Desarrollo de un Smart Contract para registro de actas de calificaciones com tecnología Blockchain para la facultad de ciencias y tecnologías UNCA

Diego Duré
Universidad nacional del Caaguazú
Coronel Oviedo, Paraguay
didure@fctunca.edu.py

Fredy Ramírez
Univerdidad nacional del Caaguazú
Coronel Oviedo, Paraguay
framirez@fctunca.edu.py

Hector Estigarribia
Universidad nacional del Caaguazú
Coronel Oviedo, Paraguay
hestigarribia64@fctunca.edu.py

***Abstract*—In the present work presented, is proposed the development and deployment of a Smart Contract to record report cards with the programming language Solidity that Ethereum has within its Blockchain as an alternative to currently used centralized systems. For this purpose a compilation was made on the protocols that support Blockchain technology to better understand how it works, highlighting your computer security and data privacy and data privacy in front of current centralized systems, in addition to studying the syntax of the Solidity programming language and its development environment. In addition, an evaluation of the cost of deployment and use of the Smart Contract resulting in a relatively low cost based on research given in a test environment. However, regarding the graphical interface, for those who are unaware of the particular operation of the development and deployment environment, it could be unfriendly.**

***Keywords*—ethereum, blockchain, smart contract, development, computer security.**

***Resumo ou Resumen*— En el trabajo presentado, se planteó el desarrollo y despliegue de un Smart Contract para registrar actas de calificaciones con el lenguaje de programación Solidity que tiene Ethereum dentro de su Blockchain como alternativa a los sistemas centralizados actualmente utilizados. Para dicho fin se realizó una recopilación sobre los protocolos que sostienen la tecnología Blockchain para entender mejor su funcionamiento, destacando su seguridad computacional y privacidad de datos en frente de los sistemas centralizados actuales, además del estudio de la sintaxis del lenguaje de programación Solidity y su entorno de desarrollo. Además, se procedió a hacer una evaluación del coste de despliegue y utilización del Smart Contract dando como resultado según la investigación dada en un ambiente de prueba un coste relativamente bajo. Sin embargo, en cuanto a la interfaz gráfica para aquellos que desconocen el particular funcionamiento del entorno de desarrollo y despliegue les podría resultar poco amigable.**

***Palavras-chave*—ethereum, blockchain, smart contract, desarrollo,seguridad computacional**

I. Introducción

A. Planteamiento del problema

El dato más valioso tanto para la institución como para los estudiantes son las calificaciones, de ahí que existe la imperiosa necesidad de proteger los mismos. En este trabajo se plantea poder hacerlo contando con las características de inmutabilidad y seguridad computacional que ofrece la Blockchain para registrar las actas de calificaciones con la fiabilidad de que dichos documentos no podrán ser borrados o alterados, esta consiste en una estructura de datos en que la información está organizada en bloques permitiendo almacenar grandes cantidades de datos ordenados em el tiempo, de forma descentralizada e inmutable. Al ser inmutable ofrece una alta veracidad a los registros de las actas de calificaciones que podrá registrar la Facultad de Ciencias y Tecnologías, que hasta ahora no cuenta con ningún sistema o software que ofrezca este alto grado de veracidad, transparencia y respaldo de información.

II. Delimitación y alcance

La idea principal de este proyecto es el desarrollo y despliegue de un Smart Contract en un ambiente de prueba para registrar actas de calificaciones con el lenguaje de programación Solidity que tiene Ethereum dentro de su Blockchain y analizar el coste de su despliegue y uso. Este Smart Contract podrá registrar permanentemente y sin posibilidad de modificación las actas de calificaciones, puesto que estará incrustada en la Blockchain de Ethereum, un sistema descentralizado e inmutable, con alta seguridad computacional y de costo relativamente bajo. Este Smart Contract registrará de las actas de calificaciones su número, código, la carrera, que en este caso, la muestra a ser utilizada será la de ingeniería en informática del quinto año, segundo semestre de la facultad de ciencias y tecnologías UNCA del año 2019, la materia, que en este caso serán, todas las materias de dicha carrera del segundo semestre, el curso, que em este caso, será la del quinto curso, el periodo, que son: la primera, la segunda, y la tercera oportunidad de cada materia, el número de documento (ci), junto con la calificación del alumno.

III. PREGUNTA DE INVESTIGACIÓN

¿Cual es la factibilidad de Desarrollar un Smart Contract com tecnología Blockchain para gestionar actas de calificaciones en la FcyT?

IV. objetivos

B. Objetivo general

Desarrollar un Smart Contract para registrar actas de calificaciones con tecnología Blockchain para la facultad de Ciencias y Tecnologías Unca.

C. *Objetivo específico*

Examinar los protocolos y funcionamiento de la Blockchain de Ethereum.Demostrar la utilidad del Smart Contract desplegado en un ambiente de prueba con los registros de las actas de calificaciones.Evaluar la factibilidad del despliegue y uso del Smart Contract.

V. JUSTIFICACIÓN

Con esta investigación estaremos analizando los protocolos de la Blockchain de Ethereum que a su vez nos facilitará desarrollar un Smart Contract basada en la necesidad puntual para mejorar la seguridad del registro de actas de calificaciones con la tecnología blockchain, puesto que es un sistema descentralizado e inmutable nos ayudará a mantener la información concisa y transparente.

Actualmente la facultad de Ciencias y Tecnologías registra las actas de calificaciones de forma física(hojas de documentos) y de forma digital, cuenta con un sistema académico hecho a medida con el lenguaje de programación foxpro. Proyecto de Microsoft al que brindó soporte hasta el 2015 [18], pasado dicho año ya no recibió actualizaciones o parches de seguridad por actuales o nuevos errores, siendo esto una crítica vulnerabilidad de seguridad informática en cualquier organización o empresa que cuente con un software desarrollado en dicho lenguaje. Como medida de seguridad y contingencia de datos se realizan procedimientos de backup de forma física y de forma virtual en un servidor. El Smart Contract a ser desarrollado funcionará como una medida de seguridad adicional a las ya mencionadas, con la diferencia de que la información estará distribuida en una red descentralizada, con un histórico irrefutable de información.

VI. MARCO TEÓRICO CONCEPTUAL, OPERACIONAL Y LEGAL

D. *Antecedentes*

La primera aparición de los Smart Contract de forma pública fue con Nick Szabo, jurista y criptógrafo en 1995, en lo que menciona lo siguiente:

La idea básica detrás de los contratos inteligentes es que muchos tipos de cláusulas contractuales (como garantías, fianzas, delimitación de derechos de propiedad, etc.) pueden integrarse en el hardware y el software con el que tratamos, de tal manera que se infrinja el contrato costoso (si se desea, a veces de manera prohibitiva) para el infractor. El éxito del derecho consuetudinario de los contratos, combinado con el alto costo de reemplazarlo, hace que valga la pena preservar y hacer uso de estos principios

cuando sea apropiado. Sin embargo, la revolución digital está cambiando radicalmente los tipos de relaciones que podemos tener En aquel entonces (1996) Nick Szabo básicamente creó la teoría sobre el funcionamiento general de los Smart Contract, lamentablemente era prácticamente imposible concretar la idea con la tecnología existente, mencionando también el costo elevado que implicaría ponerlos en práctica.Las computadoras hacen posible la ejecución de algoritmoshasta ahora prohibitivamente costosos, y conectan en red latransmisión rápida de mensajes más grandes y sofisticados [1].

Para concretar la idea de los Smart Contract se necesitaba de protocolos que en aquel entonces recién estaban siendo descubiertos, como las transacciones programables y un sistema financiero que registre todas las actividades que se realicen sin dar lugar a los fraudes.

Además, los informáticos y los criptógrafos han descubierto recientemente muchos algoritmos nuevos y bastante interesantes.

La combinación de estos mensajes y algoritmos hace posible una amplia variedad de nuevos protocolos [1].

Lo que para ese entonces era prácticamente imposible, años más tarde (2009) con el nacimiento de la Blockchain de Bitcoin se haría posible. Con la cadena de bloques se puede verificar el registro financiero de las actividades, evitando fraudes y dando lugar a las transacciones programables.

Luego se concretó la idea de los Smart Contract con el nacimiento del proyecto Ethereum, oficialmente en 2016. Basándose en la Blockchain de Bitcoin con algunas modificaciones, dando lugar en el bloque a los Smart Contract y abriendo infinidades de posibilidades para las aplicaciones descentralizadas y la realización de acuerdos entre dos o más partes sin necesidad de un intermediario. Una vez indagado en la Biblioteca así como en el repositorio digital de la Facultad de Ingeniería en Informática, se puede afirmar que no se ha encontrado un Proyecto de Trabajo de Graduación enfocado en la necesidad puntual para registrar actas de calificaciones con tecnología Blockchain.

Existen proyectos que hacen uso de los Smart Contract enotras instituciones para otros casos como los siguientes:

Explorando la Blockchain de Ethereum y el desarrollo de smart contracts, realizado por Víctor Miranda Palacios en el año 2018 por el título de Grado en Ingeniería de Sistemas de Telecomunicaciones, Grado en Ingeniería Telemática, de la universidad Politécnica de Catalunya. Se basa en la creación de un Smart Contract para historiales médicos

Los contratos inteligentes en España, la disciplina de los Smart Contract, publicado por la revista de derecho civil en el 2018 y escrito por Antonio Legerén-Molina Profesor Contratado-Doctor de Derecho civil Universidade da Coruña.

Desarrollo de un prototipo basado en Blockchain aplicado a la plataforma IOT sobre un sistema embebido, realizado por Esteban Adrián Restrepo e Daniel Arturo Olaya U en el año 2018 por el título de grado en Ingeniería Electrónica.

VII. BASES TEÓRICAS

E. *La tecnología Blockchain*

La versión puramente de igual a igual de efectivo electrónico

permitiría en línea pagos que se enviarán directamente de una parte a otra sin pasar por una institución financiera (Satoshi Nakamoto, 2009).

La tecnología Blockchain representa la base para una nueva era de confianza digital. Blockchain permite eliminar al "hombre del medio" (The Man of the Middle por sus siglas en inglés). (David Esteban Plaza Ramírez, 2019).

La blockchain es, en esencia, una base de datos distribuida, que almacena todas las transacciones que se han realizado desde el inicio del sistema (2015). Este estaría compuesto por nodos cada uno independiente del otro siguiendo así el esquema peer to peer (P2P) gestionando un registro único donde se registran todas las operaciones realizadas.

Como su nombre indica es una cadena de bloques. Cada bloque incluye una serie de transacciones y la referencia al anterior bloque de la cadena. Y una transacción no se considera válida a menos que forme parte de la cadena de bloques. De esta manera la blockchain contiene todas las transacciones consideradas válidas.Para entender mejor su operación es necesario conocer algunas definiciones:

F. Hoja única contable abierta y distribuida (Open Ledger)

La tecnología Blockchain está constituida por una base de datos distribuida, que consta de un registro único compartido por todos los nodos que contiene la información de los dueños de los activos y el histórico del intercambio de la propiedad de los mismos (transacciones). Esto quiere decir que se puede establecer la cadena de propiedad del activo desde su origen o emisión. En resumen, la hoja única distribuida guarda el estado actual de la Blockchain y a partir de esta se puede establecer "quién es dueño de que y quien se lo transfirió".

G. Nodos

Cada participante de la red Blockchain constituye un nodo. Cada vez que se generan transacciones estas se transmiten a los nodos, un bloque se confirma por prueba de consenso y se añade a la cadena de bloques una vez que el 51% de los nodos aprueben la información como correcta, esta operación actualiza la hoja contable que cada participante(nodo) almacena, esta es una de las razones por la que corromper la seguridad de la Blockchain es casi imposible, puesto que se tendría que obtener el control de más de la mitad de los nodos de la red. Se requiere un enorme esfuerzo computacional para obtener el control por lo que pierde el sentido realizar un ataque.

H. Mecanismo de consenso

Prueba de trabajo(Proof-of-Work): La propiedad de los activos y la transferencia de los mismos son validadas por los participantes de la red(mineros) donde el que tiene mayor poder computacional tiene más posibilidades de descifrar el hash y validar el bloque para ser recompensado por ello, el algoritmo usado por Ethereum es Ethash hace uso de una propiedad llamada en inglés "memory hardness" que provoca que el rendimiento del procesador dependa de la rapidez con la que puede mover datos en la memoria en lugar de por la rapidez con la que realiza operaciones de cálculo [5].Bitcoin implementa un algoritmo de tipo hashcash que consiste en encontrar un hash SHA256 de una determinada dificultad. Como cabía esperar, la fuerte competencia ha impulsado el desarrollo de procesadores específicos de tipo ASIC (acrónimo de Circuito Integrado para Aplicaciones

Específicas o Application-Specific Integrated Circuit en inglés), que sirven para optimizar esta función y realizar cálculos más rápidamente. En la actualidad uno sólo de estos procesadores es capaz de obtener más de mil millones de hashes por segundo[5]. La mayoría de nodos debe de estar de acuerdo sobre el estado resultante y el orden de cada transacción, esto se logra verificando la hoja contable distribuida i) si el activo existe (origen y emisión) ii) si este pertenece a quien transfiere (cadena de propiedad) y iii) si previamente no ha sido transferido a otro participante (evitar el fraude por doble desembolso). El mecanismo de consenso también se encarga de verificar que los nodos sean honestos, es decir, sean reales y no varios participantes falsos dominados por un atacante para ganar más participación en la decisión. Una vez verificada, una transacción no podrá ser eliminada o modificada. Ethereum utilizó este mecanismo de consenso para luego pasar a otro.

Prueba de participación(Proof-of-Stake): Con las pruebas de participación, se aprueban bloques por los mineros que mas tokens tengan o por la antigüedad de los mismos, esta es una ventaja, puesto que el que quiera aprobar un bloque debe de mostrar cierto interés en la plataforma, conseguir muchos tokens para así poder participar en la red como minero y no formar parte de la red para simplemente obtener una compensación por su trabajo sin ningún otro interés en la red de Ethereum, otra ventaja destacable es el bajo poder computacional necesario y consumo de energía, puesto que se elimina el mecanismo de resolución de problemas.

I. Bloque

Para brindar mayor seguridad las transacciones se almacenan en bloques y este se encadena a bloques anteriores. Cada bloque contiene una estampilla de tiempo que indica el momento en el cual se hicieron las operaciones contenidas en él y un número cifrado (hash) que se crea a partir de la combinación de las transacciones contenidas y la referencia al bloque anterior. Las transacciones quedan en firme cuando su bloque contenedor se enlaza a la cadena de bloques. Cualquier modificación en las transacciones modifica el hash del bloque, esto rompe la cadena y las transacciones se anulan.

La cadena de bloques de Ethereum es similar a la de Bitcoin en muchas cosas, aunque también tiene algunas diferencias. La diferencia principal entre Ethereum y Bitcoin respecto a la arquitectura de la cadena de bloques es que, a diferencia de Bitcoin, los bloques de Ethereum contienen una copia tanto de la lista de transacciones como del estado más reciente. A parte de eso, otros dos valores, el número de bloques y la dificultad, también están almacenados en el bloque. El algoritmo básico de validación de bloque en Ethereum es el siguiente:

1. Comprueba si el bloque previo referenciado existe y es válido.

2. Comprueba que el sello de tiempo del bloque es mayor que el bloque previo referenciado y menor de 15 minutos en el futuro.

3. Comprueba el número de bloque, la dificultad, la transacción raíz, la raíz tía y el límite de gas (varios conceptos de bajo nivel específicos de Ethereum) son válidos.

4. Comprueba que la prueba de trabajo del bloque es válida.

5. Establece S[0]como el estado al final del bloque previo.

6. Establece TX como la lista de transacciones del bloque, con n transacciones. Para todo i en ..n-1, establece S[i+1] = APPLY(S[i],TX[i]). Si alguna aplicación devuelve un error, o si la cantidad de gas consumido por el bloque hasta este momento excede el GASLIMIT, devuelve un error.

7. Establece S_FINAL como S[n], pero añadiendo al bloque la recompensa pagada por el minero.

8. Comprueba si la raíz del árbol de Merkle del estado S_FINAL es igual a la raíz del estado final proporcionada por la cabecera del bloque. Si esto es así, el bloque es válido, en otro caso, no lo es..

J. *El gas*

Cada transacción que se realiza en Ethereum tiene un determinado costo, este depende de la complejidad de la transacción a ser realizada, cuando hablamos de transacciones engloba tanto transacciones de ether de una cuenta a otra, como la ejecución de un Smart Contract, para esto se utiliza el gas, que es como una especie de combustible, también es una medida de seguridad para evitar la sobrecarga de la red.

K. *Ethereum Virtual Machine*

Es esta la que se encarga de ejecutar el código de los Smart Contract, la cual puede ejecutar código de complejidad algorithmica de manera arbitraria. En términos de ciencias de la computación, Ethereum es “Turing complete” o Touring completo, lo que significa que teóricamente podría resolver cualquier problema computacional razonable [8].

Cuando un contrato es diseñado en Solidity, se convierte en una secuencia de códigos de operaciones, conocidas comoopcodes, para luego pasar a bytecodes y que la EVM pueda entender.

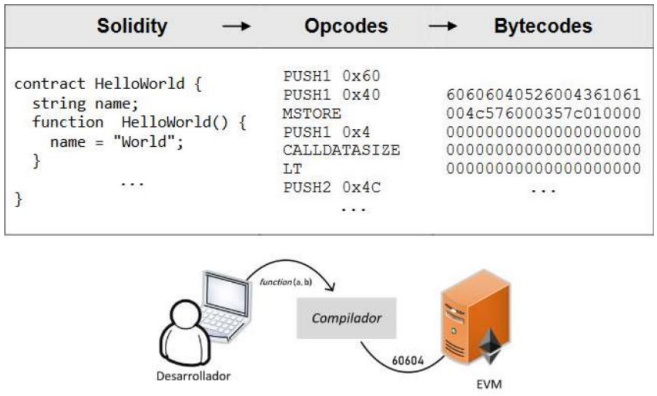

Fig. 1. Compilación del código en la EVM.

L. *Smart Contracts*

Los smart contract son programas informáticos que se asientan en una cadena de bloques, en el cual se programan unas condiciones de actuación a cumplir basados en la simple sentencia de “si, entonces”, y que se ejecutan en el momento que se cumplen esas condiciones, pueden ser aplicados en una inmensa variedad de escenarios, los smart contract tienen ciertos aspectos que lo diferencian de los contratos tradicionales como:

Inmutabilidad: La Blockchain es un libro digital incorruptible de transacciones económicas que puede programarse para registrar no solo transacciones financieras sino prácticamente todo lo que tiene valor (Don and Alex Tapscott, 2016). Una vez minado un bloque nadie podrá modificar la información guardada.

Distribuido: La red está descentralizada, lo que significa que no tiene ninguna autoridad que la gobierne o una sola persona que ejerce control total. Más bien, un grupo de nodos mantiene la red descentralizada.

Deterministas: La Blockchain está completamente organizada, y como no depende de cálculos humanos, las fallas accidentales de este sistema no son una salida habitual. Por lo que el código escrito no debería tener ninguna aleatoriedad.

Verificables: Una vez desplegado el contrato tendrá una dirección única, que puede usarse para que las partes interesadas analicen mejor el código para mayor seguridad.

Estos aspectos mejoran los acuerdos entre dos o más partes en cuanto a que su cumplimiento, no da lugar a la interpretación o manipulación por parte de terceros que generalmente son los intermediarios, consiguiendo la perfección y transparencia de los acuerdos.

Las aplicaciones de los Smart Contracts tienen una gran variedad de posibles escenarios, en los que podemos mencionar los siguientes:

Depósitos de garantía: Por ejemplo, para la compra de una vivienda o cualquier otro activo entre particulares o empresas, el comprador deposita una cierta cantidad a una dirección wallet, y en el momento que se entregue al comprador el título o el bien acordado, se libere el depósito a favor del vendedor.

Apuestas: Con la ayuda de los contratos inteligentes, cualquier persona puede realizar apuestas con alguien, sin necesidad de intermediarios que respalden el pago de tales apuestas, y con total seguridad que después de conocerse el resultado, se liberará la cantidad ganada en la apuesta en su caso.

Votaciones: Con la ayuda de los Smart contract se puede reemplazar la confianza que depositamos en las personas que realizan los conteos de las votaciones, que muchas veces son corruptibles o propensos a fallos, teniendo así la seguridad de que se realizarán votaciones transparentes y con total anonimato.

Logística y Transporte: El ámbito del comercio intervienen múltiples agentes e intermediarios: fabricantes, mayoristas, distribuidores, proveedores, empresas de mensajerías, consumidores, entre otros actores generan una enorme cantidad de documentación mayormente en papel. La aplicación de tecnologías digitales a este sector, reduciría dicho papeleo, agilizaría el transporte de mercancías y reduciría considerablemente los costes [9].

Salud y Sanidad: Algunos de los potenciales usos de la

cadena de bloques en sanidad son:

Confidencialidad para el paciente. Proporcionada por la encriptación y la seguridad de la cadena de bloques.

Trazabilidad y control de los medicamentos. Integrando el blockchain en farmacias, los médicos tendrían un control absoluto del consumo de medicamentos.

Evitar fraudes en las pólizas médicas y controlar malas prácticas en los servicios sanitarios [9].

Certificados Académicos: Con la ventaja que nos da la Blockchain de Ethereum podemos almacenar certificados emitidos de instituciones asegurándonos que estos no podrán ser alterados ni mucho menos perdidos, confirmando la identidad y la autenticidad de títulos, certificados y referencias. Los certificados digitales quedan registrados en la cadena de bloques, creando así un respaldo de seguridad.

VIII. Marco metodológico

M. Tipo de Investigación

La investigación es de tipo propositiva por cuanto se fundamenta en una necesidad o vacío dentro de la institución, una vez que se tome la información descrita, se realizará una propuesta de sistema de evaluación del desempeño para superar la problemática actual y las deficiencias encontradas.

N. Diseño de Investigación

Este trabajo se basó principalmente en el desarrollo de un Smart Contract con el lenguaje de programación Solidity que tiene Ethereum dentro de su Blockchain, para esto examinamos dicha tecnología con el fin de entender mejor su funcionamiento y sus protocolos, así como la sintaxis de dicho lenguaje de programación. Una vez desarrollado el Smart Contract lo desplegamos en un ambiente de prueba para analizar el costo que implica el uso de la misma.

El diseño de investigación según Kerlinger (2002)[21] es el plan y la estructura de un estudio, el enfoque utilizado en este proyecto es el enfoque científico, que según Ramón Ruíz (2007) [22] el método científico se emplea con el fin de incrementar el conocimiento y en consecuencia aumentar nuestro bienestar y nuestro poder.

IX. Codificación y entorno de desarrollo

O. Solidity

Solidity es un lenguaje de alto nivel orientado a contratos. Su sintaxis es similar a la de JavaScript y está enfocado específicamente a la Máquina Virtual de Ethereum (EVM) [15]. Solidity está tipado de manera estática 1 y acepta, entre otras, herencias, librerías y tipos complejos definidos por el usuario. Es utilizada para programar la lógica necesaria, para modelar y controlar los datos de una aplicación. Básicamente un Smart Contract es una colección de funciones y variables, identificados por una dirección única de la Blockchain de Ethereum, la extensión para los ficheros es .sol y todos los contratos empiezan por la siguiente línea de código:

```
6  pragma solidity >=0.4.22 <0.6.0;
```

Fig. 2. Especificación de las versiones del compilador a ser usado.

P. Entorno de desarrollo Remix

El entorno de desarrollo remix es un IDE basado en un navegador web con analisis estático integrado y una máquina virtual de blockchain para pruebas. Podemos elegir el ambiente que por defecto se encuentra en JavaScript VM para pruebas estáticas, con 5 cuentas que poseen 100 ether cada una.

Q. Metamask

Es un plugin o extensión que actualmente puede ser instalado en los navegadores, nos ayuda a desplegar los Smart Contract en un ambiente dinámico, funciona como un puente entre nuestro navegador y la Blockchain sin comprometer nuestra seguridad

X. Compilación y despliegue del smart contract

Como muchos lenguajes de programación Solidity es un lenguaje de programación de alto nivel por lo tanto dicho Smart Contract se convierte en una secuencia de códigos de operaciones, conocidas como opcodes, para luego pasar a bytecodes y que la EVM lo pueda comprender, el Smart Contract genera un archivo de tipo JSON em donde se puede obervar el código opcode y el bytecode

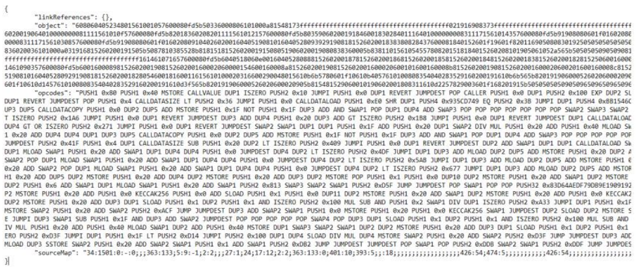

Fig. 3. Parte del codigo opcode y bytecode.

A parte de eso tambien se puede observar toda la información del bloque

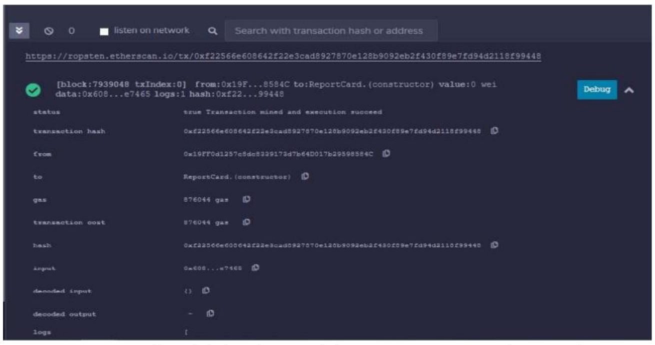

Fig. 4. Detalles del despliegue del Smart Contract en la consola Remix.

Se registraron 18 actas de calificaciones correspondientes a la carrera de ingeniería en Informática, del quinto año décimo semestre, por cuestiones de confidencialidad se utilizaron datos ficticios. Se midió el coste de cada transacción con cierta cantidad de alumnos, asi como el tiempo de cada transacción (registro) de cada acta de calificación

TABLA1

PROMEDIO GENERAL

Acta #	Bloque #	Periodo	Cantidad de Alumnos	Tiempo en segs	Costo en Ether	Costo en dolar	Costo en PYG	
74	7976164	1	50	21	0,0003447940000000000	0,0798715301000000000	543,5926148	Gestión de Proyectos Informáticos
75	7976192	2	25	35	0,0004442750000000000	0,1029163037500000000	700,431588	
76	7976218	3	15	6	0,0003412500000000000	0,0790505625000000000	538,0052431	
77	7982348	1	100	8	0,0011441660000000000	0,2650460539000000000	1803,86024	Diseño de Compiladores
78	7982421	2	50	45	0,0007119090000000000	0,1649137198500000000	1122,375896	
79	7982468	3	25	9	0,0004442750000000000	0,1029163037500000000	700,431588	
80	7982493	1	50	5	0,0007119090000000000	0,1649137198500000000	1122,375896	Auditoría en Informática
81	7982509	2	25	26	0,0004442750000000000	0,1029163037500000000	700,431588	
82	7982534	3	15	18	0,0003412620000000000	0,0790533423000000000	538,024162	
83	7982581	1	100	1	0,0012471800000000000	0,2889092470000000000	1966,269243	Tecnología en Redes y Telecomunic
84	7982629	2	50	1	0,0014240000000000000	0,3298696000000000000	2245,038729	
85	7982657	3	25	3	0,0008890000000000000	0,2059368500000000000	1401,572633	
86	7982791	1	50	33	0,0014290000000000000	0,3310278500000000000	2252,92159	Proyecto Final de Grado
87	7982826	2	25	4	0,0013330000000000000	0,3087894500000000000	2101,570664	
88	7982854	3	15	3	0,0010240000000000000	0,2372096000000000000	1614,409872	
89	7983014	1	100	33	0,0037420000000000000	0,8668343000000000000	5899,532952	Contratos y Licitaciones
90	7987537	2	50	35	0,0177980000000000000	4,1229067000000000000	28059,83097	
91	7987555	3	25	4	0,0111070000000000000	2,5729365500000000000	17510,98677	
PROMEDIOS			44,16666667	16,11111111	0,0024956275000000000	0,5781121103750000000	3934,536791	

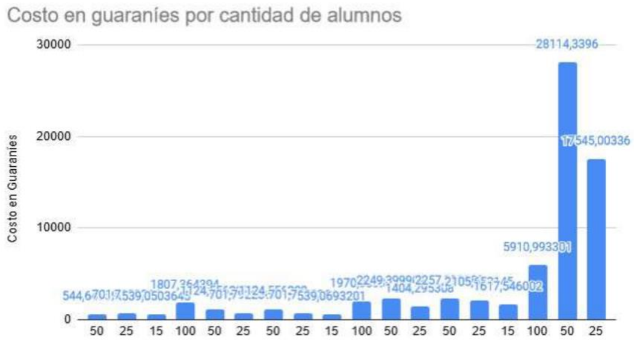

Fig. 5. Gráfico de costo de transacción en guaraníes por cantidad de alumnos.

R. Interpretación de Resultados

Gracias a los costes em guaraníes captados por cantidad de alumnos registrados en cada acta de calificación, podemos observar una alza de coste considerable, em la primera transacción (registro de acta) com 50 alumnos dio un coste de 544,66 gs. Mientras que en la penultima transacción con la misma cantidad de alumnos dio un coste de 28144,3396 gs. Esto a causa de una alza en el precio que Ethereum sufrió, por lo tanto los costes de transacción tambien subieron, el coste promedio de las 18 actas de calificaciones fueron 3934,536791 gs.

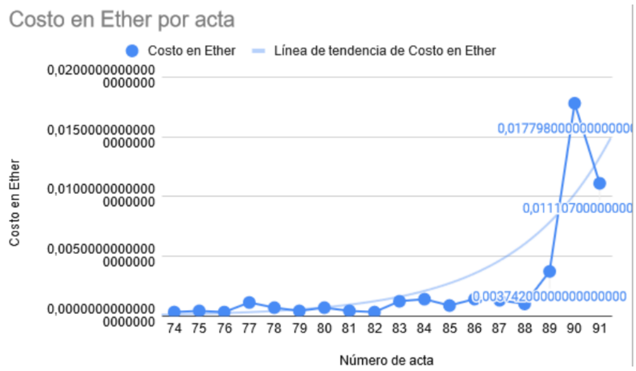

Fig. 6. Gráfico de costo en ether por cada acta y línea de tendencia del costo.

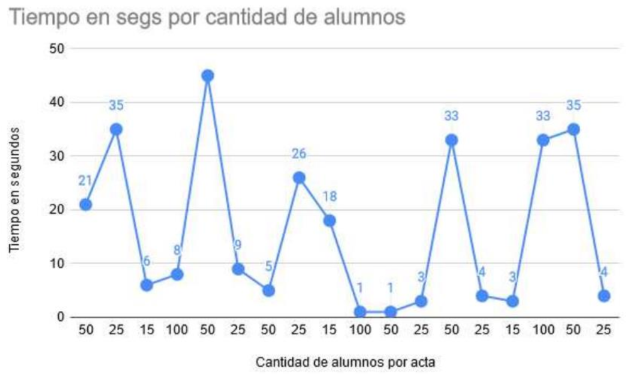

Fig. 7. Gráfico de tiempo de transacción en segundos.

TABLA II

PROYECCIÓN DE COSTE POR AÑO

Proyección del coste por año:

Informática Por semestre (un curso)	0,04384600000000000000	9,5900 $	64819 gs.
Informática Por año (un curso)	0,08742100000000000000	19,1100 $	129993 gs.
Informática Primer al quinto (por año)	0,43747000000000000000	95.55 $	647351 gs.
Todas las carreras Primer al quinto (por año)	1,75032000000000000000	383,18 $	2589404 gs.

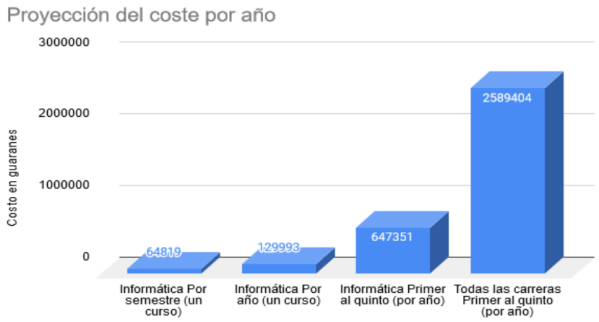

Fig. 8. Gráfico de coste por semestre, por año de un curso, todos los cursos de una carrera y todos los cursos de todas las carreras (Informática, Electricidad, Electrónica y Civil).

XI. CONCLUCIÓN

En el trabajo propuesto se analizó la tecnología Blockchain y sus protocolos, así como la sintaxis de su lenguaje de programación Solidity y su entorno de desarrollo, para luego con base a lo aprendido se pudiera desarrollar un Smart Contract que sirviera como otra alternativa de respaldo a los registros de las actas de calificaciones

Al realizar una prueba piloto se constató que es posible

diseñar de una manera sencilla aplicaciones descentralizadas y aprovechar todas las características que ofrecen, basándose en la privacidad de los datos, información descentralizada y la no existencia de una entidad o empresa que almacene los datos de forma centralizada, o que de alguna manera se pudieran alterar o borrar la información. Sin embargo, la prueba piloto sirvió para dar a entender qu un usuario ajeno a esta tecnología podría verse con dificultades para adaptarse al entorno e interfaz del Smart Contract para interactuar con el mismo, al terminar también podemos apreciar la volatilidad del coste de Ethereum y a su vez los costos de transacción que implica desplegar y calificaciones con la tecnología Blockchain.

AGRADECIMENTOS

Al universo entero por conspirar a mi favor para que este trabajo sea posible.

A mi padre y a mi madre absolutamente por todo lo que me brindaron.

A todos a quienes contribuyeron en la elaboración y conclusión de este trabajo.

REFERÊNCIAS

[1]Nick Szabo, Smart Contracts: Building Blocks for Digital Markets, 1996.

[2]Víctor Miranda Palacios, Explorando la Blockchain de Ethereum y el desarrollo de smart contracts, 2018.

[3] Antonio Legerén-Molina Profesor Contratado-Doctor de Derecho civil, Los contratos inteligentes en España, la disciplina de los Smart Contract, publicado por la revista de derecho civil em el 2018.

[4]Esteban Adrián Restrepo e Daniel Arturo Olaya U, Desarrollo de un prototipo basado en Blockchain aplicado a la plataforma IOT sobre un sistema embebido, 2018.

[5] Alex Preukschat Libro Blockchain, Consenso. [En línea] . Disponible en: https://libroblockchain.com/consenso/ 14 Febrero 2017/18 Enero 2019.

[6] Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 31 enero 2009.

[7] Vitalik Buterin, GENERATION Ethereum White Paper A NEXT GENERATION SMART CONTRACT & DECENTRALIZED APPLICATION PLATFORM, 2013.

[8] Ethereum Homestead: Que es Ethereum? - Documentación. [En línea]. Disponible en: https://ethereum- homestead-es.readthedocs.io/en/latest/introduction/what-is- ethereum.html

[9] Aplicaciones, utilidad y casos de uso del blockchain com ejemplos de 2020. [En línea]. Disponible en: https://criptomoneda.ninja/aplicaciones-blockchain/

[10] Ciro Espinoza Montes, Metodología de Investigación Tecnológica: Pensando en Sistemas, 2010.

[11] Qué es Scrum, Proyectos ágiles. [En línea]. Disponible en: https://proyectosagiles.org/que-es-scrum/

[12] Alexander Menzinsky, Gertrudis López, Juan Palacio, Scrum Manager V. 2.6, Julio 2016.

[13] Metodología: Capítulo 3. [En línea]. Disponible en: http://tesis.uson.mx/digital/tesis/docs/22832/Capitulo3.pdf

[14] Gustavo Daniel Gil, Herramientas para implementar LEL y Escenarios (TILS), 2002.

[15] Documentación de Solidity. [En línea]. Disponible en: https://solidity-es.readthedocs.io/es/latest/

[16] Solidity: Todos los recursos sobre el lenguaje de programación de los smart contracts. [En línea]. Disponible en: https://www.miethereum.com/smart-contracts/solidity/

[17] Documentación de Remix, Ethereum-IDE. [En línea]. Disponible en: https://remix-ide.readthedocs.io/en/latest/

[18] A message to the Community. [En línea]. Disponible en: https://docs.microsoft.com/en-us/previous-versions/visualstudio/foxpro/mt490297(v=msdn.10)?redirectedfrom=MSDN

[20] GAVIN WOOD, ETHEREUM: A SECURE DECENTRALISED

GENERALISED TRANSACTION LEDGER, abril 2014.

[21] Kerlinger, Investigación del comportamiento , 2002