

Creación de una criptomoneda que funcione como token de utilidad dentro de un ecosistema de empresas, permitiendo el pago de productos y servicios

Creating a cryptocurrency which will work as a utility token within a business ecosystem, making possible the payment of products and services

*Santiago Mancini¹, Federico Scuotegazza², Luis Gurméndez³,
Rafael Sotelo⁴, Marcos Begérez⁵, Gabriel Álvarez⁶*

Recibido: Abril 2019

Aprobado: Mayo 2021

Resumen - Con el creciente avance de las tecnologías de la información, cada vez aparecen más ramas de la materia. Comunidades de muchas partes del mundo se encuentran investigando y desarrollando en diferentes áreas de la informática con el fin de encontrar soluciones innovadoras. Blockchain es una de estas ramas que busca la transparencia, seguridad, y descentralización de las bases de datos y sistemas informáticos.

La constante evolución de esta tecnología nos lleva a la necesidad de estudiar los beneficios y herramientas que actualmente están siendo desarrollados en esta área de las tecnologías de la información. Investigar cuáles son las facilidades que nos brinda el hecho de incorporar esta tecnología emergente en nuestros sistemas y soluciones informáticas, cuáles son sus desventajas, así como también evaluar su posibilidad de integrarse con otras herramientas, son algunas de las inquietudes que surgen inmediatamente.

Este proyecto consistió en investigar las soluciones más populares de blockchain e implementar el despliegue de una criptomoneda que permita realizar transacciones entre un grupo cerrado de organizaciones.

Palabras Clave: Blockchain, Criptomoneda; Sistema de pagos empresariales; Hyperledger; Token Digital.

Summary – *With the recent advances on information and communication technologies, every year we can see more and more branches in this topic. Communities from all over the world are continuously investigating and developing in different categories of the informatics with the idea of finding new innovative solutions to all type of problems. Blockchain is one of this branches that tries to solve transparency, security, and decentralization issues from the databases and informatic systems.*

The constant evolution of this technology leads us to the necessity of studying the benefits and tools that are being developed in this topic of the information and communication technologies. Investigate what are the facilities that we may have by using this emerging technology in our systems and informatics solutions, what are the disadvantages, as well as to evaluate the possibility of integrating it to other tools, are some of the restlessness that immediately come up.

This article arises from a grade thesis made by the authors for a technology center ICT4V (Information and Communication Technologies For Verticals), and it intends to resume what was investigated and learnt during the project.

Keywords: Blockchain, Cryptocurrency; Business Payment System; Hyperledger; Digital Token.

¹ Ingeniero Informático. Universidad de Montevideo, smancini@correo.um.edu.uy , ORCID iD: <https://orcid.org/0000-0002-6096-6777>

² Ingeniero Informático. Universidad de Montevideo, fscuotegazza@correo.um.edu.uy , ORCID iD: <https://orcid.org/0000-0002-7872-5285>

³ Ingeniero Informático. Universidad de Montevideo, lgurmendez@correo.um.edu.uy , ORCID iD: <https://orcid.org/0000-0002-9508-0589>

⁴ Doctor Ingeniero Electricista. Universidad de Montevideo, rsotelo@um.edu.uy , ORCID iD: <https://orcid.org/0000-0002-4034-3177>

⁵ Ingeniero Informático. Bantotal, mbegerez@dlya.com.uy , ORCID iD: <https://orcid.org/0000-0002-4234-9951>

⁶ Ingeniero Informático. Bantotal, galvarez@dlya.com.uy , ORCID iD: <https://orcid.org/0000-0002-5515-5646>

1. Introducción.- Con el creciente avance de las tecnologías de la información, cada vez aparecen más ramas de la materia. Comunidades de muchas partes del mundo se encuentran investigando y desarrollando en diferentes áreas de la informática con el fin de encontrar soluciones innovadoras a diversos. Blockchain es una de estas ramas que busca la transparencia, seguridad, y descentralización de las bases de datos y sistemas informáticos.

Blockchain es una tecnología en pleno desarrollo. A pesar de esto, existen muchas soluciones que utilizan blockchain como la tecnología base. Al redactarse este artículo, no hay muchos desarrolladores aprendiendo esta tecnología ni desarrollando activamente utilizando herramientas relacionadas.

Este proyecto consistió en el despliegue de una criptomoneda y su uso como token que permite realizar transacciones entre un grupo cerrado de organizaciones para dar cumplimiento a acuerdos establecidos entre las mismas y representados digitalmente bajo la forma de programas (contratos inteligentes). Más concretamente, el sistema diseñado permite establecer acuerdos y realizar transacciones entre los socios de ICT4V, un Centro Tecnológico de TICs que funciona en el Parque del LATU y que nuclea a cuatro universidades uruguayas, agencias públicas de gobierno, y varias empresas privadas. Una de esas empresas es De Larrobla & Asociados, vinculada al mundo bancario, que desarrolla el core bancario Bantotal, y fue la contraparte del proyecto en ICT4V.

El proyecto brinda a un grupo de organizaciones una forma alternativa para el intercambio de servicios. También implica un estudio en el campo de la tecnología blockchain, la cual es de suma importancia para el centro ICT4V debido al gran impacto y crecimiento que está teniendo.

El producto final cuenta con una solución de blockchain, donde se guarda toda la información con respecto a las transacciones. Una REST API que funciona como el backend con el cual se va a comunicar todos los clientes, y de ser necesario consultar la blockchain también se hará desde aquí. Y por último una webapp, que es el medio por el cual los clientes se conectan para interactuar con el sistema.

En este artículo se describirá lo realizado durante el proyecto. Se introducirán las tecnologías estudiadas y se verá un ejemplo práctico realizado para cada una de ellas. Se exponen ventajas y desventajas de usar una sobre otra. También se especifica cuál de las herramientas se usó para el desarrollo de la consigna, cuáles fueron los motivos de su elección, y cómo se trabajó con dicha tecnología.

Se espera que el artículo sirva como introducción a la tecnología blockchain, dando a conocer sus particularidades y facilidades que brinda al implementarse en un sistema informático.

2. Blockchain.- En 2008 Satoshi Nakamoto publica un *whitepaper* [39] introduciendo un concepto que solucionaría el problema de tener una entidad centralizada para poder realizar transacciones entre varios nodos. Propone un sistema peer-to-peer de transferencia de valor que se conoce como Bitcoin. A este concepto de almacenamiento de información distribuida se le llamó blockchain y es la tecnología que hace posible la implementación de criptomonedas. Blockchain o cadena de bloques es, en simples palabras, una base de datos distribuida en donde se agrupan y guarda registro de todas las transacciones de forma ordenada dentro de bloques, manteniendo cierta estructura que hace imposible la modificación de un registro. Como especifica el nombre, estos bloques están vinculados entre sí para generar esta estructura.

Existen varias implementaciones de blockchain en donde varían algoritmos de consenso, algoritmos de encriptado, tipos de nodos, roles, permisos, etc [40]. Pero todas estas soluciones siguen la misma estructura, la cadena de bloques. Se utilizará a Bitcoin como ejemplo de una implementación de blockchain. En donde se explicará, a grandes rasgos, las funcionalidades principales y cómo estas alcanzan la integridad de datos distribuida, utilizando una cadena de bloques.

Cuando se hace referencia a la entidad bloque, este es simplemente una estructura de datos que contiene un header y una lista de transacciones. Dentro del header se puede encontrar información del bloque en sí, como:

- Número de bloque
- Número de transacciones del bloque
- Fecha de creación (Timestamp)
- Nonce
- Dificultad del bloque
- Hash del bloque anterior
- Merkle Root

La lista de transacciones, es simplemente una lista que contiene los datos relevantes que se desean guardar dentro de la blockchain.

Para una mejor comprensión de esta tecnología consideramos necesario explicar el concepto de hashing. Un hash⁷ es una secuencia de caracteres que se obtiene al aplicar una función de hashing a una cantidad arbitraria de elementos de datos. El hash tiene la particularidad de mantener siempre una cantidad fija de caracteres, y es imposible que a partir de un hash se consigan los datos que lo generaron, es decir una función inversa. Asimismo, cualquier cambio en uno de los valores de entrada de función de hash genera un hash completamente distinto. En Bitcoin la función de hashing que se utiliza es el algoritmo SHA-256⁸ y se le llama hash de un bloque a el resultado de aplicar esta función usando como entrada, el hash del bloque anterior, el número de bloque, las transacciones y nonce.

La cadena de bloques es una concatenación de bloques en donde se utiliza el hash del bloque anterior para generar el hash del bloque actual. El primer bloque se llama el bloque génesis y tiene un número de bloque igual a 0 y es el único caso que no tiene referencia al bloque anterior.

Para agregar un bloque a la cadena se inicia un proceso de minería, en donde varios nodos llamados mineros, compiten para determinar quién es el que generará el siguiente bloque. En Bitcoin la creación de bloques se determina a través de un algoritmo de PoW (Proof of Work), que funciona de la siguiente manera. Los mineros están constantemente aplicando el algoritmo SHA-256 utilizando como valor de entrada una concatenación del hash del bloque anterior, el número de bloque, las transacciones y un nonce. El hash del bloque anterior, así como el número de bloque y las transacciones, son constantes para cada bloque y lo único que se varía es el valor del nonce, que lo inventan los mineros para obtener un hash distinto.

El primer minero que encuentre un nonce el cual genera un hash objetivo⁹, es el ganador. Este minero genera el bloque nuevo, guardando el nonce en el header del bloque y algunas de las transacciones que estén sin validar y lo transmite a todos los nodos contiguos en la red. Además el minero crea una transacción especial, donde se agrega a su propia cuenta una cantidad de bitcoins como comisión; actualmente son 6,5 y se reduce a la mitad cada 4 años. Esta es la motivación de los mineros para generar bloques. Los mineros además cumplen un rol muy importante, que es el de validar las transacciones o ejecutar contratos inteligentes.

Los contratos inteligentes o smart contracts son un script o una pieza de software que ejecuta cierta lógica, que en algunos casos puede representar un acuerdo entre partes como si fuese un contrato normal, donde se define lo que se puede hacer, cómo hacerlo, etc. Estos contratos están escritos en un lenguaje de programación y residen dentro de la blockchain. En otras palabras, son scripts o programas informáticos que corren dentro de la blockchain. Estos contratos se ejecutan automática y autónomamente sin intermediarios, lo cual brinda infinitas posibilidades y oportunidades. Cada blockchain tiene su propia implementación de smart contracts, como por ejemplo, Ethereum y Bitcoin. Ethereum está fuertemente basado en smart contracts y soporta smart contracts turing complete, es decir que brinda todas las operaciones, en cambio Bitcoin no. En Ethereum los smart contracts juegan un rol muy importante ya que brindan la posibilidad de crear tokens y a impulsado fuertemente la creación de las Initial Coin Offering (ICO). Las ICO son una forma de financiar proyectos en el cual se venden tokens de una moneda que esta es una etapa temprana. Estas monedas o tokens viven dentro de otras blockchains, son en definitiva smart contracts.

Para obtener más información en profundidad de la tecnología blockchain, se recomienda leer el artículo del enlace [42] de la bibliografía, en donde se explican conceptos como el “Longest Chain Rule”.

3. Desarrollo realizado.- Para el desarrollo de sistemas basados en blockchain están disponibles distintas plataformas, la mayoría de ellas open source. El equipo debió realizar un estudio previo de las posibles tecnologías a utilizar. Se resolvió escoger tres posibles plataformas y estudiarlas no sólo teóricamente sino también con un breve ejemplo práctico.

Se decidió estudiar las tecnologías HyperLedger [11], Ethereum [12] y Quorum [28]. Para tomar esta decisión, el equipo evaluó los siguientes puntos de cada tecnología (los primeros puntos pesan más en la decisión, siendo Adaptabilidad al problema el punto a destacar):

1. Adaptabilidad al problema
2. Documentación
3. Mantenimiento
4. Apoyo de la comunidad

⁷ Hash: El resultado de aplicar una función de hash a un conjunto arbitrario de elementos de datos, representado en una serie de caracteres de texto de longitud fija.

⁸ SHA-256: Algoritmo o función de hash, que transforma un conjunto arbitrario de elementos de datos, como puede ser un fichero de texto, en un único valor de longitud fija, llamado hash.

⁹ Hash objetivo: Un hash que cumple cierta regla. Ej. Comience con cinco 0s consecutivos.

5. Empresas y organizaciones detrás de la plataforma

Luego de haber estudiado ampliamente cada una de las tres herramientas, realizado los trabajos prácticos, y debatido las ventajas y desventajas de utilizar una u otra, el equipo logró construir la siguiente tabla:

	Hyperledger Fabric	Ethereum	Quorum
Adaptabilidad al problema	Alta	Media	Alta
Compañía detrás	IBM, Linux Foundation	No	JPMorgan
Permisado	Si	No	Si
Consenso	Acepta distintas variedades	Basada en Mining Proof of work (PoW)	Raft Consensus
Código de Smart Contract	Javascript, Golang, Java	Solidity	Solidity
Moneda	Ninguna. Se puede definir una en la lógica	Ether	Ninguna
Rendimiento	>2000tps ¹²	~200tps ¹³	~100tps ¹⁴
Privacidad de las transacciones	Soportada bajo algunas condiciones	No	Si
Documentación	Buena documentación. Actividad en internet por parte de los desarrolladores	Excelente documentación. Mucha actividad en internet de parte de la comunidad	Poca documentación. No se encontró mucha actividad de los desarrolladores.

Tabla I. Muestra las propiedades y puntos comparados de las diferentes tecnologías estudiadas

Puede ser de interés leer el artículo de la revista Academic de la cita [43] de la bibliografía, en donde se hace una comparación reciente de las diferentes plataformas blockchain.

La plataforma elegida a utilizar es **Hyperledger**.

Se entendió que el problema era fácilmente solucionable con Hyperledger Fabric o Quorum. Basta con definir un balance y agregar un costo para las transacciones que impliquen un intercambio de servicios. En Ethereum esto se resolvería con los Ether, pero el equipo se encontraría con el problema del costo que lleva realizar transacciones que no sean de intercambio de servicios ya que en Ethereum todas las transacciones tienen un costo de “gas” asociado para que queden plasmadas en la blockchain.

La herramienta está construida especialmente para problemas muy parecidos al planteado en la propuesta. También cuenta con una empresa detrás como IBM y el apoyo de The Linux Foundation.

La privacidad en las transacciones que presenta Hyperledger se cree que es muy importante. Ya que en Ethereum no existe ésta privacidad, todas las transacciones son públicas y se pueden rastrear. En Ethereum también es obligatorio el uso de Ether, mientras que en Hyperledger o Quorum no es necesario.

Hyperledger también cuenta con una mejor performance y fácil conexión con aplicaciones Angular y Node.js. Lo que facilita, en caso de ser necesario, el desarrollo de éstos para la solución.

Si bien Quorum era una buena opción, esta es una herramienta muy nueva, y como todo software nuevo, no es recomendable adaptarlo en un ambiente de producción.

Una vez que se eligió la tecnología para trabajar en la solución final, se pensó en la posible arquitectura para el sistema. Luego de evaluar distintas alternativas, se llegó a que la arquitectura sería la siguiente:

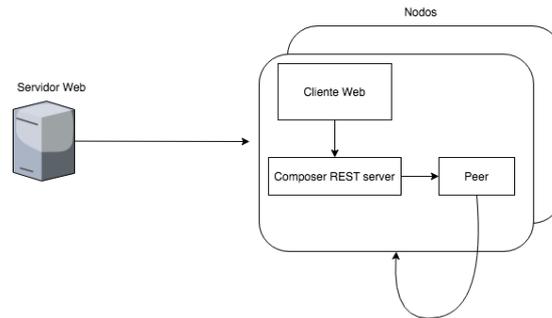


Figura I. Cuarta arquitectura

Las ventajas de esta solución se encuentran mayormente en la descentralización de la API. Esto permite que no se tenga que confiar en la disponibilidad de ningún nodo central, o que exista uno. Por lo que esto además hace que los nodos tengan las mismas funcionalidades y no haya diferencia entre ellos. Esto es una ventaja ya que es altamente escalable.

En marcos generales el funcionamiento del sistema sería el siguiente. Existen varios nodos conectados entre ellos, en donde cada uno corre contenedores de Docker que ejecutan componentes de la red. Cada nodo debe de tener al menos un peer ejecutando para que este se conecte un Composer REST Server. El cliente obtiene la aplicación web que es servida por el servidor de aplicación. Este servidor puede ser una máquina EC2 corriendo en Amazon, o en cualquier proveedor de infraestructura que permita exponer servicios. La aplicación web realiza las llamadas REST que se corresponden con transacciones dentro de la blockchain. Estas llamadas son manejadas por el Composer REST Server, que se comunica con el Peer y es este el encargado de difundir la transacción. Una vez que la transacción forma parte de un bloque válido, esta se ejecuta y el estado de la blockchain cambia en todos los nodos.

Se utilizó la tecnología Hyperledger Composer para el desarrollo de la solución. Si bien esta arquitectura está pensada con el supuesto que la tecnología usada es Hyperledger Composer, la misma sirve exactamente igual para Hyperledger Fabric. Ya que Hyperledger Composer es una solución basada en Hyperledger Fabric cuyo propósito es facilitar el desarrollo de la infraestructura de la blockchain. La gran diferencia es que se abstrae el servidor local que expone los servicios REST, o las transacciones y se crea un propio servidor escrito en Node.js. Este servidor expondrá también servicios REST que consume localmente la aplicación web.

Básicamente, los modelos [9] que se programaron fueron los que muestra la Figura II:

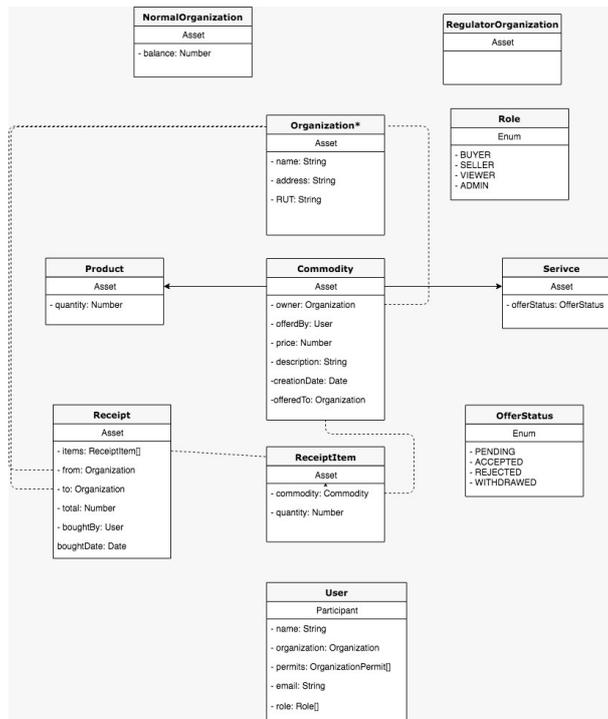


Figura II. Esquema del modelado para la solución al objetivo general

Los usuarios tienen distintos permisos dentro de la blockchain dependiendo de su rol y sus condiciones. La Access Control List [9] que se programó se puede representar a gran escala en la Figura III:

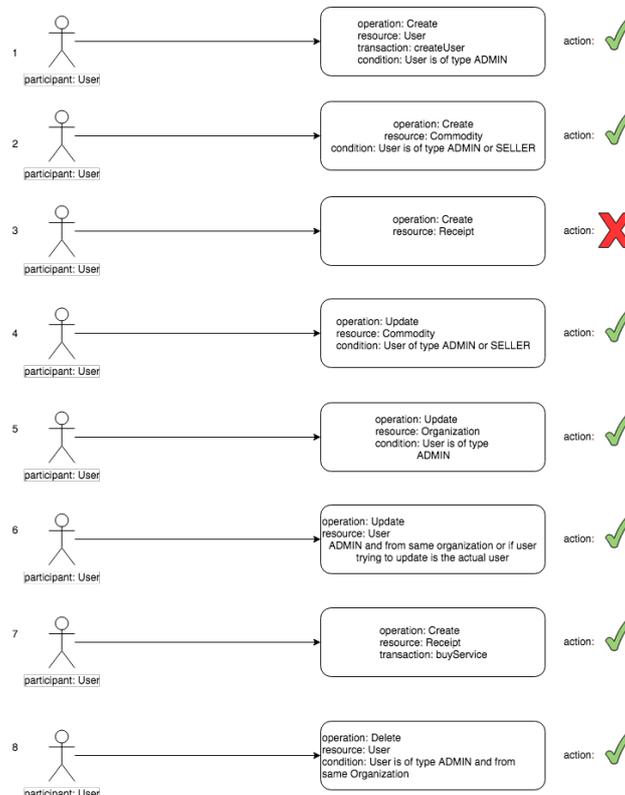


Figura III. Diagrama de permisos.

Lo que se pretendía desarrollar primordialmente, es la compra y venta de un servicio. Con los modelos y los permisos mencionados anteriormente, podemos ver el funcionamiento de la compra y venta de un servicio en el siguiente diagrama:

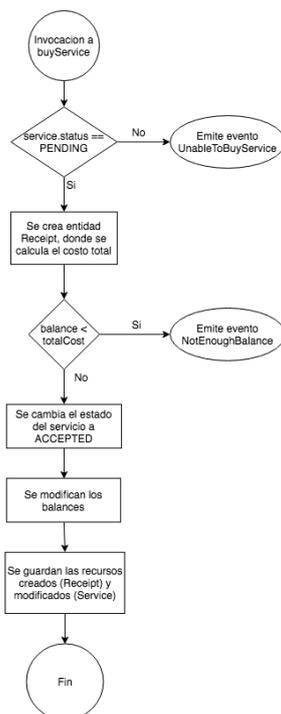


Figura IV. Diagrama de flujo de la compra de un servicio

Una vez terminado el desarrollo de la lógica de la Blockchain, es necesario el despliegue de la blockchain en varios nodos. Se pretende llegar a visualizar la red en al menos dos máquinas (virtuales o físicas) y validar la correcta sincronización constante de lo que ocurre.

Cuando se hacen modificaciones en la blockchain, un peer se comunica con un solo nodo de la red. Luego, este nodo deberá comunicar esta modificación a los demás nodos para así mantener sincronizados todos los peers de la red.

Esto es el caso ideal de una solución blockchain. Una solución multi-peer en donde existe una clara descentralización tanto de la información en la red como el procesamiento de las transacciones.

Para la realización de esta tarea, el equipo encontró dos posibles alternativas. La primera a través de la herramienta IBM Cloud [27] o la segunda, realizando el despliegue con las herramientas de Fabric y Composer.

La IBM Cloud es una herramienta que ofrece la compañía IBM para realizar distintas tareas en la nube. Una de las funcionalidades que brinda es la posibilidad de realizar el deploy de contratos inteligentes con extensión .bna (smartcontracts de Hyperledger Composer) en muy poco tiempo.

Se evaluó esta alternativa tomando en cuenta el aporte que tendría hacer el despliegue de esa manera y el costo de tomar esa alternativa.

Con respecto al aporte de realizar el despliegue a través de la IBM Cloud, si bien se cree que el deploy multi-peer de la solución es un punto importante de la investigación, se concluyó que de esta forma no aportaría grandes conocimientos técnicos.

Con respecto al costo, si bien IBM Cloud ofrece un saldo gratis para comenzar, de todas formas se requieren recursos económicos adicionales para usar esta solución. Además, la proyección económica a futuro de usar esta opción suponía grandes gastos. Se calificó esta solución como una alternativa costosa monetariamente.

El equipo se reunió con representantes de IBM para obtener más información tanto técnica como económica de los planes disponibles.

Se decidió por lo dicho previamente que se debía buscar y evaluar otra alternativa para la realización del despliegue multi-nodo en Hyperledger Composer.

La segunda alternativa que se evaluó fue hacer el despliegue de la blockchain localmente y en dos máquinas distintas.

En un principio el equipo entendió que lo ideal sería desplegar el .bna en dos máquinas virtuales que corren en diferentes hosts, y configurar dichas máquinas para que se sincronicen correctamente.

Luego de una extensa investigación, el equipo encontró que Hyperledger Composer no puede ser desplegado en múltiples nodos sincronizados. Por definición, Composer es un set de herramientas para trabajar sobre Hyperledger Fabric que facilita el desarrollo de aplicaciones que deben ser desplegadas luego en una red blockchain como Hyperledger Fabric[44], y este set de herramientas no incluye alguna interfaz para lograr el despliegue en múltiples hosts. Se llegó a que era necesario primero desplegar Hyperledger Fabric en al menos dos nodos, y luego usar el .bna generado por Composer, instalándolo en estos nodos [41].

Como se explica en los resultados obtenidos, no fue posible realizar el despliegue multi-peer con la herramienta localmente en dos máquinas ya que ésta no lo permite fácilmente. Debido a ello se decidió que esto se encontraba fuera del alcance de este proyecto.

Luego de desplegada la blockchain, se programó una aplicación web para que los usuarios finales puedan trabajar con la blockchain transparentemente. A su vez, se hizo uso del Composer REST Server [10] para la comunicación entre la blockchain y la aplicación web.

4. Resultados.- Se logró investigar en profundidad la innovadora y novedosa tecnología blockchain. Se estudiaron tres plataformas diferentes, se logró una extensa comparación y se escogió una de ellas para desarrollar la solución final. La plataforma elegida fue HyperLedger Fabric utilizando el módulo Hyperledger Composer para el desarrollo de la lógica.

Se desarrolló también el servidor que se comunica con la blockchain, y la aplicación web para el usuario final. Es posible realizar transferencias y operaciones con la blockchain desde la aplicación web correctamente.

La investigación fue fructífera al lograr comprender el estado actual de las diferentes tecnologías disponibles en el área, y se espera que este trabajo sirva como base para futuras investigaciones y desarrollos.

Por otro lado, también es válido citar un comentario descubierto mientras se investigaba sobre el tema. Simon Stone, líder del proyecto Composer para IBM, expuso en la página oficial de Hyperledger que la herramienta no continuará siendo mejorada. Si bien seguirá tratando de mantenerse lo que ya se tiene, no se continuará agregando más funcionalidades. Se deja claro que IBM en general no dispondrá más recursos para el desarrollo de Composer, y que el nuevo camino será hacia Hyperledger Fabric.

“However - we at IBM believe that there are some fundamental problems with the architecture and design of Composer, as it is today, that have made us reconsider our future direction and plans.

...

At this time, IBM has decided to reduce the investment it makes towards developing Composer, in order to focus the team on directly delivering improvements into Fabric. ...”

Este comentario se publicó en los foros oficiales de Hyperledger [38] el día treinta de agosto de dos mil dieciocho. Se cree que esto demuestra un claro indicio de que Hyperledger Composer seguirá manteniéndose como una herramienta de desarrollo, y no apuntada a ambientes de producción.

5. Conclusiones.- Una vez finalizado el proyecto, en retrospectiva, se puede apreciar cómo durante el transcurso del mismo se agregó valor tanto a los miembros del equipo de la Universidad de Montevideo, como al centro ICT4V y a los representantes de Bantotal.

Desde un comienzo el objetivo principal fue tener una solución basada en la tecnología de blockchain para el intercambio de servicios entre los diferentes miembros de ICT4V. Si bien no se pudo implementar la solución a nivel productivo debido a los problemas que se encontraron a la hora de realizar el despliegue, se provee el sistema base desde donde se puede seguir iterando para mejorar la solución, y eventualmente llegar a desplegar el sistema en ambiente productivo.

Otro de los objetivos, que el centro ICT4V comience a conocer las distintas tecnologías de blockchain y exista el intercambio de este conocimiento entre los miembros de la organización, fue cumplido. Se dio a conocer los hallazgos de este proyecto con distintos becarios que se encontraban comenzando proyectos con tecnologías como blockchain, Hyperledger Composer, Hyperledger Fabric. Advirtiendo sobre los distintos problemas encontrados durante el transcurso de esta solución y los beneficios de utilizar estas tecnologías.

6. Agradecimientos.- Este proyecto fue realizado con el apoyo de ICT4V (www.ict4v.org)

7. Referencias.

- [1] PasswordGenerator.net, “SHA256 Hash Generator”. <https://passwordgenerator.net/sha256-hash-generator/>
- [2] Anders, “Blockchain” <https://anders.com/blockchain/blockchain.html>
- [3] Instalar pre-requisitos de Hyperledger Composer, Hyperledger. <https://hyperledger.github.io/composer/v0.16/installing/installing-prereqs.html>
- [4] Instalar el cliente de Composer e instalar Hyperledger Fabric, Hyperledger. <https://hyperledger.github.io/composer/v0.16/installing/development-tools.html>
- [5] Yeoman Introduction, Yeoman. <http://yeoman.io/>
- [6] Loopback Introduction, Loopback. <https://loopback.io/>
- [7] Passport.js Introduction, Passport.js. <http://www.passportjs.org/>
- [8] Hyperledger Fabric Architecture, Hyperledger. <https://hyperledger-fabric.readthedocs.io/en/release-1.2/arch-deep-dive.html>
- [9] Definición de modelos en Composer, Hyperledger. https://hyperledger.github.io/composer/latest/reference/cto_language
- [10] Generando una API REST en Composer, Hyperledger. <https://hyperledger.github.io/composer/v0.16/integrating/getting-started-rest-api>
- [11] Hyperledger Fabric Propiedades, Hyperledger. <https://hyperledger-fabric.readthedocs.io/en/release-1.3/arch-deep-dive.html>
- [12] Gavin Wood (2013), “Formal definition of the Ethereum protocol”, Github. <https://github.com/ethereum/yellowpaper>
- [13] Amy Yu, “Ethereum Development Tutorial”, Github. <https://github.com/ethereum/wiki/wiki/Ethereum-Development-Tutorial>
- [14] Alyssa Hertig, “How do Ethereum Smart Contracts Work”, Coin Desk. <https://www.coindesk.com/information/ethereum-smart-contracts-work/>
- [15] Dete (Sep 2017), “ERC: Non-fungible Token Standard”, Github. <https://github.com/ethereum/eips/issues/721>
- [16] James Ray, “A Next-Generation Smart Contract and Decentralized Application Platform”. Github <https://github.com/ethereum/wiki/wiki/White-Paper>
- [17] Alejandro Nieto (2017). “¿Qué es una ICO? La tecnología que está revolucionando la financiación empresarial”, Xataka. <https://www.xataka.com/empresas-y-economia/que-es-una-ico-la-tecnologia-que-esta-revolucionando-la-financiacion-empresarial>
- [18] Alyssa Hertig, “How Ethereum Mining works”. Coin Desk. <https://www.coindesk.com/information/ethereum-mining-works/>
- [19] Dmitry Melanchenko (May 2018) “Containers Part 2: What is docker-compose?”, Medium. <https://engineering.salesforce.com/containers-part-2-what-is-docker-compose-50a3a5685cc4>
- [20] Abdul Wahab (Feb 2018), Hyperledger Fabric on Multiple Hosts. Medium. <https://medium.com/@wahabjawed/hyperledger-fabric-on-multiple-hosts-a33b08ef24f>
- [21] How nodes work, Docker. <https://docs.docker.com/engine/swarm/how-swarm-mode-works/nodes/#manager-nodes>
- [22] Chernov y Gutierrez, (Ago 2018), Deploying a Multi-Node Hyperledger Fabric Network in 5 Steps. Altoros. <https://www.altoros.com/blog/deploying-a-multi-node-hyperledger-fabric-network-in-5-steps/>
- [23] Varun Raj (Mar 2018), “Setting up a Blockchain Business Network With Hyperledger Fabric & Composer Running in Multiple Physical Machine”, Medium. <https://medium.com/hyperlegendary/setting-up-a-blockchain-business-network-with-hyperledger-fabric-composer-running-in-multiple-bf4e38b6c6>
- [24] Hyperledger, “Architecture Explained”. <https://hyperledger-fabric.readthedocs.io/en/release-1.3/arch-deep-dive.html>
- [25] Mallikarjun Sarvepalli (Oct 2018), Hyperledger Fabric 1.2 on Multiple Hosts using Docker Swarm and Compose, Medium. <https://medium.com/@mallikarvepalli/hyperledger-fabric-1-2-on-multiple-hosts-using-docker-swarm-and-compose-11c13635e69e>
- [26] VirtualBox Introduction, VirtualBox. <https://www.virtualbox.org/>

- [27] IBM Cloud - Blockchain Platform, IBM. <https://www.ibm.com/cloud/blockchain-platform>
- [28] Quorum Main Page, JP Morgan. <https://www.jpmorgan.com/global/Quorum>
- [29] Blockchain en AWS, Amazon Web Services. https://aws.amazon.com/blockchain/?nc1=h_ls
- [30] CouchDB Main, CouchDB. <http://couchdb.apache.org/>
- [31] Docker Properties, Docker. <https://www.docker.com/why-docker>
- [32] Material Design, Material. <https://material.io/design/>
- [33] Hyperledger Certificate Authority, Hyperledger. <https://openblockchain.readthedocs.io/en/latest/Setup/ca-setup/>
- [34] NodeJS Documentation, NodeJS. <https://nodejs.org/en/docs/>
- [35] NPM Documentation, NPM. <https://docs.npmjs.com/>
- [36] NVM Documentation, NVM. <https://github.com/creationix/nvm>
- [37] Angular Documentation, Angular. <https://angular.io/docs>
- [38] Composer TSC update as of 30 Aug 2016,
https://lists.hyperledger.org/g/composer/topic/composer_tsc_update/25106848?p=,,,20,0,0,0::recentpostdate%2Fsticky,,,20,2,0,25106848
- [39] Bitcoin: A Peer-to-Peer Electronic Cash System. <https://bitcoin.org/bitcoin.pdf>
- [40] What Different Types of Blockchains are There? <https://dragonchain.com/blog/differences-between-public-private-blockchains>
- [41] LahiruE (2018). StackOverflow. Hyperledger composer in multiple host (local network) [Online]. <https://stackoverflow.com/questions/52286838/hyperledger-composer-in-multiple-host-local-network>
- [42] The limits of trust-free systems: A literature review on blockchain technology and trust in the sharing economy. <https://www.sciencedirect.com/science/article/pii/S1567422318300292>
- [43] Comparison of blockchain platforms: a systematic review and healthcare examples. <https://academic.oup.com/jamia/article/26/5/462/5419321>
- [44] Documentación de Hyperledger Composer. <https://hyperledger.github.io/composer/latest/introduction/introduction.html>