the payment it complies the other part of a transaction without no human intervention, i.e., auto-runnable. It is determined in this work that the emergence of Smart Contract Auto-Runnable is a necessity to massive operations.

D. Analysis of the use of blockchain technology for information management in residential alarm systems.

Carrion [4], analyses the blockchain technology, which has been used in many implementations as an emerging and innovative technology, this used case for a residential alarm system. The Ethereum platform provides a way to offer the service without an intermediary. The management and acquisition of information have evolved; relying on decentralised systems and smart contracts have improved the inefficiencies of the current centralised services. Comparing these and the decentralised one through Ethereum is much improved, offers better features, and the security is more efficient than those possessed by a centralised system. In Ecuador, blockchain technology caused the elimination of the need to rely on a central server, offering 24-hour service availability in case of an emergency making residential alarm systems in the decentralised model an efficient and secure tool.

III. Materials and Methods

Qualitative and bibliographic [5] research are used respectively. It is established a structure for the implementation of smart contracts in student registration, as well as it is collected studies related to the development of smart contracts on the Ethereum platform with blockchain technology. This allowed to perform the registration of students in a contract deployed in the ropsten test network and the simulation of the real environment with the aforementioned architecture.

TABLE I
Tools used to the implementation of the solution

Tools	Description
Visual Studio Code	Visual Studio is a light open-source editor with powerful features that is execute in the desktop and it is available to Windows, macOS and Linux
Remix IDE	It is a powerful open-source tool that helps to write and deploy Smart Contract using Solidity from the browser [7]–[19]
MetaMask	It serves to create wallet for cryptocurrency. Nevertheless, it helps to control every interaction of the user with the DApp, and it makes the necessary operations to carry out them [20]
Ropsten Testnet	Also it is known as Ropsten Ethereum or Ethereum Testnet. It is a test net to execute the same Ethereum Protocol and it is used for testing the same purposes before the implementation on the main net (Mainnet) [21]–[23].

A. Experimentation

1) Analysis: On based to the analysis presented in [6], it is used the follow tools to deploy the solution on Smart Contract as shown in Table I.

The web application to be created must comply the follow requirements among them: Student Enrollment, Academic Booking, Reports, Assistant Control, Automatic Control of Gates.

2) Design and Implementation: In order to design the application, it uses the *C4Model* that allows to have a complete overview of the system to create as shown in Fig. 1.

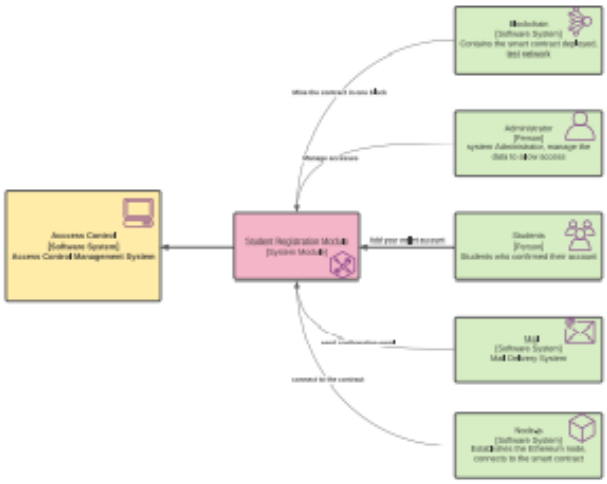


Fig. 1. System Overview of the architecture

Once it is defined the architecture, it is created the internal modules such as: Creation of Students (see Fig. 5) and Validation Students (see Fig. 6).

It is carry out all the necessary steps to deploy the smart contract on the blockchain in this case of the Ropsten testnet, in addition to monitoring all the transactions that took place within it. Some libraries are used to proceed to instantiate the contract and interact with it through the use of calls to

the previously created functions to add students and obtain their information. Moreover the node is established to run the blockchain and in this monitor the generations of transactions that are adhering. The control is done through the use of the Etherscan platform of the Ropsten testnet, which has an interface with information on the transactions in each block that is mined with its respective hash.

3) Testing: The unit tests for the smart contract functions and integrity tests are performed using Remix ID

IV. Results

1) Smart Contract Deployment: Remix, a tool for development and deployment of the smart contract, was used to connect to Metamask's personal account, which allowed us to deploy the smart contract on the Ropsten test network, taking into account the parameters shown in the Table II.

TABLE II
TOOLS USED TO THE IMPLEMENTATION OF THE SOLUTION

Module	Parameters
Compiler	Pragma solidity 0.7.1
Languaje	Solidity
Enviroment	Injected Web3
Account	Solidity provided account 0xb3B8.....037
Gas Limit	Default 30000 and using 0 wei

Injected Web3 environment connects to the Metamask account which displays the different possibilities of test networks, in our case is used Ropsten. Metamask indicate the account number, balance and transaction values that is used to proceed with the deployment of the smart contract.

Once the deployment of the smart contract for user registration is done, the Remix console indicates the output of information as shown in Fig. 2, where the compilation of the smart contract creation process in the Ropsten network is checked and results in the creation of the smart contract.

Fig. 2. Remix IDE smart contract deployment control

It is validated the success transaction in the Ropsten Net as shown in Fig. 3, i.e., this process has been validated in the web https://ropsten.etherscan.io.

2) Web application: The student registration screen is shown in Fig. 4 and the process to registry students is shown in Fig. 5.

The student confirm the registry process to go on with the generation of the transaction as shown in Fig. 6.

Fig. 3. View of the transaction after deployment of Smart Contract on Ropsten Net

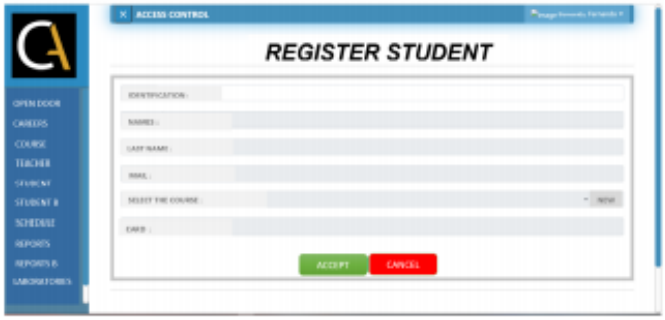


Fig. 4. UI Web App to register a new student

3) Testing: When the transactions are successful, the information can be accessed from the system to visualise the data stored in the blockchain, for this the function *get()* of the Smart Contract be accessed, this obtain the information of a registered student through wallet account using the email which is used in the registry phase, as can be seen in 7, the console output of the transaction for the *function get()* can be visualised in Fig.8.

V. Conclusions and Future Works

The results were satisfactory in the Smart Lab UNL, through the web application with the implementation of a decentralized, secure, accessible, and traceable registry in the user records analyzed in the Ropsten test network.

The process to deploy the smart contract in this project was done through Remix IDE, this contract kept the student data and remained mined in a block of the Ropsten test network, also with the use of MetaMask the transactions sent to each account were signed allowing to optimize the student registration process allowing a more personal interaction.

The functionality in the DApp through the deployment of the smart contract and its interactions provides efficiency and optimizes the student registration process meeting the expectations expected in the Smart Lab UNL as well as the ease of access.

The Ropsten test network used in the experimentation responds appropriately by returning transaction information for each registered student.

Migrate the web system for managing access to the macro-laboratories from php to node.js, which would improve re-

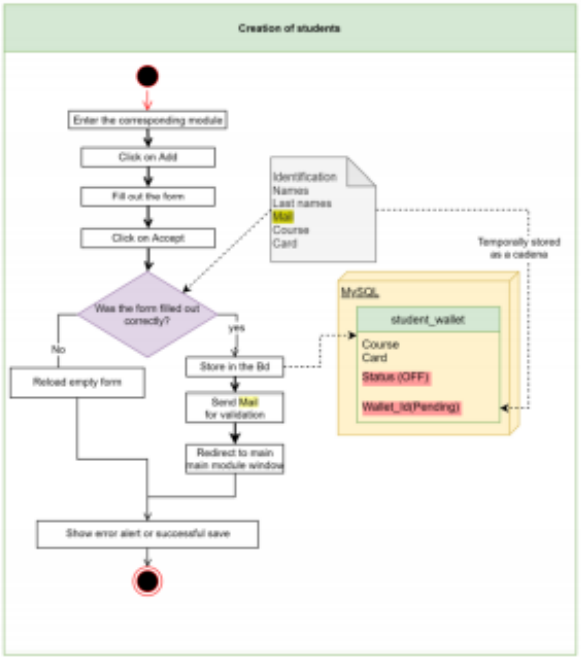


Fig. 5. Workflow to add a new students to the smart contract

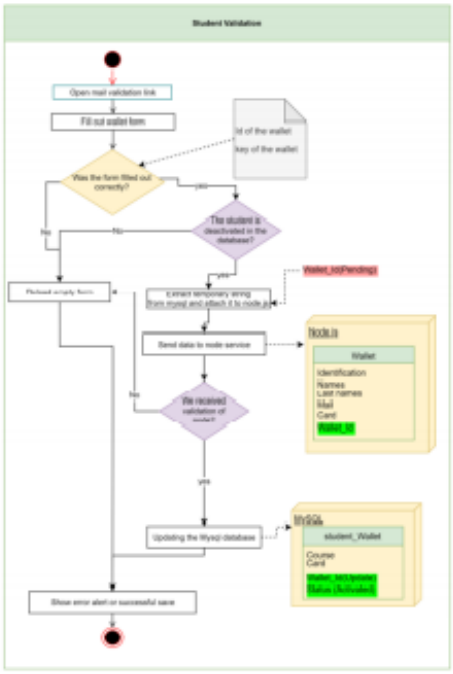


Fig. 6. Workflow to confirm that new students are added to the smart contract via MetaMask

sponse times in the management and control of users and their iteration with the ehtereum network.

It can be adopted in the current web system of access control the application of the Blockchain with its characteristics being one of this the smart contracts to improve productivity and optimize all processes, highlighting the use of the Ethereum

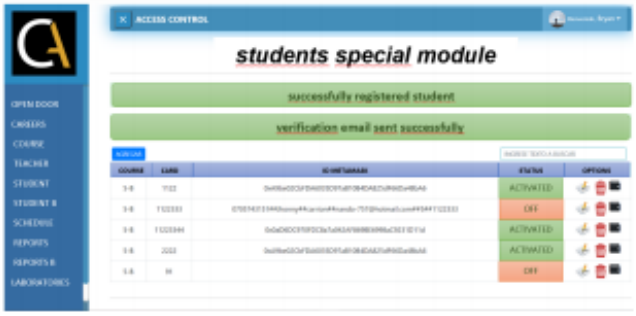


Fig. 7. Information of a registered student

```
Transaction:  Result {
  '0': '705743151',
  '1': 'JHONNY',
  '2': 'CARRION',
  '3': '5',
  '4': '1122333'
}
```

Fig. 8. Console in prompt when the transaction is successful using get() function

platform, to allow generating faster processes and with greater security providing a comparison with the results obtained in this investigative work and improve the implementation of new technologies in the country.

In the future, new smart contracts can be developed in solidity, to optimize the iteration processes between the use of Smart Lab UNL and the knowledge acquired by the students, which will allow comparing the different learning results and decentralised control (knowledge - use in Smart Lab UNL), allowing traceability in the teaching-learning processes.

REFERENCES

[1] L. E. Rosero Correa, "Propuesta de una aplicación basada en la tecnología blockchain para el registro de títulos académicos," B.S. thesis, Quito: UCE, 2019.

[2] M. Á. García Merizalde, "Modelo de solución mediante el uso de smart contracts para el registro de matrículas de estudiantes en la uce," B.S. thesis, Quito: UCE, 2019.

[3] M. E. Sáenz, "Contratos electrónicos autoejecutables (smart contract) y pagos con tecnología blockchain," *Revista de estudios europeos*, no. 70, pp. 69–97, 2017.

[4] A. K. Carrión Basantes, "Análisis de la utilización de la tecnología blockchain para la gestión de la información en sistemas de alarmas residenciales," B.S. thesis, Quito, 2018., 2018.

[5] C. M. Castaño Garrido and M. R. Quecedo Lecanda, "Introducción a la metodología de investigación cualitativa," 2002.

[6] L. B. M. Ramiro-Cristian Narvaez, Jhonny Carrión and M. del Cisne Ruilova, "Smart Contracts for user registration on Ethereum technology: Systematic Literature Review - In Press." Singapore: Springer, 2022. [Online]. Available: https://www.springer.com/gp/book/9789811641251aboutAuthors

[7] D. Čeke and S. Kunosić, "Smart contracts as a diploma anti-forgery system in higher education-a pilot project," in *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*. IEEE, pp. 1662–1667.

[8] K. Hasan, Haya R, "Combating deepfake videos using blockchain and smart contracts," *Ieee Access*, vol. 7, pp. 41 596–41 606, 2019.

[9] K. Hasan, "Blockchain-based proof of delivery of physical assets with single and multiple transporters," *IEEE Access*, vol. 6, pp. 46 781–46 793, 2018.

[10] K. Hasan, Haya, "Proof of delivery of digital assets using blockchain and smart contracts," *IEEE Access*, vol. 6, pp. 65 439–65 448, 2018.
[11] W. Xiong and L. Xiong, "Smart contract based data trading mode using blockchain and machine learning," *IEEE Access*, vol. 7, pp. 102 331–102 344, 2019.
[12] S. Wang, X. Wang, and Y. Zhang, "A secure cloud storage framework with access control based on blockchain," *IEEE Access*, vol. 7, pp. 112 713–112 725, 2019.
[13] M. Debe, K. Salah, M. H. U. Rehman, and D. Svetinovic, "Monetization of services provided by public fog nodes using blockchain and smart contracts," *IEEE Access*, vol. 8, pp. 20 118–20 128, 2020.
[14] I.-S. Park, Y.-D. Lee, and J. Jeong, "Improved identity management protocol for secure mobile cloud computing," in *2013 46th Hawaii International Conference on System Sciences*. IEEE, 2013, pp. 4958–4965.
[15] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1594–1605, 2018.
[16] M. Debe, K. Salah, M. H. U. Rehman, and D. Svetinovic, "Iot public fog nodes reputation system: A decentralized solution using ethereum blockchain," *IEEE Access*, vol. 7, pp. 178 082–178 093, 2019.
[17] J.-S. Park, T.-Y. Youn, H.-B. Kim, K.-H. Rhee, and S.-U. Shin, "Smart contract-based review system for an iot data marketplace," *Sensors*, vol. 18, no. 10, p. 3577, 2018.
[18] M. Debe, K. Salah, M. H. U. Rehman, and D. Svetinovic, "Blockchain-based decentralized reverse bidding in fog computing," *IEEE Access*, vol. 8, pp. 81 686–81 697, 2020.
[19] O. López-Pintado, L. García-Bañuelos, M. Dumas, I. Weber, and A. Ponomarev, "Caterpillar: A business process execution engine on the ethereum blockchain," *Software: Practice and Experience*, vol. 49, no. 7, pp. 1162–1193, 2019.
[20] J. P. Claros Romero *et al.*, "Aplicación de blockchain para el uso de transportes," 2021.
[21] M. Neto, "Get ropsten ethereum—the easy way," 2018.
[22] G. Gürsoy, C. M. Brannon, and M. Gerstein, "Using ethereum blockchain to store and query pharmacogenomics data via smart contracts," *BMC Medical Genomics*, vol. 13, pp. 1–11, 2020.
[23] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for secure ehrs sharing of mobile cloud based e-health systems," *IEEE access*, vol. 7, pp. 66 792–66 806, 2019.

Anexo 7: Código del programa

En el cd 2 se presenta todos los archivos del sistema.