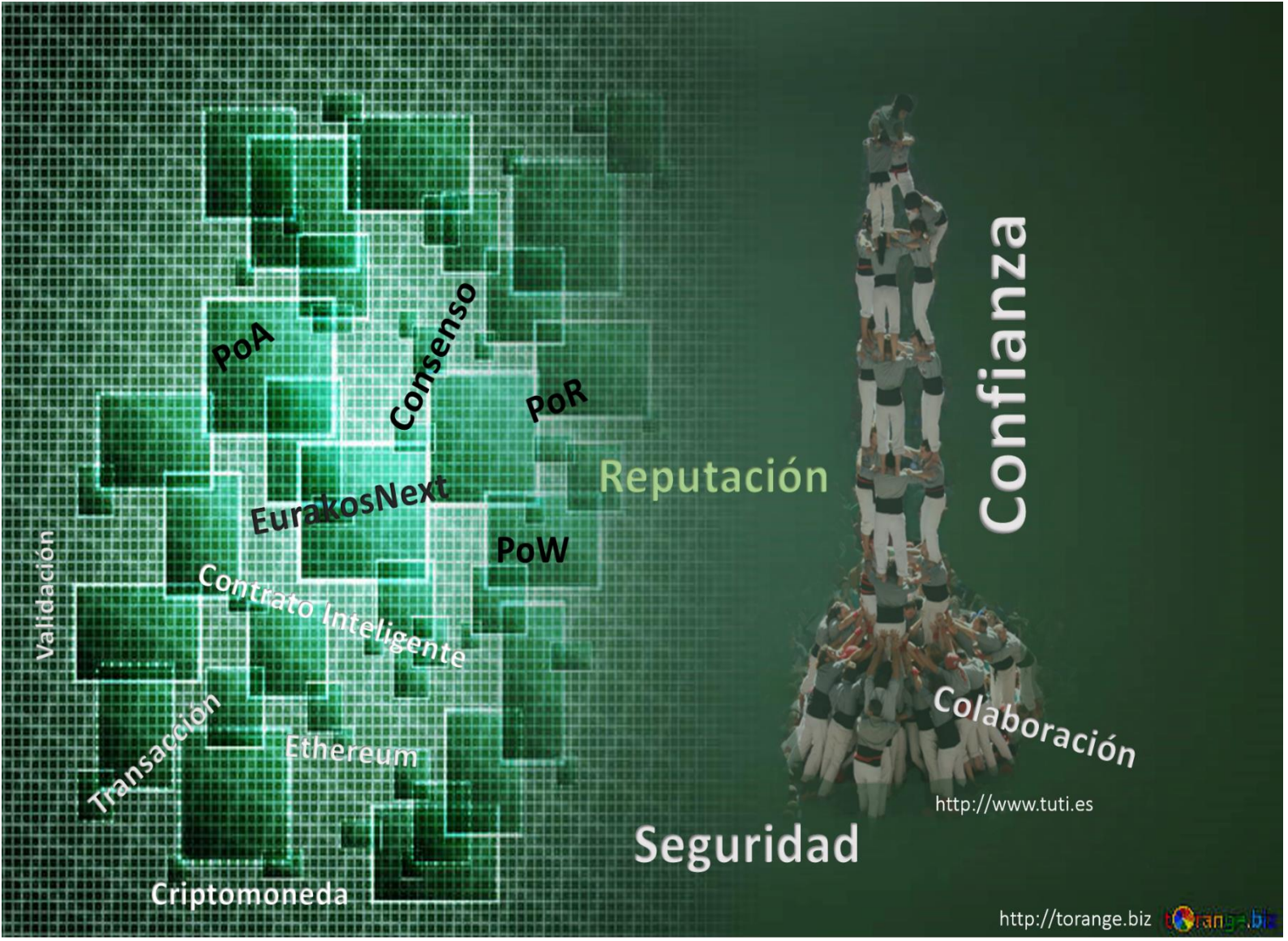
REDUCCIÓN DE BARRERAS DE ADOPCIÓN DE CRIPTOMONEDAS: INNOVACIÓN TECNOLÓGICA EN LOS PROCESOS DE VALIDACIÓN DE TRANSACCIONES

Paulo Nicolás Carrillo Peña



TESIS DOCTORAL

Reducción de barreras de adopción de criptomonedas: innovación tecnológica en los procesos de validación de transacciones

Paulo Nicolás Carrillo Peña

Universitat de Girona

2019

Universitat de Girona

TESIS DOCTORAL

Reducción de barreras de adopción de criptomonedas: innovación tecnológica en los procesos de validación de transacciones

POR: PAULO NICOLÁS CARRILLO PEÑA

2019

PROGRAMA DE DOCTORADO EN TECNOLOGÍA

DIRECTORES: DR. JOSEP LLUÍS DE LA ROSA I ESTEVA

DRA. CLARA INÉS PEÑA DE CARRILLO

Tesis presentada como requisito parcial para optar al título de

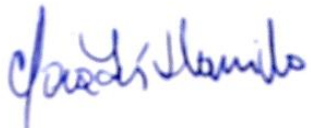
Doctor en Tecnología por la Universitat de Girona

Universitat de Girona

El Dr Josep Lluis de la Rosa i Esteva, de la Universitat de Girona y la Dra Clara Ines Peña de Carrillo, de la Universidad Autónoma de Bucaramanga,

DECLARAN:

Que el trabajo titulado *Reducción de barreras de adopción de criptomonedas: innovación tecnológica en los procesos de validación de transacciones* que presenta el Sr. Paulo Nicolás Carrillo Peña ha sido realizado bajo nuestra dirección,



Dr Josep LLuis de la Rosa i esteva

Dra. Clara Inés Peña deCarrillo

Girona, Mayo de 2019

Esta tesis fue financiada en parte por el proyecto Europeo Vircoin2SME del Programa de Investigación e Innovación Horizon 2020 (2014-2020), en virtud del acuerdo de subvención Marie Skłodowska-Curie No. 654767.

AGRADECIMIENTOS

Primero que todo quisiera agradecer a mi supervisor el Doctor Josep Lluís de la Rosa por incorporarme en el mundo de las monedas digitales y complementarias y por los lineamientos y la paciencia ofrecidos a lo largo del proyecto.

A mi madre y también supervisora, la Doctora Clara Inés Peña de Carrillo por todo el apoyo metodológico brindado para el desarrollo de la tesis y por todo el soporte externo a la tesis para poder alcanzar con éxito la meta planteada.

En general a mis padres los Doctores Gilberto Carrillo Caicedo y Clara Inés Peña de Carrillo por siempre estar ahí pendientes para ayudar, para aconsejar, para recomendar y para apoyar económicamente mi sostenimiento durante estos años de investigación.

Adicionalmente expreso mi especial reconocimiento y aprecio a Julio Gisbert y Rogelio Segovia (Q.E.P.D) que desde su experticia me ayudaron a contextualizar los conceptos para concretar el estado del arte de la presente tesis.

LISTA DE PUBLICACIONES

Capítulo de libro en proceso de publicación

1. Peña de Carrillo, C.I., De La Rosa i Esteva, J.Ll. y, Carrillo Peña, P.N. (2019). Monedas Sociales-Complementarias-Virtuales: Identificación de barreras para su adopción en contextos de PYMES y consumidores. Red de Investigadores de la Asociación Colombiana de Facultades de Administración RIACO-ASCOLFA, Colombia.

Artículos e informes publicados

1. Peña de Carrillo, C., De la Rosa i Esteva, J.Ll., Carrillo Peña, P.N. and Pharow, Peter. (2018). Identification of barriers and solutions for adoption of social, complementary and/or virtual currencies. *International Journal of Community Currency Research 2018*, *Volume* (22) (Summer), 125-140, ISSN 1325-9547. Doi: http://dx.doi.org/10.15133/j.ijccr.2018.020.

2. Peña de Carrillo, C.I., Acosta, R., and, Carrillo Peña, P.N. (Mayo, 2017). Design Thinking and Gamification as strategies to build a social virtual currency for adaptive healthy habits promotion at the workplace. IV Conferencia Internacional Monedas Sociales y Complementarias, Dinero, Conciencia y Valores para el Cambio Social. Universidad Oberta de Catalunya, Barcelona, España, 12p.

3. Peña de Carrillo, C.I., De la Rosa, J. Ll., Pharow, P., Lucht, M., Carrillo, P.N., Santiago, D., Smets, W., Félix, J., and, Tito, D. (Mayo, 2017). Identifying barriers to adoption of social, complementary and/or virtual currencies. IV Conferencia Internacional Monedas Sociales y Complementarias, Dinero, Conciencia y Valores para el Cambio Social. Universidad Oberta de Catalunya, Barcelona, España, 27p.

4. Peña de Carrillo, C.I., De la Rosa, J. Ll., Pharow, P., Lucht, M., Carrillo, P.N., Santiago, D., Smets, W., Félix, J., and, Tito, D. (Mayo, 2017). Guidelines/Proposals for developing sustainable business models based on the use of complementary and/or virtual currencies in a networked society. IV Conferencia Internacional Monedas Sociales y Complementarias, Dinero, Conciencia y Valores para el Cambio Social. Universidad Oberta de Catalunya, Barcelona, España, 27p.

5. Carrillo, P.N., De la Rosa, J.Ll., and Peña de Carrillo, C.I. (Mayo, 2017). Validating EurakosNext cryptocurrency transactions using the Artificial Bees Colony Algorithm – an approach. IV Conferencia Internacional Monedas Sociales y Complementarias, Dinero, Conciencia y Valores para el Cambio Social, Universidad Oberta de Catalunya, Barcelona, España, 23p.

6. Carrillo, P.N., Peña, C.I., De la Rosa, J.Ll. (2016). Eurakos Next: A Cryptocurrency Based on Smart Contracts. En Nebot, A., Binefa, X. and López, R., Artificial Intelligence Research and Development, *Frontiers in Artificial Intelligence and Applications*, *Vol* (288), IOS Press, 221-226. Doi: 10.3233/978-1-61499-696-5-221.

7. Peña de Carrillo, C.I., Carrillo Peña, P.N., and, De la Rosa, J.Ll. (Octubre, 2016). Economía social: un medio para el desarrollo sostenible. 5 Congreso Internacional de Gestión Tecnológica y de la Innovación COGESTEC 2016, Universidad Industrial de Santander, Bucaramanga, Colombia, 1492-1529.

8. Carrillo Peña, P.N., Peña de Carrillo, C.I., and, De la Rosa, J. Ll. (Octubre, 2016). JoinCoin: un entorno de trabajo común para criptomonedas. 5 Congreso Internacional de Gestión Tecnológica y de la Innovación COGESTEC 2016, Universidad Industrial de Santander, Bucaramanga, Colombia, 1622-1649.

9. Carrillo Peña, P.N. (2016). Secondment Report: Project n°: 645767 VirCoin2SME, Marie Sklodowska-Curie Actions MSCA-RISE-2014, 39p.

LISTA DE TÉRMINOS

Árbol de Merkle[1]: En criptografía e informática, un *hash tree* o *Merkle tree* es un árbol en el que cada nodo hoja está etiquetado con el código *hash* de un bloque de datos y cada nodo no hoja está etiquetado con el código *hash* criptográfico de las etiquetas de sus nodos secundarios. Los árboles *hash* permiten una verificación eficiente y segura de los contenidos de grandes estructuras de datos. Un árbol de Merkle se define recursivamente como un árbol binario de listas de códigos *hash* donde el nodo principal es el código *hash* de sus hijos, y los nodos de hoja son códigos *hashes* de los bloques de datos originales.

Big Data[2]: Conjuntos de datos o combinaciones de conjuntos de datos cuyo tamaño (volumen), complejidad (variabilidad) y velocidad de crecimiento, dificultan su captura, gestión, procesamiento o análisis mediante tecnologías y herramientas convencionales, tales como bases de datos relacionales y estadísticas convencionales o paquetes de visualización, dentro del tiempo necesario para que sean útiles. Con una cantidad tan grande de información, los datos pueden ser moldeados o probados de cualquier manera que la empresa considere adecuada, esto le permitirá a las organizaciones identificar los problemas de una forma más comprensible y mejorar la toma de decisiones.

Bitcoin: Fue la primera criptomoneda en el mercado lanzada como software de código abierto en 2009. Es una moneda digital descentralizada sin un banco central o administrador único, que se puede enviar de usuario a usuario perteneciente a la red de Bitcoin, sin la necesidad de intermediarios. Las transacciones se verifican mediante nodos de red a través de la criptografía y se registran en un libro mayor público distribuido denominado *Blockchain*.

Blockchain: o cadena de bloques, es un libro de cuentas en el que los registros (los bloques) están enlazados y cifrados para proteger la seguridad y privacidad de las transacciones. Consiste en una base de datos distribuida y segura (gracias al cifrado) que se puede aplicar a todo tipo de transacciones que no tienen por qué ser necesariamente económicas.

[1] https://en.wikipedia.org/wiki/Merkle_tree
[2] https://www.powerdata.es/big-data

Contrato Inteligente[3]**:** O *Smart Contract*, es una pieza de código (software). No es un contrato verdadero, en el sentido legal. En sus términos más simples, un contrato inteligente es una pieza de código que dos o más partes programan para que ciertas acciones ocurran como resultado de condiciones específicas que se dan o no.

Criptomoneda: Medio digital de intercambio que utiliza criptografía para asegurar las transacciones financieras, controlar la creación de unidades adicionales y verificar la transferencia de activos.

Crowdfunding: Financiamiento de un proyecto o una empresa mediante el recaudo de pequeñas cantidades de dinero de un gran número de personas, generalmente a través de Internet.

DDoS[4]: O ataque distribuido de denegación de servicio (*Distributed denial-of-service*), es un intento malicioso de interrumpir el tráfico normal de un servidor, servicio o red de destino al abrumar al objetivo o la infraestructura que lo rodea con una gran cantidad de tráfico de Internet. Los ataques DDoS logran efectividad al utilizar múltiples sistemas informáticos comprometidos como fuentes de tráfico de ataques. Las máquinas explotadas pueden incluir computadores y otros recursos en red como dispositivos *IoT*.

eBusiness: Uso de las Tecnologías de Información y Comunicación para realizar actividades de negocios.

eGovernment: Uso de las Tecnologías de Información y Comunicación para ofrecer servicios públicos a los ciudadanos y otras personas en un país o región.

eHealth: Práctica de cuidados de la salud apoyada en Tecnologías de Información y Comunicación.

[3] https://medium.com/creativeblockchain/lets-disintermediate-all-the-lawyers-smart-contracts-on-the-blockchain-why-blockchain-matters-to-the-cd031e40a75e
[4] https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/

Ethereum: Plataforma de computación distribuida basada en un *Blockchain* público en donde las aplicaciones se ejecutan exactamente como están programadas sin ninguna posibilidad de fraude, censura o interferencia de terceros.

Eurakos: Prototipo de moneda complementaria virtual de Girona, desarrollada para apoyar las PYMEs locales.

EurakosNext: Criptomoneda propuesta por la presente tesis doctoral basada en la moneda virtual Eurakos.

Gamificación: Es un término definido por el anglosajón Sebastian Deterding para denotar el uso de las mecánicas de juego en entornos ajenos al juego.

Hash[5]: Consiste en una salida alfanumérica de longitud normalmente fija, creada a partir de una entrada (ya sea un texto, una contraseña o un archivo, por ejemplo) que representa un resumen de toda la información que se le ha dado (es decir, a partir de los datos de la entrada se crea una cadena que sólo puede volverse a crear con esos mismos datos). Estas funciones no tienen el mismo propósito que la criptografía, sino varios cometidos, entre ellos el de asegurar que no se ha modificado un archivo en una transmisión, hacer ilegible una contraseña o firmar digitalmente un documento.

IoT: Término referido a la interconexión digital de objetos cotidianos con Internet. Sigla del *Internet of Things* o Internet de las Cosas.

IOTA[6]: Es una tecnología de contabilidad distribuida de código abierto, cuyo objetivo es permitir de forma segura el intercambio de información y valor en el Internet de las Cosas. Una de las principales innovaciones de IOTA radica en que, en vez del tradicional *Blockchain*, utiliza una arquitectura propia (*Tangle*) basada en un concepto matemático llamado Grafo Acíclico Dirigido (DAG).

---

5 https://www.genbeta.com/desarrollo/que-son-y-para-que-sirven-los-hash-funciones-de-resumen-y-firmas-digitales

6 https://es.wikipedia.org/wiki/IOTA_(protocolo)

Lenguaje Go[7]: Lenguaje de programación concurrente y compilado inspirado en la sintaxis de C. Ha sido desarrollado por Google y se utiliza para desarrollar aplicaciones de *Ethereum*.

LETS: Sistemas comerciales de intercambio local (*Local Exchange And Trading Systems*).

Modelo de negocios CANVAS[8]: Fue diseñado por Osterwalder & Pigneur en 2010. Permite describir visualmente la propuesta de valor de una empresa, ya sea para un servicio o producto. El esquema se divide en tres grandes sectores: el segmento emocional, el analítico y el de recursos económicos. El primero hace foco en los clientes, sus segmentos, la definición de la propuesta de valor y cómo se entablan las relaciones con los consumidores. El sector analítico hace hincapié en la infraestructura e incluye los recursos, los procesos y las alianzas estratégicas. Por último, el segmento financiero representa los costos e ingresos.

P2P (*Peer to Peer*)[9]: Red de pares o iguales. Consiste en una red de equipos de cómputo en la que todos o algunas propiedades funcionan sin clientes ni servidores fijos, es decir, son serie de nodos con un comportamiento igual entre sí. Este tipo de funcionamiento permite el intercambio directo de información, en cualquier formato, entre todos los equipos interconectados.

PIB: Producto Interno Bruto.

Principio KISS[10]: Del inglés *Keep It Simple, Stupid*! («¡Mantenlo sencillo, tonto!»), es un acrónimo usado como principio de diseño. Establece que la mayoría de sistemas funcionan mejor si se mantienen simples en lugar de hacerlos complejos; por ello, la simplicidad debe ser mantenida como un objetivo clave del diseño, y cualquier complejidad innecesaria debe ser evitada.

---

[7] https://goethereumbook.org/en/

[8] https://www.7xfundacionitau.org/nota.php?idn=25&cat=1&gclid=EAIaIQobChMIwKW9643V3wIVw0SGCh0jmQPREAAYASAAEgL3wvD_BwE

[9] https://diccionarioactual.com/p2p/

[10] https://es.wikipedia.org/wiki/Principio_KISS

Falla bizantina[11]: Sucede cuando un componente, como un servidor, puede aparecer de manera incoherente, fallando y funcionando para sistemas de detección de fallas, presentando diferentes síntomas a diferentes observadores. Es difícil para los otros componentes declarar qué falló y cerrarlo fuera de la red, ya que primero necesitan llegar a un consenso sobre qué componente falla. El término se deriva del problema de los generales bizantinos (Lamport, Shostak & Pease, 1982) donde los actores deben acordar una estrategia concertada para evitar una falla catastrófica del sistema, pero algunos de los actores no son confiables.

Proof of Work (PoW): O prueba de trabajo, es un protocolo desarrollado por Dwork & Naor (1993), para impedir los ataques de denegación de servicio y otros abusos como el *spam* en una red, al requerir algo de trabajo del solicitante del servicio, lo que generalmente significaba tiempo de procesamiento de computador. Bitcoin utiliza este sistema como protocolo de consenso para validar transacciones realizadas con esta criptomoneda. Para que un bloque sea aceptado por los participantes de la red, los validadores de transacciones deben completar una prueba de trabajo que cubra todos los datos del bloque y generen un código *hash*. La dificultad de este trabajo se ajusta para limitar la velocidad a la que la red puede generar nuevos bloques, a uno cada 10 minutos. Debido a la muy baja probabilidad de generación exitosa, esto hace que sea impredecible qué equipo de cómputo de la red podrá generar el siguiente bloque[12].

Proof of Authority (PoA): O prueba de autoridad, fue desarrollado principalmente como una solución al problema de los ataques de *spam* en la red de prueba *Ropsten* de *Ethereum*. Es un protocolo *Proof of Stake* optimizado que aprovecha la identidad como herramienta para validar. Esta identidad es dada por un grupo de validadores denominados autoridades (*Authorities*) que han sido aprobados previamente para validar transacciones y bloques dentro de la red respectiva. Generalmente se supone que el grupo de validadores es bastante pequeño (~ 25 o menos) para garantizar la eficiencia y la seguridad de la red que se puede administrar.

[11] https://es.wikipedia.org/wiki/Tolerancia_a_faltas_bizantinas#cite_note-BGP_Paper-1
[1212] https://en.Bitcoin.it/wiki/Proof_of_work

Proof of Stake (PoS): Permite la incorporación de validadores al proceso, dependiendo de la cantidad de ahorros o moneda que tengan en sus billeteras, es decir, que si se mantiene una cierta cantidad de moneda en la billetera durante un mínimo de tiempo, se puede ser validador.

PYME: Pequeña y mediana empresa

Satoshi[13]: Mínima expresión del Bitcoin (0,00000001), nombrada así en honor al seudónimo de su creador. Por ejemplo, cuando se habla de pagar 100 *satoshis* por ver una publicidad online, significa que se recibirán 0,00000100 Bitcoins.

Scrypt: Algoritmo de encriptamiento utilizado por algunas criptomonedas para generar códigos *hash*.

Solidity[14]: Es un lenguaje de programación de alto nivel cuya síntesis es similar a otro de los lenguajes de programación más usados hoy en día como es el Javascript. Está diseñado y compilado en código de bytes (bytecode) para crear y desarrollar *Contratos Inteligentes* que se ejecuten en la Máquina Virtual Ethereum (EVM de sus siglas en inglés).

RES: Moneda complementaria de Bélgica que opera en formato digital.

Timestamp[15]: O marca de tiempo, es una secuencia de caracteres o información codificada que identifica cuándo se produjo un evento determinado. Suele dar la fecha y la hora del día, a veces con una fracción de segundo pequeña.

TIC: Tecnologías de Información y Comunicación

VCC: Moneda complementaria virtual (*Virtual Complementary Currency*)

[13] http://aulaBitcoin.com/basicos/que-es-un-satoshi/
[14] https://miethereum.com/smart-contracts/solidity/
[15] https://en.wikipedia.org/wiki/Timestamp

LISTA DE FIGURAS

LISTA DE TABLAS

CONTENIDO

Lista de Anexos

RESUMEN

Con la popularización de los juegos en línea y las redes sociales, las monedas virtuales han adquirido un auge creciente como solución de pago alternativo. Esto se debe a la facilidad para adaptarse a las necesidades del intercambio de bienes o servicios virtuales que ofrecen transacciones de valor más rápidas, más seguras y de bajo costo. Sin embargo, todavía existen barreras con respecto a los factores humanos (emociones y confianza), la gestión y la infraestructura, que coexisten en proveedores, consumidores y administradores de estas monedas, que se deben reducir para hacer que su adopción sea fácil y natural.

Esta investigación propone una solución para la reducción de algunas de estas barreras, mediante el diseño del protocolo de consenso denominado Prueba de Reputación *(Proof of Reputation - PoR*) que se basa en la tecnología *Blockchain* de *Ethereum*, para confirmar transacciones realizadas con criptomonedas.

El PoR consiste en un híbrido entre los conceptos de *Proof of Work – PoW* y *Proof of Authority –PoA*. Éste exige como requisito básico para participar como validador, el mantener una *Reputación*, la cual se obtiene validando alternativamente con *PoW y PoA* según parámetros pre-establecidos. Esta combinación, permite mitigar múltiples tipos de ataques y simplificar la validación a largo plazo. En ese sentido, su comportamiento consiste en exigir un cierto esfuerzo del validador para confirmar que ejecuta la validación con una máquina propia (PoW) y también que tiene una reputación adquirida honestamente que le permite continuar validando con menos esfuerzo pero con la misma seguridad (PoA).

La reputación establecida, además de ser un concepto integrador para facilitar la implementación de un sistema hibrido, permite darle una definición más significativa al trabajo que realizan los validadores al confirmar transacciones. Un validador se considera altamente confiable si ha alcanzado una reputación alta durante los procesos de validación.

Es importante resaltar, que este sistema podría involucrarse dentro de la estructura de cualquier criptomoneda para la confirmación de transacciones. Precisamente su funcionamiento fue verificado utilizando la criptomoneda EurakosNext que fue diseñada también como aporte de

esta tesis doctoral, tomando como base la moneda virtual Eurakos implementada en Girona (España).

Palabras clave: criptomonedas, barreras de adopción de monedas complementarias y virtuales, EurakosNext, Prueba de Reputación (PoR), Prueba de Trabajo (PoW), Prueba de Autoridad (PoA).

ABSTRACT

With the popularization of online games and social networks, virtual currencies have become increasingly popular as an alternative payment solution. This is due to, the facility to adapt to the needs of the exchange of goods or virtual services that offer faster, safer and lower cost value transactions. However, there are still barriers with respect to human factors (emotions and trust), management and infrastructure, which coexist in suppliers, consumers and administrators of these currencies, which must be reduced to make their adoption easy and natural.

This research proposes a solution for the reduction of part of the aforementioned barriers, through the design of the consensus protocol called *Proof of Reputation* (PoR) that is based on the *Blockchain* technology of *Ethereum*, to confirm transactions made with cryptocurrencies.

The PoR consists of a hybrid between *Proof of Work* - PoW and *Proof of Authority* -PoA, which demands as a basic requirement to participate as a validator, maintaining a *Reputation*, which is obtained by validating alternatively with PoW and PoA according to established parameters. This combination allows to mitigate multiple types of attacks and simplify long-term validation. In that sense, its behavior consists in demanding a certain effort from the validator to confirm that it executes the validation with its own machine (PoW) and also that it has an honestly acquired reputation that allows it to continue validating with less effort but with the same security (PoA).

The established *Reputation*, in addition to being an integrating concept to facilitate the implementation of a hybrid system, allows to give a more meaningful definition to the work performed by validators when confirming transactions. A validator is considered highly reliable if it has achieved a high reputation during validation processes.

It is important to highlight that this system could be involved within the structure of any cryptocurrency for the confirmation of transactions. Precisely its operation was verified using the EurakosNext cryptocurrency that was also designed as a contribution of this thesis, based on the Eurakos virtual currency implemented in Girona (Spain).

Keywords: cryptocurrencies, barriers to adoption of complementary and virtual currencies, EurakosNext, Proof of Reputation (PoR), Proof of Work (PoW), Proof of Authority (PoA).

RESUM

Amb la popularització dels jocs en línia i les xarxes socials, les monedes virtuals han adquirit un auge creixent com a solució de pagament alternatiu. Això a causa de, la facilitat per adaptar-se a les necessitats de l'intercanvi de béns o serveis virtuals que ofereixen transaccions de valor més ràpides, més segures i de baix cost. No obstant això, encara hi ha barreres pel que fa als factors humans (emocions i confiança), la gestió i la infraestructura, que coexisteixen en proveïdors, consumidors i administradors d'aquestes monedes, que s'han de reduir per fer que la seva adopció sigui fàcil i natural.

Aquesta investigació proposa una solució per a la reducció de part de les barreres abans esmentades, mitjançant el disseny del protocol de consens denominat Prova de Reputació (Proof of Reputation - PoR) que es basa en la tecnologia Blockchain de Ethereum, per confirmar transaccions realitzades amb moneda digital.

El PoR consisteix en un híbrid entre Proof of Work - PoW i Proof of Authority - PoA, que exigeix com a requisit bàsic per a participar com validador, el mantenir una *Reputació*, la qual s'obté validant alternativament amb PoW i PoA segons paràmetres establerts. Aquesta combinació, permet mitigar múltiples tipus d'atacs i simplificar la validació a llarg termini. En aquest sentit, el seu comportament consisteix a exigir un cert esforç del validador per confirmar que executa la validació amb una màquina pròpia (PoW) i també que té una reputació adquirida honestament que li permet continuar validant amb menys esforç però amb la mateixa seguretat (PoA).

La reputació establerta, a més de ser un concepte integrador per facilitar la implementació d'un sistema híbrid, permet donar-li una definició més significativa a la feina que realitzen els validadors en confirmar transaccions. Un validador es considera altament fiable si ha assolit una reputació alta durant els processos de validació.

És important ressaltar, que aquest sistema podria involucrar dins de l'estructura de qualsevol moneda digital per a la confirmació de transaccions. Precisament el seu funcionament va ser verificat utilitzant la moneda digital EurakosNext que va ser dissenyada també com a aportació d'aquesta tesi doctoral, prenent com a base la moneda virtual Eurakos implementada a Girona (Espanya).

Paraules clau: criptocrècies, barreres a l'adopció de monedes complementàries i virtuals, EurakosNext, Prova de reputació (PoR), Prova de treball (PoW), Prova d'autoritat (PoA).

1. INTRODUCCIÓN

1.1 Generalidades

Frente a los sistemas monetarios convencionales del capitalismo[16], que concentran desproporcionadamente la riqueza y fomentan la competencia sistemática, la acumulación acelerada y las injusticias sociales, se posicionan hoy en día los sistemas monetarios financieros sociales. Estos estimulan procesos sociales y solidarios pues consideran que el dinero en sí no tiene valor ni sentido si no va asociado a la creación de bienestar. En consecuencia, permiten articular mecanismos para generar sólo pequeñas acumulaciones de capital que propicien poner en marcha proyectos de autoempleo y microempresas en el ámbito local, valorizando los recursos de las comunidades.

Las monedas sociales, complementarias, comunitarias, locales o criptográficas se consideran actualmente, nuevos medios de desarrollo socio-económico que generan sistemas de intercambio local. Estos tienen el fin de impulsar las relaciones entre los miembros de la comunidad y generar participación ciudadana para satisfacer necesidades que la moneda oficial no ofrece; también marcan distancia del monopolio que ejercen los bancos con el manejo del dinero de los ciudadanos.

Sin embargo, la adopción de estas monedas no es fácil aún, debido a la existencia de una serie de barreras que deben superarse para que el proceso sea natural, seguro, sostenible y esté al alcance de todos. Estas barreras tienen que ver con factores humanos o emocionales (por ejemplo, falta de competencias ciudadanas para aprender a vivir juntos, respeto por el otro, trabajo colaborativo, envidia, egoísmo, desconfianza, etc.), procesos de gestión de la moneda y la infraestructura y entorno donde éstas operan.

[16] La moneda convencional o de curso legal se conoce como moneda por decreto (sin valor intrínseco), moneda fiduciaria o moneda fiat.

1.2 Planteamiento

La presente investigación, enfoca sus esfuerzos en la generación de soluciones que reduzcan las barreras antes mencionadas en términos de: los factores emocionales - generando confianza entre los miembros de la comunidad asociada a la moneda; la gestión de la moneda – quitando a los usuarios dicha responsabilidad y delegándola a la tecnología; y la infraestructura y entorno donde operan las monedas - seleccionando de acuerdo al contexto, el mejor formato de la moneda.

En ese sentido, la pregunta de investigación propuesta es la siguiente:

¿Lo*s nuevos diseños de monedas complementarias, teniendo en cuenta el desarrollo de las redes sociales y las tecnologías blockchain y de contratos inteligentes, cómo pueden mejorar los negocios?*

Está claro entonces, que la solución al problema involucra la explotación del uso de la tecnología en contextos socio-económicos donde la apropiación de la misma en la cotidianidad, está consolidada.

La innovación social digital ha surgido como un nuevo tema en torno al cambio de siglo, provocada por una serie de avances en las tecnologías de Internet. El cambio de una sociedad jerárquica a una sociedad en red, ha creado una cultura de dependencia de los demás, canalizada por Internet y nuevas aplicaciones de Tecnologías de Información y Comunicación.

Baeck y Bria (2014) afirman que las innovaciones sociales pueden ser el resultado de cuatro tendencias tecnológicas principales que de una u otra manera favorecen a las personas y fortalecen sus relaciones de colaboración:

a. Hardware abierto, que hace que el hardware digital esté disponible para adaptarse, manipularse y transformarse en herramientas para el cambio social;
b. *Conocimiento abierto*, que se refiere a grandes grupos de ciudadanos reunidos a través de plataformas en línea para analizar colectivamente datos, desarrollar y analizar nuevos tipos de conocimiento o para presentar proyectos sociales de financiación colectiva (*crowdfunding*);

c. Datos abiertos, que se refiere a formas innovadoras de abrir, capturar, usar, analizar e interpretar datos abiertos; y,
d. Redes abiertas, que describe la manera como los ciudadanos están desarrollando nuevas redes e infraestructuras, por ejemplo, redes de sensores, donde conectan sus dispositivos, como teléfonos y módems de Internet, para compartir recursos colectivamente y resolver problemas.

La tecnología *Blockchain* específicamente, está revolucionando la forma de almacenar datos. Es una base de datos distribuida pública o privada con datos sellados e inmutables de cada transacción. Es el corazón de la criptomoneda Bitcoin que, además de respaldar las criptomonedas que surgen cada día, permite el desarrollo de aplicaciones en contextos tales como: innovación en servicios financieros (Fintech), Internet de las cosas, democracia electrónica, identificación digital, atención médica, seguros, e-learning, etc.

Los bancos, igualmente, ya están realizando investigaciones y pruebas de concepto con esta tecnología porque agiliza el proceso transaccional de forma segura, y evita intermediarios. Esto conlleva a comisiones más bajas y servicios más atractivos a los consumidores.

La solución del problema de investigación de esta tesis, se lleva a cabo mediante el método científico con base en un diseño experimental cuantitativo y con razonamiento deductivo, soportado en las siguientes concepciones:

1. En los últimos treinta años, se han desarrollado más de 4000 sistemas monetarios complementarios en más de 50 países alrededor del mundo con una variedad de modelos y esquemas de aplicación. Desafortunadamente, debido a las numerosas barreras y limitaciones identificadas y estudiadas por el proyecto VirCoin2SME[17], un gran porcentaje de estos sistemas ha fracasado en su intento de crear prácticas alternativas que apunten a apoyar las economías locales hacia ecosistemas sostenibles, mientras mejoran las interacciones sociales entre los diferentes actores involucrados.

[17] VIRCOIN2SME: Social, complementary or community virtual currencies transfer of knowledge to SME: a new era for competitiveness and entrepreneurship – Proyecto Europeo de iniciativa H2020 Ref. 645767

2. En la era de la disrupción del *Blockchain* y de los nuevos avances tecnológicos en redes sociales y dinero social, las monedas sociales virtuales-complementarias esperan alcanzar los más altos niveles no sólo de desarrollo e innovación, sino también de usabilidad, seguridad y confianza. Sin embargo, antes de realizar este exitoso salto, es necesario comprender y enfrentar las barreras para su adopción y replicación. Además, se deben explorar limitaciones con el fin de superarlas y mejorar el desempeño, y con ello, satisfacer los objetivos, características y mecanismos para crear a largo plazo modelos de negocio que impulsen el desarrollo socio-económico local, regional o global.

3. La existencia de numerosas tecnologías y métodos de validación de criptomonedas, dificulta la equivalencia entre unas y otras, y por tanto imposibilita su intercambio. Es importante el aprovechamiento de esa diversidad, para generar soluciones lo suficientemente robustas en términos de seguridad y rapidez, que permitan cierto nivel de personalización y nuevos métodos de validación implementables sobre una tecnología común.

4. Las tecnologías basadas en el blockchain, específicamente *Ethereum*, permiten diseñar nuevas aplicaciones mediante su propio lenguaje de programación y probar nuevos métodos de validación sobre éstas de manera abierta y libre.

5. En términos de protocolos de consenso por ejemplo, el problema que tiene el tradicional método *Proof of Work* del Bitcoin, es que a medida que haya más y más validaciones, y estas se requieran a mayor velocidad, se aumenta la dificultad y hace que no cualquier usuario pueda participar, dando origen a que después de cierto tiempo lo único que se logre es que el poder que tienen los bancos sobre las monedas de curso legal, se traslade a los validadores de criptomonedas con suficiente potencia de validación.

6. La implementación de otros sistemas de validación aislados, puede que solucione determinados problemas, pero al final afecta la velocidad de uso de la moneda o facilita la generación de otros ataques, por tanto, el autor de esta tesis plantea un método de validación híbrido para aprovechar las bondades de los protocolos participantes mitigando entre sí, los posibles ataques que puedan surgir entre ellos, y realizar eficientemente la tarea para la cual fue concebido, que es motivar el uso y apropiación de las criptomonedas en los negocios, reduciendo las barreras para su adopción (por ejemplo: confianza, baja liquidez, poca oferta, costo incurrido en la validación, esfuerzo para su mantenimiento y viralización y, gestión centralizada, entre otras) e impulsar con ello el desarrollo socio-económico en contextos locales, regionales o globales.

7. El método del estudio de caso utilizado para conocer el funcionamiento de la moneda digital RES y la moneda virtual Eurakos (durante la participación del autor de esta tesis en el proyecto Vircoin2SME), permitió adquirir conocimiento para orientar el diseño de la criptomoneda EurakosNext que aprovecha los beneficios de la tecnología *blockchain* y ofrece la oportunidad de innovar en los procesos de validación de transacciones, con el fin de motivar su adopción en contextos económicos locales de micronegociaciones, de manera segura, fácil y sostenible. Su uso permitiría reducir las barreras relacionadas con el factor humano (emociones y confianza), facilitaría la gestión de la moneda por parte de las PYMES y los consumidores, y ampliaría el contexto en donde se utilizan, especialmente cuando el acceso a la tecnología está garantizado a la población.

8. El círculo de la ingeniería de la usabilidad (analizar contexto, conceptualizar, diseñar la solución, probarla y ajustarla para liberarla como final o para retroalimentar el ciclo), que se enfoca en orientar el procedimiento para generar soluciones software con: facilidad de uso (múltiples formas de intercambiar información entre el usuario y el sistema) y, facilidad de aprendizaje para nuevos usuarios que garantizan interacción efectiva, máximas prestaciones y satisfacción del usuario, incluyendo el soporte al mismo para el alcance de las metas. (Rosson y Carroll, 2001)

1.3 Objetivos

Los objetivos planteados para dar solución al problema de investigación son los siguientes:

Objetivo general

Diseñar un protocolo de consenso que inspire seguridad y confianza en los usuarios de criptomonedas para el fomento de su uso y apropiación, y que impulse el desarrollo socio-económico en contextos locales, nacionales e internacionales.

Objetivos específicos

1. Definir el contexto económico para la creación de iniciativas sostenibles basadas en el uso de monedas sociales-complementarias.
2. Identificar las barreras de adopción de monedas sociales-complementarias y posibles soluciones para su reducción desde el punto de vista de los usuarios.
3. Analizar desarrollos tecnológicos existentes en torno al concepto, estructura y uso de criptomonedas, para la identificación de variables que permitan soluciones que amplíen su rango de uso de manera directa, natural y segura en comunidades locales, nacionales e internacionales.
4. Diseñar la criptomoneda EurakosNext y el protocolo de consenso *Proof of Reputation* (PoR) para el fomento, uso y adopción de criptomonedas en los negocios, que minimicen algunas de las barreras identificadas en 2) en contextos económicos locales, regionales y nacionales.

1.4 Organización del texto

Este documento está organizado de la siguiente manera:

En la sección 2 se desarrolla el marco teórico y estado del arte de esta tesis, que involucra temas relacionados con la economía, las monedas sociales y complementarias, las barreras identificadas para la adopción de estas monedas de manera natural, segura y sostenible entre otras y las criptomonedas que ofrecen la base científico-tecnológica que permite el logro de los objetivos de la investigación.

La sección 3 introduce el marco metodológico de la investigación.

En la sección 4 se presenta la manera como se diseñó y abordó la investigación, resaltando el diseño de la criptomoneda *EurakosNext* y el protocolo de consenso *Proof of Reputation* (PoR) como pruebas de concepto para la solución del problema de investigación planteado.

La sección 5 introduce el proceso llevado a cabo para la recolección de datos.

La sección 6 presenta el análisis e interpretación de los resultados de la investigación

Y, finalmente la sección 7 cierra el documento con las conclusiones y el trabajo futuro.

2. MARCO TEÓRICO Y ESTADO DEL ARTE

2.1 Situación y entorno

La Economía se considera una ciencia que estudia los recursos, la creación de riqueza y la producción, distribución y consumo de bienes y servicios para satisfacer las necesidades humanas (Mochón & Beker, 2008). En este sentido, las actividades humanas económicamente relevantes son la producción, la distribución y el consumo (Resico, 2010).

El consumo implica la asignación de los medios productivos y recursos disponibles para la obtención de bienes y servicios que mejor satisfagan las necesidades y deseos del comprador. La producción es la actividad económica de la que derivan los bienes y servicios que luego de su distribución son aplicados a la satisfacción del consumo. La producción de bienes abarca las actividades extractivas o primarias, las de procesamiento o secundarias y las de distribución o terciarias (Resico 2010).

La distribución es la actividad económica que parte de lo producido y determina la proporción de los bienes y servicios para cada uno. Este proceso responde a la capacidad del mercado de premiar a los más productivos con mayores ingresos mientras castiga con menos ingresos a los menos productivos. Este proceso espontáneo no es perfecto, debido a externalidades tales como la herencia, la educación, la propiedad, el conocimiento, etc. Por lo cual, puede ser complementado con una reasignación social. Esta reasignación tiene diferentes etapas como la solidaridad individual, la solidaridad social, o la respuesta social del estado. En conclusión, la distribución no solo depende de consideraciones económicas sino también de solidaridad y ética social y ambiental.

La economía considera fundamental tener en cuenta los factores de producción, típicamente los recursos naturales, el capital humano y el capital físico (Mochón & Beker, 2008 y Resico, 2010) aunque cada vez más se considera necesario tener en cuenta como factor de producción el conocimiento (Lietaer, 2001; Carrillo, De la Rosa, & Canals, 2007).

Los Recursos Naturales incluyen la tierra para producción agrícola o instalación de plantas industriales; los recursos extractivos y los elementos que hacen posible la extracción, se encuentran en la naturaleza y que la presión ecologista lleva a que se contabilicen más abiertamente (Lietaer 2001); y el cuidado de recursos como el agua, los recursos forestales y el medio ambiente.

El capital humano o recurso humano, incluye las capacidades humanas requeridas por el proceso productivo tales como el trabajo físico o intelectual, el conocimiento técnico, la iniciativa, la innovación y la capacidad de organización del proceso productivo. Así mismo, incluye de modo indirecto, el capital social o capital humano relacional como la capacidad de establecer y mejorar las instituciones o la capacidad de cooperación y autogobierno. Sin embargo, el foco ha sido considerado por el número de personas con capacidad de trabajar así como sus habilidades. Por ello, y sobre todo por vivir en una sociedad de conocimiento, se destilan propuestas de separar específicamente estas condiciones especiales en un nuevo factor de producción, el conocimiento y la economía del conocimiento (Lietaer, 2001).

Otro factor considerado muy importante y base de trabajo de la economía es el capital físico o capital que considera todos aquellos bienes usados para producir otros bienes y que son fabricados por el hombre (máquinas, computadores, instalaciones, etc.). El dinero no se considera bien de capital.

La estructura económica puede basarse en decisiones individuales o de unidades económicas (familias, empresas, etc.) conocido como sistema de mercado, o puede basarse en decisiones jerárquicas burocráticas o sistema de planificación centralizada. En la economía de mercado los individuos o unidades económicas realizan las actividades con autonomía, debido a la existencia de la propiedad privada, y la coordinación de estas actividades se hace mediante el sistema de precios.

A su vez, en este aspecto, el dinero se ha convertido en el medio por el cual la economía se desarrolla, debido a que facilita la realización de las transacciones, tiene un valor intrínseco, sirve como unidad de cuenta y permite documentar deudas y fijar precios. Dicen Mochón y Beker (2008): "… el dinero es, todo medio de pago, generalmente aceptado, que puede intercambiarse por bienes y servicios. El precio de un bien es el número de unidades de dinero que se intercambian por una unidad del bien" (p.13) y "un sistema económico, por su parte, se define como el conjunto de relaciones básicas, técnicas e institucionales que caracterizan la organización económica de una sociedad y condicionan el sentido general de sus decisiones fundamentales, así como los cauces predominantes de su actividad" (p.14).

La toma de decisiones económicas (qué, cómo y para quién producir) se realiza mediante una mezcla de mercados y gobierno. El mercado es el mecanismo por el cual compradores y vendedores intercambian bienes y servicios. El mercado es por principio, liberalizado. El gobierno, al contrario, trata de estructurar, organizar y controlar el flujo de capitales y la asignación estructurada de bienes y servicios.

Según Kennedy y Kennedy (1998), el mercado convencional presenta problemas en la realidad debido a: 1. Creencia de que existe un solo tipo de crecimiento (exponencial). 2. Creencia de que sólo pagan intereses quienes pagan directamente un préstamo en dinero. 3. Creencia de que bajo el actual sistema monetario todos son afectados en igual medida por las tasas de interés. 4. Creencia de que la inflación es parte integral de la economía.

En los libros de economía no se hace referencia normal a los bancos, aunque sin ellos es imposible entender funciones colaterales al mercado como el ahorro, la creación de dinero, las tasas de interés, y, adicionalmente las distorsiones del mercado. Los bancos por su parte, se han convertido en elementos base de la gestión económica. Ellos han jugado, por décadas, un papel importante en la economía en vista de sus servicios de almacenamiento y protección del dinero, y han disfrutado de un poder considerable en vista de su situación monopólica. La inexistencia de alternativas les brinda un poder considerable ya que junto a políticas de salarios precarias han ofrecido la oportunidad de proponer nuevos servicios financieros que han terminado siendo desproporcionados y piramidales.

Los fallos estructurales del sistema monetario son enunciados por Lietaer, Arnsperger, Goerner y Brunnhuber (2012) así: 1. La tendencia procíclica de creación y flujo monetarios, 2. El cortoplacismo promovido por el descuento actual de beneficios futuros, 3. La presión al crecimiento permanente, 4. La concentración de la riqueza y 5. La devaluación del capital social.

Una moneda convencional tiene valor por el reconocimiento que se le dé en los contextos económicos. Si alguien en Argentina o Suecia pretendiera pagar un hotel o comprar una hamburguesa con Lilangenis (moneda de Suazilandia), seguramente no tendría hospedaje o pasaría hambre. Sin embargo, si la persona llevara dólares (moneda de EEUU) o Euros (moneda europea), seguramente sería hospedado o servido sin ambages. Esto se debe al reconocimiento casi universal de la economía estadounidense y de la europea, por lo cual se da crédito a la autoridad de los correspondientes bancos centrales, sin cuestionar políticas o metodologías particulares. El manejo económico de estos países y posiblemente su respaldo por la Reserva Federal o el Banco Central Europeo, respectivamente, han validado la moneda y las transacciones que en ella se dan.

La economía, por su parte, ha crecido en forma desmedida hasta el punto que cada 10 años se presenta un colapso. Este es condicionado por el crecimiento no respaldado en activos, la demora y el costo en/de las transacciones, la multiplicidad de divisas, y la orientación cada vez mayor al crecimiento económico.

Por esta razón, han aparecido diversos tipos de monedas, denominadas sociales, complementarias y virtuales, con el fín de superar los problemas surgidos en la banca y economía tradicionales, y en forma alterna a ellos como administradores del dinero.

Hoy en día existen más de 1500 criptomonedas o monedas digitales. Entre las más conocidas están: Bitcoin, Ethereum, Litecoin, Blackcoin, RES, Sardex, etc. Sin embargo, existen diferentes modelos de comportamiento de las mismas y específicamente entre las criptomonedas se diferencian unas de otras por la forma en que se validan las transacciones, aunque prevalece el *Proof of Work* como algoritmo original de validación, que es un proceso que cada día es más difícil de ejecutar y se ha convertido en una tarea exclusiva de los que puedan pagarse un buen hardware como requisito para poder validar, y al final no termina siendo tan accesible como debiera ser. Por ello, esta tesis pretende aportar a las estrategias de validación y a las posibilidades de transferencia de valor de las criptomonedas, para con ello contribuir a motivar su uso y a beneficiar económicamente a las comunidades de una manera más igualitaria.

2.2 Monedas sociales y complementarias

2.2.1 Definición

Como una alternativa a los sistemas financieros y monetarios convencionales que se caracterizan por su tendencia a la especulación y el acaparamiento, han surgido los sistemas sociales, que se centran en las finanzas solidarias, y están vinculados al microcrédito y las micro-finanzas.

La articulación de tales sistemas ha dado lugar a la economía social, que libera la moneda de curso legal del desarrollo local, e intenta encontrar nuevas soluciones a problemas sociales, económicos o ambientales que han sido ignorados o abordados de manera inadecuada. El objetivo de esta economía no es en principio, acumular ganancias, sino generar bienes y servicios a través de estructuras productivas justas, donde la igualdad laboral, la igualdad de género y el respeto por el planeta tienen prioridad (Blondeau et al., 2004).

Si las soluciones están diseñadas para alcanzar objetivos sin ánimo de lucro, según Scott y Green (2013), una economía social tiene un rol único en la creación de una sociedad fuerte, sostenible, inclusiva, próspera y solidaria.

Para operar en una economía social se requiere un tipo de moneda, que puede ser local, comunitaria, complementaria o virtual. HRZONE (2015), define la moneda social como un conjunto de recursos y habilidades creadas y puestas a disposición a través de comunidades y redes (en línea o fuera de línea); Vivaldi Partners (2013) afirman que las monedas sociales equivalen al grado con el que los clientes comparten una marca (como resultado simple de la interacción entre personas) o una información de marca o negocio con otros. Según Gisbert (2016b), la moneda social es un medio de intercambio que tiene paridad con el dinero de curso legal y no es depósito de valor. Al final, la participación social para conocer o usar todo tipo de información (por ejemplo, productos, servicios, hábitos, historia, etc.) es la que otorga valor a esta moneda.

En general, el objetivo de una moneda alternativa es proporcionar servicios y características específicas que la moneda de curso legal no puede ofrecer, como por ejemplo, garantizar que las personas satisfagan sus necesidades básicas, promover la ciudadanía activa, fortalecer comportamientos sostenibles y ambientales (LANZAROTE, 2013), crear cultura cívica, promover el comercio local, crear una sociedad de aprendizaje, etc.

2.2.2 Diseño

Para Lietaer y Belgin (2012), el diseño de una moneda alternativa varía según la forma en que responde a los diferentes orígenes, principios y requisitos. Sin embargo, Blanc (2011) afirma que está más organizado y enfoca mejor el objetivo, si ese diseño responde a la filosofía y el propósito general de un proyecto caracterizado por sus diseñadores. En ese sentido, propone tres tipos de proyectos que constituirían la raíz de cualquier tipo de sistema monetario: un *proyecto territorial*, enfocado principalmente en un espacio geopolítico; un *proyecto comunitario*, enfocado principalmente en una comunidad preexistente o adhoc o un *proyecto económico*, enfocado en actividades de producción y mercados de intercambio.

Las monedas complementarias se utilizan en entornos económicos locales actuales (pero limitados), operan bajo el principio de combinar su uso con la moneda de curso legal, y tienen el sentido, por suparte, de estimular la actividad económica local mediante la reestructuración de los gastos de consumo diario. Para garantizar el éxito de las experiencias con esta moneda, es necesario vincular a las pequeñas y medianas empresas (PYMEs) y a los gobiernos locales (preferiblemente sobre todo por los pagos de servicios públicos).

Vivaldi Partners (2012) proponen la existencia de seis tipos de comportamientos sociales básicos de los usuarios para ser tenidos en cuenta en la implementación de negocios basados en monedas sociales y complementarias. Estos son:

1. Utilidad, que consiste en ganar valor mediante la interacción entre negocios y usuarios
2. Información, para recibir y compartir con otras personas información valiosa sobre los negocios establecidos en la red.
3. Conversación, para hablar a otros sobre una marca o negocio
4. Promoción, para promover o defender una marca o negocio.
5. Afiliación, para conectarse y ser miembro de una comunidad de personas vinculada a una marca o empresa.

6. Fidelidad, que es una tendencia actitudinal y de comportamiento para favorecer a una marca sobre todas las demás y recomendarla a otros clientes. La fidelidad de los usuarios en los sistemas de moneda social ha sido uno de los factores que más ha influido en su éxito, especialmente porque es una cuestión que depende del diseño de los entornos de interacción considerando los comportamientos de los usuarios mencionados por Vivaldi.

Con base en lo anterior, Berger (2013) concluye que la gente recomienda y comparte con los demás, las cosas que les ha hecho sentir bien (por ejemplo, buen servicio, buena relación calidad/precio, entrega rápida, etc.) y que se deben tener en cuenta los siguientes elementos para ser incorporados en los sistemas de negocios sociales:

1. Identificar el sello más importante o distintivo de la oferta del producto o servicio y ponerlo siempre visible (si está en una interfaz de red social virtual, utilizar un cuadro central al frente del usuario).
2. Usar la mecánica de juegos para alentar a los clientes a participar en ciertas acciones para el logro de los objetivos. Esto permite conocer cómo es el desempeño del cliente con respecto a los demás, lo que fomenta el espíritu de competitividad y lo motiva internamente para ser mejor que otros.
3. Hacer que los clientes se sientan exclusivos. En ese sentido, Cialdini (2006) afirma que si hay menos existencia de un producto o menos disponibilidad de servicio, los clientes lo desearán más. Si algo es raro y poco común, la mayoría de la gente lo querrá, entonces compartirán con otros la marca y es posible que esto atraiga nuevos clientes.

2.2.3 Representación

Una moneda social o complementaria puede tener una representación física a manera de billetes temáticos, libretas de notas contables o bonos, o una representación digital/virtual como pueden ser los puntos acumulados para actividades llevadas a cabo a través de comunidades virtuales (negocios y juegos en línea, por ejemplo). El uso de cualquiera de ellas requiere el cumplimiento de las reglas establecidas por la comunidad que las haya creado.

El Banco Central Europeo definió la moneda virtual, como un tipo de dinero digital que funciona en un entorno no regulado emitido y controlado por sus desarrolladores y utilizado como método de pago entre los miembros de una comunidad virtual específica (Buntinx, 2015). La gestión de dichas monedas puede realizarse de manera centralizada o descentralizada.

La evolución futura de las monedas virtuales complementarias está ciertamente vinculada al progreso tecnológico (con el uso de Internet y dispositivos móviles), a su reconocimiento como un elemento clave de las políticas públicas, y a su uso como una herramienta para soluciones ambientales.

La implementación de estos sistemas no es fácil, ya que existe una serie de factores a articular para que las experiencias sean exitosas. Precisamente, uno de los objetivos del proyecto Vircoin2SME, al que estuvo vinculado el autor de esta tesis, fue identificar las barreras que pudieran surgir para adoptar de forma natural monedas sociales y complementarias en cualquiera de sus representaciones, la idea era fortalecer economías locales (tendiendo más hacia representaciones virtuales) y encontrar posibles soluciones para reducirlas.

2.2.4 Ejemplos

Algunos estudiosos del campo de las monedas sociales y complementarias las categorizan entre otras como aquellas basadas en la confianza mutua o en conocimiento, o respaldadas con dinero de curso legal. A continuación se introduce el concepto y aplicación de las mismas desde el punto de vista que le concierne a esta tesis doctoral.

Sistemas de Intercambio Local LETS (Local Exchange Trading Systems)

Son monedas generadas con base en la confianza mutua, que nacieron en la Columbia Británica, Canadá, en la década de los 80 y se han expandido a otros países especialmente al Reino Unido, Australia, Francia, Alemania y a España, como consecuencia de la crisis económica.

En este caso, no circulan billetes ni monedas sino cada socio crea su cuenta y utiliza una libreta para registrar su saldo (proceso manual). Las compras generan puntos negativos y las ventas puntos positivos. Al final, la suma de todas las cuentas debe quedar siempre en cero, es decir, alguien tiene que quedarse con el saldo negativo (si compra algo) para que otro quede con el saldo positivo (si vende algo).

No se cobra ninguna tasa de interés a los saldos negativos y el deudor puede cancelar su saldo cuando quiera con lo que tiene, bien sea un producto (por ejemplo: arroz o cebollas) o un servicio (por ejemplo: fontanería, carpintería, clases de informática, asistencia de hogar, etc.).

Normalmente estos grupos no son grandes y muy pocos sistemas han conseguido reunir más de 200 socios activos porque muchos socios dejan de ser activos al no encontrar formas eficaces de gastar su saldo positivo, sin embargo, es una herramienta útil, que permite la construcción de cohesiones sociales y es una buena iniciativa para generar empleo y para fomentar la economía local (Hirota, 2012).

En España, la experiencia viva con los sistemas LETS la representa, El Zoquito (la más antigua), que funciona en Jerez de la Frontera (Cádiz) desde 2006. Luego han surgido otras iniciativas dentro de las que es importante mencionar, el Puma (Sevilla), el Eco Alt Congost (Barcelona) y el Eco usado en diferentes cooperativas integrales, de las cuales la más destacada es la Cooperativa Integral Catalana que usa esta moneda social como herramienta estratégica para aumentar la autosuficiencia de sus propios socios. La Turuta en Vilanova i La Geltrú (Barcelona) es una variante de este tipo de moneda que no permite que los socios tengan saldo negativo (sólo la Oficina de Cambio Local cuando se aprueban proyectos locales), es decir, esta restricción hace que se aumente la velocidad de gasto de la moneda y por supuesto se dinamice la economía local (Hirota, 2014).

Monedas basadas en el conocimiento

Las economías basadas en el conocimiento han surgido de la evolución de la sociedad frente a la aparición de la tecnología para el manejo de la información y las comunicaciones, y a la capacidad de innovar y crear valor con base en el conocimiento. Este, por su parte, se actualiza día a día por medio del aprendizaje alcanzado no solo a través de las aulas de clase, sino en los lugares de trabajo, los laboratorios, los centros de investigación, etc.

En esta sección, se presentan varios casos de estudio que se han llevado a cabo a través de la investigación, con el fin de utilizar de una u otra manera, la concepción de monedas complementarias/sociales como medio para:

a. Establecer nuevas formas de orientar los programas educativos en el logro de aprendizajes significativos, que repercutan en el saber-hacer en contexto (desempeño profesional).

b. Construir conocimiento de manera colaborativa y solidaria desde las aulas de clase con participación activa de todos los actores del proceso educativo.

c. Gestionar el conocimiento resultante de procesos de investigación, que incentive al investigador a seguir generando conocimiento y lo motive a mejorar la calidad de sus aportaciones, así el conocimiento se convierte en motor del desarrollo social y del crecimiento económico de una región o país.

Se inicia entonces, dando a conocer la propuesta de Seymour, Everhart y Yoshino (2013), quienes afirman que la estructura de la educación universitaria debe cambiar en la utilización de las "unidades de medida" del aprendizaje. Parten del hecho de que los sistemas tradicionales organizan los currículos con base en el sistema de créditos-horas (considerados moneda complementaria), que es estático y no ofrece suficiente información para la toma de decisiones a los diferentes gestores y actores de los programas educativos y que repercute al final en la generación de un conocimiento poco útil para la innovación y el desarrollo de la sociedad. Entonces, su idea se centra en establecer las competencias, como una nueva moneda complementaria, que soporte las credenciales profesionales y ofrezca beneficios reales a actores y procesos de ecosistemas educativos complejos, cada uno desde su mirada, de manera natural y comprometida.

Las ventajas de utilizar las competencias, como unidad de valor para la generación o transferencia de conocimiento, se reflejan en el contexto educativo así:

a. Las entidades gubernamentales, ofrecen beneficios económicos o de empleo, si las competencias están alineadas con el mejoramiento de la mano de obra.

b. Las instituciones educativas que lideran los procesos de formación por competencias mejoran la satisfacción, retención y grado de atención de los estudiantes.

c. Las entidades de gestión educativa, ofrecen el andamiaje para la acreditación de credenciales, si las competencias se diseñan para cumplir requerimientos claros y lógicos.

d. Los expertos temáticos claramente articulan los logros del aprendizaje con y a través de las disciplinas.

e. El cuerpo de profesores lleva a cabo un entendimiento transparente del logro del aprendizaje de los estudiantes.

f. Los evaluadores del aprendizaje, se benefician de las competencias, si éstas están bien definidas y reflejan los logros de aprendizaje a ser evaluados y medidos tanto para estudiantes individuales, como para programas y grados.

g. Los estudiantes se benefician del entendimiento transparente de las competencias requeridas para su profesión.

h. Los empleadores se benefician del entendimiento transparente de las competencias de los egresados.

Continuando con las posibles aplicaciones de las monedas complementarias en el campo educativo, se presenta a continuación la propuesta de De la Rosa, Batlle, Batlle, Szymanski y Krishnamoorthy (2009), que se enfoca en soluciones para motivar (mediante incentivos con fuertes lazos comunitarios), a estudiantes, profesores y padres de familia, a trabajar colectivamente por el logro de los objetivos de aprendizaje.

Como resultado de la iniciativa antes mencionada, se diseñó la moneda Wits (cosas con sentido), con el fin de incentivar toda acción que tenga sentido en la comunidad a favor del aprendizaje. Es decir, si un estudiante hace algo que tenga sentido para otro estudiante y que le "ayude" a aprender, se recompensa con esta moneda, que tiene varias connotaciones dependiendo del contexto en el que se desarrolle la acción. Por ejemplo, podría representarse en "notas más altas" o en el fortalecimiento de las conexiones con la comunidad.

Si los profesores igualmente, fuera de sus labores docentes propias, ayudan a los estudiantes al desarrollo de sus objetivos de aprendizaje, también pueden ser recompensados con Wits, que en este caso, serán de responsabilidad de las entidades públicas o de los padres de los estudiantes. Las recompensas siempre serán otorgadas por meritocracia, así que la comunidad educativa se autorregulará frente al uso de estos Wits.

Monedas respaldadas con dinero de curso legal

CHIEMGAUER

Es una moneda complementaria (regional) fundada en 2003 en la región de Prien am Chiemsee, Baviera, Alemania y sus alrededores. La iniciativa de crear esta moneda fue de las alumnas de una escuela Waldorf (primaria y secundaria funcionando bajo la pedagogía de Rudolf Steiner) con la supervisión de Christian Gelleri, profesor de economía, cuando ellas estuvieron interesadas en implementar un sistema tras conocer la teoría de monedas complementarias.

Los socios para poder utilizar la moneda, deben cambiar sus euros en Chiemgauers en la oficina de la asociación que quieren apoyar (ONG) o utilizando la tarjeta Regiocard. Luego, los pagos se pueden realizar con esta moneda en los comercios locales que la acepten. Los comercios pueden comprar sus mercancías en Chiemgauer o rembolsarlos en euros si no les importa perder el 5% de comisión, de la cual el 2% se gastará como costo administrativo de la oficina de Chiemgauer y el restante 3% se destinará a beneficiar la asociación vendedora (Hirota, 2012).

De acuerdo con Gisbert (2016b), este sistema favorece tanto a los consumidores como a las empresas locales, definiéndose la participación de los diferentes actores locales de la siguiente manera:

a. ONG: obtener 100 Chiemgauer al precio de € 97 que luego se pueden "revender" a sus socios a € 100, ganando como consecuencia € 3 para gastar en sus propias actividades.
b. Consumidores: compran 100 Chiemgauer a 100 € y los gastan en comercios locales por su valor nominal, donando el 3% de su consumo a las ONG locales.
c. Comercios locales asociados: aceptan 100 Chiemgauers y los aceptan para sus compras en otros comercios locales o los reconvierten en 95 € en las oficinas de Chiemgauer pagando el 5% de comisión; se considera que este 5% es el costo de publicidad y de desarrollo del sistema (impresión de billetes, materiales, etc.), y se destina también a proyectos de ONG locales.

d. Oficina de Chiemgauer: Vende 100 Chiemgauers a 97€ y paga 95€ al reconvertirlos; se gasta la diferencia de 2€ para fines administrativos y el 3% restante para proyectos locales.

Chiemgauer es una moneda oxidable (es decir pierde su valor con el tiempo) y se requiere pegar un sello del 2 % del valor del billete (€ 0,10 para 5 Chiemgauer, por ejemplo) cada tres meses para mantener su validez, lo que hace que los portadores de esta moneda complementaria no la acumulen sino que la gasten cuanto antes para, eventualmente, estimular la economía regional (Hirota, 2012).

WIR

En Suiza, el ideólogo económico Silvio Gesell, autor del libro "El nuevo orden económico" y referido en EUMED (2002), puso en marcha esta moneda que inició como un círculo de compensación bajo el concepto del dinero libre, tras la gran depresión económica. Después se conformó un banco cooperativo y hasta 1948, se pagaba en WIR (que significa "nosotros" en Alemán) en vez de con francos suizos porque la circulación del dinero tradicional no bastaba. Hoy se utiliza para las transacciones entre PYMES suizas (cerca de 75.000), que han podido mejorar sus resultados gracias a los préstamos más baratos que firman con esta moneda complementaria.

El WIR es un sistema de moneda local, cuya contabilidad está ligada a los bancos, pero que es independiente de éstos y equilibra las fluctuaciones económicas, complementando a la economía convencional, principalmente en los sectores de hostelería, construcción, fabricación, venta al por menor y servicios profesionales.

Los usuarios de WIR utilizan esta moneda en paralelo con los francos suizos a la hora de realizar transacciones. Esto es, si hubiera que realizar un pago de 100 francos, éste se haría entregando 30 WIR y 70 francos suizos. Y la proporción cambia conforme al estado del franco suizo. Es decir, más WIR cuando la economía no es tan próspera, menos WIR en situaciones de estabilidad y crecimiento económico (Stodder, 2009).

Los miembros de esta cooperativa obtienen préstamos con tasas de interés más bajas que las que ofrecen los bancos que se manejan en francos suizos porque el banco WIR puede crear WIR por sí mismo, sin necesidad de solicitar francos suizos del Banco Central con la tasa de interés, como hace el resto. Y puede ofrecer, por ejemplo, un préstamo de 2% en WIR y otro de 5% en francos suizos si la tasa oficial es del 3%. La frontera de esta moneda no deja escapar este poder adquisitivo, ya que sólo puede emplearse entre los miembros (Remon, 2011).

Esta moneda ha servido de inspiración a muchas otras que han surgido para su funcionamiento de manera complementaria al Euro (€) con resultados exitosos en sus comunidades, como por ejemplo el RES (que significa "cosa" en Dutch) que inició en Bélgica hace 20 años y ya se utiliza en España (filial en Cataluña RES.cat). A continuación se introduce en detalle su estructura y funcionamiento, porque el autor de esta investigación trabajó de cerca con esta moneda para identificar las barreras de adopción de las monedas sociales-complementarias virtuales y proponer posibles soluciones para su reducción.

RES

La casa matriz de la moneda complementaria RES está en Lovaina – Bélgica. La organización Admin Leuven creó en 1996 esta moneda como una red cooperativa de 5000 comerciantes locales, PYMES, comerciantes individuales y profesionales que utilizan el concepto del negocio RES (ver Figura 1) para apoyar a los comerciantes locales y las PYMES. Opera de manera similar a la moneda WIR de Suiza, es decir, como una organización centralizada pero con gran influencia de los participantes.

Es una moneda digital que funciona de forma complementaria al Euro. Su principal objetivo es generar ingresos adicionales (3 a 5%) a través del intercambio comercial entre sus miembros, incluyendo préstamos sin intereses para financiar inversiones (hasta de 2000 RES para comenzar). La cooperativa cuenta con miembros de todos los sectores, que van desde ladrillos, suministros de oficina, impresoras y servicios de lencería.

El objetivo de RES es ofrecer herramientas para impulsar las economías locales a través de las PYMEs y los comerciantes autónomos. Por esa razón, las grandes empresas, las organizaciones gubernamentales nacionales y las multinacionales no pueden participar en la red.

Inicialmente, RES operaba solo entre PYMEs, pero desde 2008, la compañía agregó tarjetas prepago para consumidores para crear un intercambio más vivo en la red RES. Desde 2012, la compañía extendió sus actividades hacia Cataluña - España, centrándose en Girona.

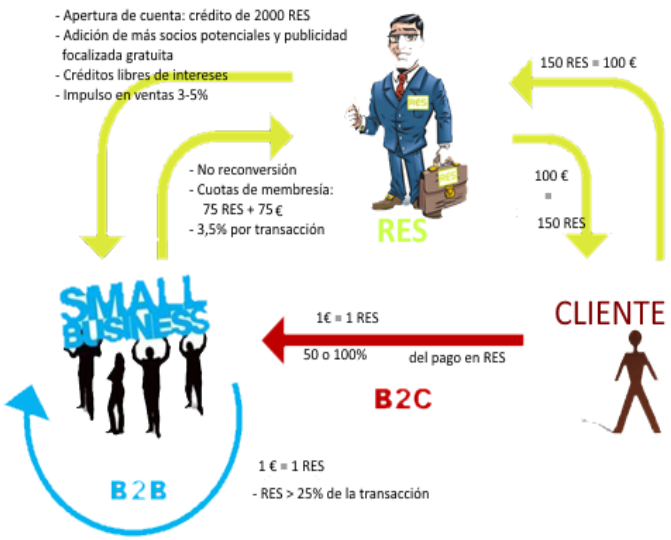
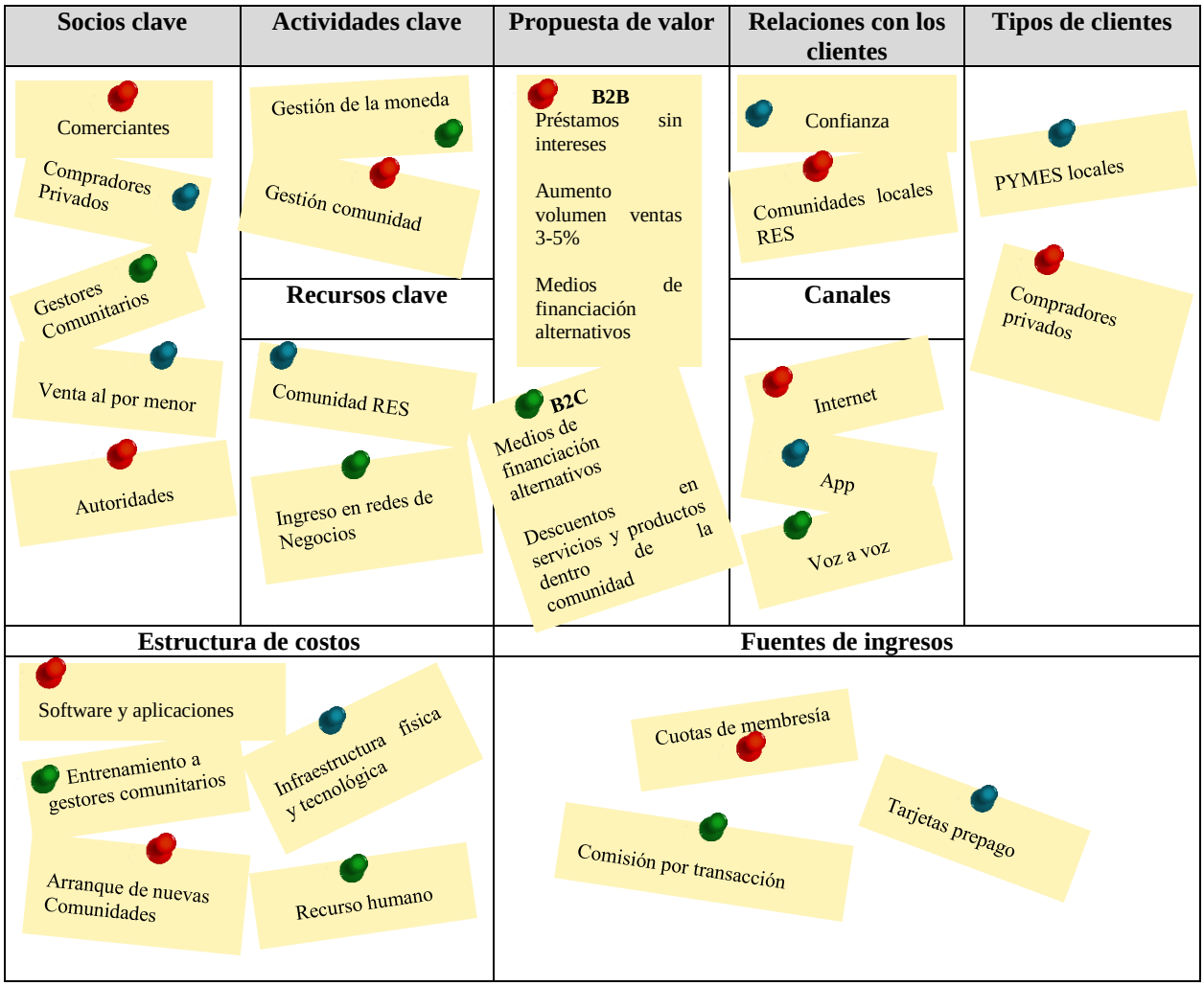


Figura 1. Proceso de operación de la moneda digital RES en sus relaciones B2B y B2C

Fuente: Deliverable 3-2: Commercial and Business Aspects, Vircoin2SME Project

A continuación en la Tabla 1 se presenta el lienzo Canvas de su modelo de negocio. El lienzo CANVAS presenta de manera visual los socios clave, las actividades y recursos clave, la propuesta de valor, las relaciones con los clientes, los canales de distribución, los tipos de clientes, la estructura de costos, la infraestructura y las posibles fuentes de ingresos.

Tabla 1. *Lienzo CANVAS del modelo de negocio de la moneda digital RES*



Fuente: Deliverable 3-2: Commercial and Business Aspects, Vircoin2SME Project

Eurakos

El Centro Easy Innova de Girona (España) implementó en 2015, una moneda virtual llamada *Eurakos* representada por el símbolo € y que funcionaba como una aproximación de los llamados "contratos digitales". Era un entorno de negociación simple entre consumidores y proveedores y se desarrolló para ser utilizado sólo a través de dispositivos móviles. Cada tipo de usuario tenía su propia interfase.

La operación de la moneda es la siguiente: los usuarios (consumidores) deciden la cantidad de dinero a invertir por adelantado en Eurakos (equivalencia 1 Euro = 1 Eurako) para comprar bienes y / o servicios en la red en un período de tiempo específico (preferiblemente hasta 6 meses); la plataforma, a través de un esquema de subasta busca las diferentes posibilidades ofrecidas por los comerciantes en la red sobre esas necesidades del consumidor y se las presenta al consumidor organizadas por precios y ubicación en la localidad (barrio, zona, etc.). El cliente decide entonces la opción más conveniente teniendo en cuenta la calidad, la cantidad y los descuentos ofrecidos durante el tiempo asignado. Esto le asegura ingresos fijos a las PYMEs participantes y oportunidad de fidelización de los clientes.

Por ejemplo, si el consumidor sabe que cada semana gasta 10 € en la tienda de frutas y verduras del barrio, entonces, puede hacer una cesta de 20 €, así puede programar comprar en la tienda de frutas y verduras por dos semanas. El consumidor puede recibir adicionalmente, porcentajes de dinero extra de los proveedores como recompensa a esta planificación para que pueda obtener más producto por el mismo dinero.

El resumen de la operación de Eurakos se presenta en la Figura 2 y se describe a continuación:

1. El consumidor debe comprar Eurakos con Euros (1 € = 1 Euro) para comenzar a operar.

2. El consumidor crea una cesta con los tipos de productos, la cantidad de dinero para gastar en cada uno de ellos según sus necesidades y la fecha límite para hacer la compra (ver Figura 2-(2) que muestra dos cestas de categorías "Bar de copas" y "frutería" respectivamente).
3. Los proveedores del sector del barrio reciben una notificación con la información producida en 2., y luego como si fuera una subasta, hacen diferentes ofertas tratando de coincidir con las preferencias de los consumidores.
4. Los proveedores establecen sus ofertas con descuentos que pueden dar por la cantidad de dinero establecida por el consumidor (véase la Figura 2-(3) que muestra el conjunto de ofertas para la categoría de "frutería" con descuentos de 10% – esto significa 10% más de Eurakos).
5. El consumidor elige la única oferta que piensa que encaja con sus intereses, es decir, precio, ubicación, calidad, etc., observar en la Figura 2-(4), la selección de la oferta del "Petit mercat". En este caso, el consumidor decidió gastar 5€ en el "Petit mercat" pero como este negocio había ofrecido 10% más de dinero para sus compras (0,50 más de €), realmente le quedan 5,50€ para gastar en este mercado.
6. El consumidor acepta la oferta y entonces el contrato se establece (ver en la Figura-2-(5) el contrato establecido con el comerciante del "Petit mercat").
7. El consumidor puede resolver el contrato dentro de las fechas establecidas (es decir, de 11/10/2015 hasta el 24/11/2015), esto significa que puede realizar transacciones con base en este acuerdo hasta que se cumpla el plazo.

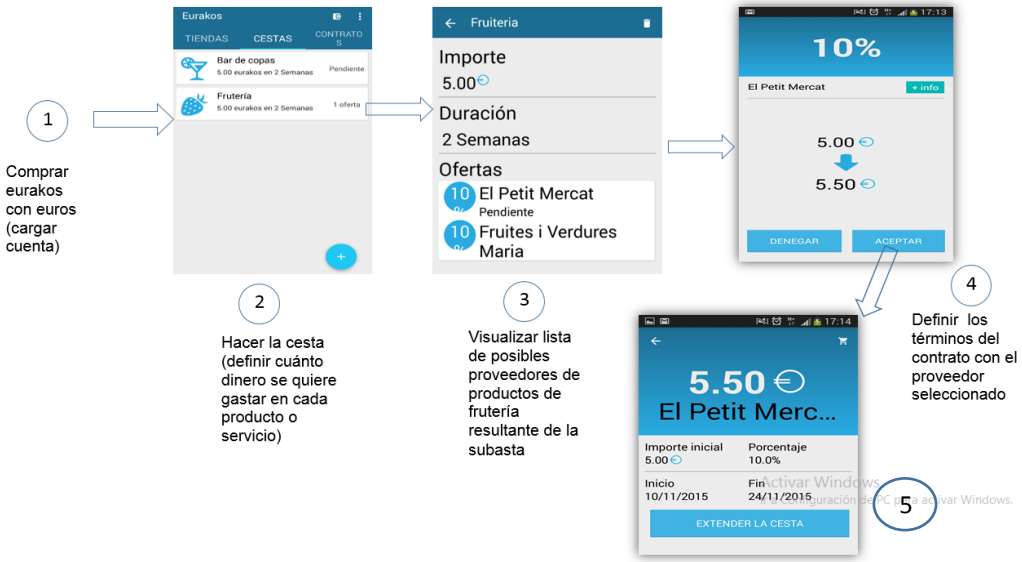
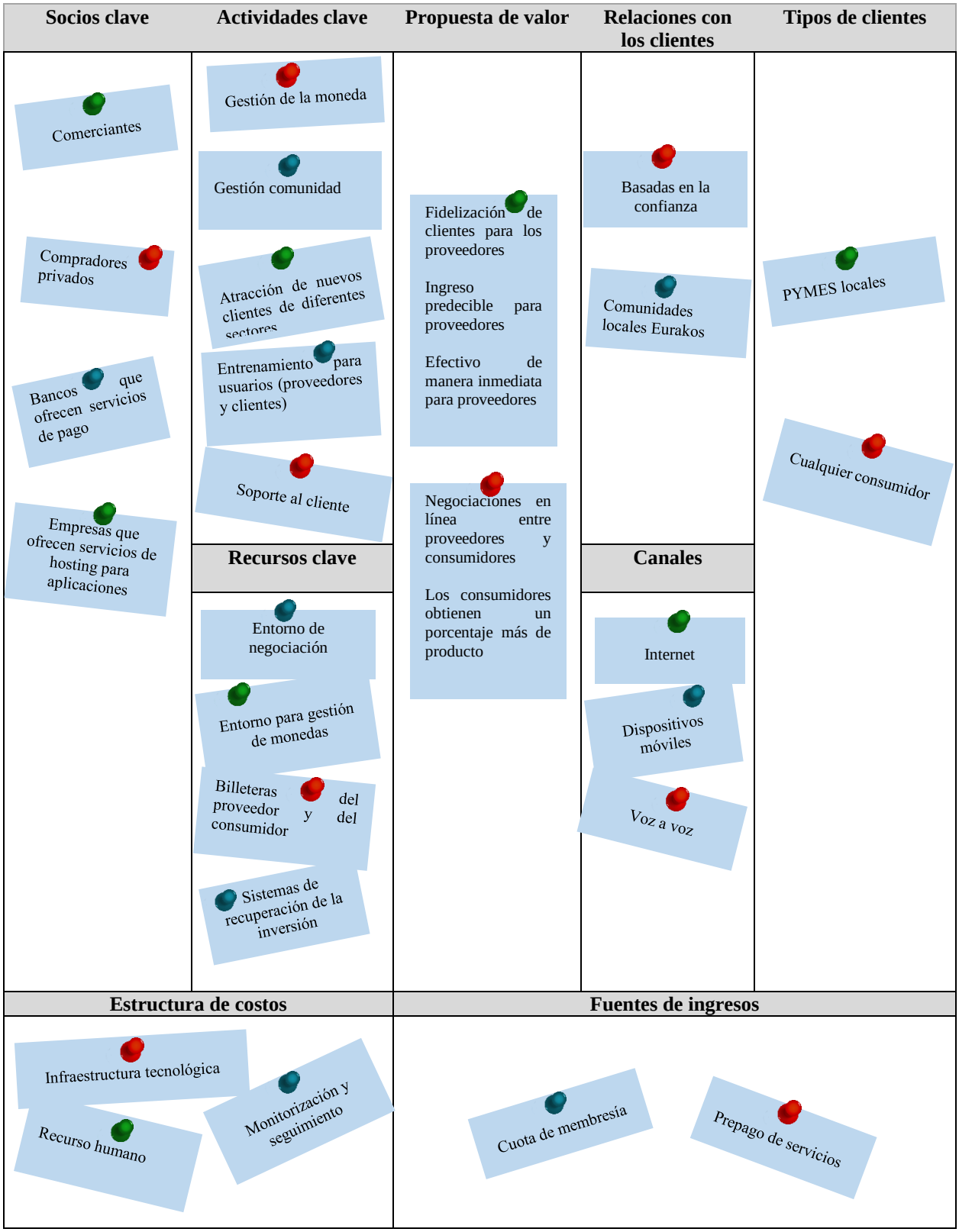


Figura 2. Proceso para crear un contrato digital con la moneda virtual Eurakos

Fuente: Carrillo, Peña and De la Rosa (2016)

La desagregación de la propuesta de valor de esta moneda se puede apreciar en la Tabla 2 mediante el lienzo CANVAS.

Tabla 2. *Lienzo CANVAS del modelo de negocio de la moneda virtual Eurakos*



Fuente: Deliverable 3-3: Applications of virtual currencies, Vircoin2SME Project

2.3 Barreras de adopción de monedas sociales, complementarias y virtuales

2.3.1 Introducción

La experiencia en el uso de la moneda RES en Bélgica y Girona, y Eurakos en Girona, por cada uno de los investigadores asignados al Proyecto Vircoin2SME que tuvieron la oportunidad de unirse al sistema, permitió corroborar la teoría con la práctica en tiempo real. Los participantes adquirieron la tarjeta de cliente RES o una cuenta de Eurakos, y luego la utilizaron en los establecimientos pertenecientes al sistema monetario complementario correspondiente, con el fin de encontrar fortalezas y/o barreras que podrían presentarse en el funcionamiento de las mismas.

Adicionalmente, los usuarios del sistema (administradores, PYMEs y consumidores) mediante encuestas y entrevistas, expresaron sus opiniones sobre los factores involucrados en el uso de estas monedas desde el punto de vista emocional, ambiental, tecnológico y de gestión según su contexto.

La identificación de barreras para una aplicación apropiada de sistemas monetarios virtuales, sociales y complementarios (incluidos aquellos que no tienen una dimensión monetaria directa), es un desafío. Por lo general, esas barreras se concentran en aspectos técnicos que derivan en conclusiones sobre aspectos no-técnicos. No obstante, hay muchas consideraciones que se pueden hacer teniendo en cuenta una tecnología genérica, sin tener que excavar demasiado en un dominio de aplicación específico.

A continuación se introducen las barreras identificadas más representativas y las posibles soluciones encontradas para reducirlas.

2.3.2 Descripción de barreras de tipo emocional

Con respecto a los dominios del diseño Persona-Moneda-Interacción y Dinero, el estudio de Wang y Mainwaring (2008) muestra que en este caso y contexto particular, las implicaciones para el diseño y desarrollo de aplicaciones, los aspectos de gamificación, los instrumentos de motivación y, principalmente, los criterios de utilidad y usabilidad son importantes.

La manera y las metodologías sobre cómo se percibe, obtiene y gasta una moneda complementaria pueden influir en el comportamiento y experiencia de los usuarios al utilizar una aplicación.

Las monedas virtuales y las reales se pueden correlacionar de maneras complejas para que promuevan, amplíen y/o interfieran con el valor y el carácter del mundo de los elementos y aplicaciones. Traer dinero real al diseño de la interacción Persona-Moneda puede aumentar los problemas existentes de realidad, confianza y equidad, y por tanto presentar nuevos desafíos y oportunidades para la innovación de la experiencia del usuario.

A continuación, se presentan los factores psicológicos y emocionales relevantes que influyen en el uso de la moneda virtual, lo cual permite analizar las características de éxito y fracaso de las aplicaciones y esquemas existentes y emergentes. El análisis comienza con la descripción de algunos resultados de un estudio sobre la creación de equipos virtuales, así como las barreras y obstáculos identificados.

El advenimiento del nuevo siglo ha mostrado nuevas tendencias para los seres humanos, las aplicaciones, los sistemas y, principalmente, la forma en que las personas interactúan. Los seres humanos solían vivir en grupos desde un comienzo, no como personas independientes. Así que trabajar en grupos, intercambiar conocimientos, intercambiar productos y aprender de los trabajos y resultados de cada uno fue fundamental para el proceso de convertirse en aquello que se denomina *Homo sapiens*. A medida que las cosas han cambiado hacia una digitalización de la sociedad, el trabajo en grupo en el nuevo siglo, observa nuevas tendencias en las organizaciones y la administración tratando de acomodar personas y procesos, al cambiante entorno global.

Un grupo de personas que trabajan juntas virtualmente, se puede describir como un Equipo Virtual. Sólo se pueden considerar personas que trabajan juntas, uniendo fuerzas, en una tarea común y cruzando las fronteras del espacio, el tiempo, las organizaciones y la cultura, con el apoyo de herramientas y redes TIC. Dichos grupos y equipos ofrecen una variedad de posibilidades; sin embargo, su funcionamiento no siempre es fructífero o incluso posible.

Las barreras para el trabajo cooperativo virtual, aparecen especialmente en los sistemas asincrónicos como mensajería, *eHealth, eBusiness, eGovernment*, etc. La calidad de la comunicación es menor porque las señales no verbales son limitadas - una persona que se comunica no ve a la otra persona, pierde el lenguaje corporal de la otra persona -, sin embargo, puede observar mientras tanto, algunos intentos de expresar sentimientos usando sustitutos de emociones (emoticones). En consecuencia, el trabajo virtual genera problemas con el conocimiento del contexto y con la comprensión compartida.

La comunicación puede orientarse más a tareas y volverse más empresarial ya que parece fácil olvidarse de las relaciones sociales y centrarse sólo en la tarea. Otro problema potencial es la falta de habilidades del grupo virtual o de los miembros del equipo para el trabajo telecooperativo, como la autogestión y las habilidades interculturales, la confianza interpersonal, las habilidades para usar nuevas tecnologías y para establecer contactos virtuales con otros.

La ciencia del conocimiento contiene al menos dos categorías generales, el impacto de los medios y el uso de los medios. El impacto de los medios describe lo que los medios hacen con los seres humanos, en un sentido físico, mental o psicológico. La segunda categoría aborda la pregunta de qué hacen los seres humanos con los medios, y cómo los utilizan.

Una de las categorías más importantes y relevantes para evaluar el éxito de una aplicación en términos de aceptación, es el par compuesto por la usabilidad y la utilidad. La usabilidad se define, por un lado, como la medida en que un producto puede ser usado por usuarios específicos para alcanzar objetivos específicos con efectividad, eficiencia y satisfacción. La aplicación, la usabilidad del sistema y la utilidad son propiedades interdependientes de la interacción entre la aplicación y el sistema, que en combinación determinan la satisfacción y el uso del sistema.

2.3.3 Descripción de barreras relacionadas con la administración de las monedas

En el análisis de las encuestas desarrolladas, se identificaron tres barreras comunes para las PYMEs y los administradores de las monedas. Estas son: la fidelización de los clientes dentro del ecosistema de cada moneda complementaria, la gestión de la moneda en sí misma y el aseguramiento contable en términos de registros, informes financieros y procesos impositivos (en cumplimiento de las normas y regulaciones vigentes).

Esfuerzo involucrado en la gestión de la moneda y la viralización de su uso

Sin duda, la gestión de la moneda es una de las barreras más importantes para los tipos de sistemas de monedas complementarias. Esto se debe a que el mantenimiento y la viralización son dos claros indicadores en términos de confianza y satisfacción percibidos por los comercios (PYMEs) y los usuarios (Clientes).

Estas palabras clave *Confianza y Satisfacción* son definitivamente parte de la columna vertebral de cada sistema de moneda alternativo (especialmente al comenzar a implementar una nueva moneda virtual) y deben analizarse no sólo en términos de las barreras emocionales descritas anteriormente, sino también cuando este nuevo sistema se inicia en una red local o comunitaria. En tal caso, es importante el conocimiento de las operaciones/servicios, el compromiso de cumplir con las reglas del juego, de parte del Administrador/Gerente de la Plataforma y de cada PYME que se une a la red.

Por lo tanto, el sistema podría crear y construir una reputación respetada que fomente la incorporación de muchos otros nuevos actores (PYMEs, consumidores y otros miembros).

Baja fidelidad del cliente al inicio de la experiencia

La barrera de fidelización del cliente, es fundamental, principalmente, durante los primeros años de lanzamiento de un nuevo ecosistema/red y es un tema definitivamente crucial que puede definir el éxito o el fracaso. Para ello, se hace necesario planificar y desarrollar estrategias para difundir y presentar los valores y la misión del nuevo sistema de moneda complementaria virtual o VCC.

Aseguramiento contable

Existen barreras intrínsecas que la mayoría de los sistemas VCC, deben solucionar para llevar a cabo actividades de capacitación y control apropiadas que prevengan la existencia de irregularidades y controversias con respecto al cumplimiento de la normativa vigente en términos de procesos contables y tributarios, y en el comportamiento de los miembros dentro de la comunidad o red social.

Alta complejidad del servicio a ser comunicado

Algunas entrevistas y actividades de observación realizadas durante el Proyecto Vircoin2SME en Cataluña, mostraron que la mayoría de las PYMEs contactadas tenían escepticismo por la complejidad del funcionamiento de cada sistema de moneda complementaria (RES.cat / Eurakos) y sólo unos pocos miembros estaban satisfechos con la forma de entender y transmitir las reglas del juego del sistema a otros miembros y sus clientes. Se observó, adicionalmente, que este pequeño grupo incluía a los miembros con el mayor número de transacciones dentro del ecosistema monetario complementario.

Usabilidad

Una barrera que a veces enfrentan los sistemas de negocios mediados por monedas sociales, complementarias y virtuales está relacionada con encontrar los medios apropiados para lograr su misión y sus objetivos (usabilidad). Por ejemplo, los grandes avances tecnológicos hacen creer a las comunidades que no existe más opción que estar "en línea" todo el tiempo. Sin embargo, eso no es posible especialmente en los países en desarrollo y emergentes, y en algunas áreas de los países desarrollados donde las personas no tienen acceso a la virtualidad bien sea por problemas de conectividad o de equipos.

Día tras día es evidente la diversidad de la tecnología y los nuevos canales de información que han surgido. Para los sistemas de monedas complementarias este no es un escenario diferente, debido a que los miembros, consumidores y otros actores de cada ecosistema, también están enfocados en comprender cuán intuitivo y novedoso podría ser el utilizar monedas sociales y complementarias a través de cuentas en línea, aplicaciones móviles y otras posibilidades.

Uso marginal

Las transacciones realizadas con RES y Eurakos en Girona, alcanzaron menos del 0,01% del PIB en Cataluña, lo cual hace que estos sistemas no sean económicamente representativos en la región. Aquí podría definirse como una barrera en términos del alcance e importancia relativa en la magnitud económica. Por supuesto, el RES en Bélgica es una moneda consolidadada, ya que en 2011 el volumen de transacciones alcanzó los 35 millones de Euros[18].

Tiempo de recuperación del dinero

Esta barrera tiene que ver con las restricciones implícitas que surgen cuando un cliente o miembro de la red de monedas complementarias desea canjear una moneda social / complementaria con una moneda fiduciaria, con el fin de acceder a un bien o servicio que no es ofrecido en la red. El cambio no es inmediato y el cliente debe esperar un tiempo considerable para redimir su moneda complementaria en dinero de curso legal.

2.3.4 Descripción de barreras relacionadas con el entorno en donde operan las monedas

Además de las barreras genéricas que existen para cada uno de los productos, sistemas, aplicaciones o servicios, cada proyecto de implementación relacionado con monedas complementaarias identifica sus propias barreras adicionales.

En el contexto del Proyecto Vircoin2SME que recopiló información por medio de encuestas aplicadas a las partes interesadas de la moneda complementaria o información adquirida por observación directa, se encontraron los siguientes resultados con respecto a la existencia de barreras relacionadas con el entorno para la adopción de las monedas complementarias o virtuales:

[18] Deliverable 3.2: Commercial and Business Aspects. Vircoin2sME Project

1. En el ámbito de las PYMEs, las barreras tienen que ver con los niveles comparativamente bajos de tecnología y recursos de inversión disponibles. Estos problemas incluyen igualmente la falta de estándares existentes para la adopción de las monedas complementarias por un lado, y los altos costos de producción de las mismas, por el otro. Además, las PYMEs declaran no tener suficientes recursos para financiar, capacitar y utilizar la tecnología apropiada.

Las comunidades de PYMES analizadas consideran como el principal obstáculo, el no tener un nivel suficiente de experiencia con el esquema de desarrollo de monedas sociales, complementarias y virtuales, así como el desconocimiento del alcance jurisdiccional para fines legales. También afirman tener dificultades de integración con las comunidades debido a las ideologías existentes en términos de antecedentes políticos, sociales y culturales y, finalmente, piensan que la incapacidad de comerciar e intercambiar la moneda complementaria con la moneda de curso legal es otra barrera importante que debe tenerse en cuenta.

2. A nivel gubernamental en las regiones y países de los participantes en el Proyecto, fue interesante identificar el temor por el establecimiento de estructuras descentralizadas de gobierno autónomo con el fin de cooperar en términos de monedas sociales, complementarias y virtuales, especialmente por la inexistencia de políticas y estrategias para controlar el desarrollo, la implementación y la permanencia en el mercado de este tipo de moneda.

3. Las instituciones financieras involucradas en este tipo de proyectos, mostraron su preocupación en aspectos relacionados con la regulación que se aplica únicamente a la moneda de curso legal y no a ninguna otra moneda, especialmente las monedas virtuales. Por lo tanto, la falta de políticas y estrategias para crear y ejecutar con éxito un sistema virtual de moneda social complementaria se considera una barrera debido al riesgo de generar crisis políticas o económicas que repercutan en la inestabilidad de las regiones y los países.

4. Al entrevistar a turistas, éstos comentan que la principal barrera que identifican es la incapacidad de cambiar la moneda complementaria a moneda de curso legal en el momento que lo deseen, y que tampoco sería posible “regatear” en las tiendas con una moneda complementaria.

En resumen, el proyecto Vircoin2SME no encontró tantas barreras diferentes en relación con el entorno donde operan las monedas sociales complementarias y virtuales. En general, lo más importante para mencionar es la ausencia de un marco legal y político para el funcionamiento de este tipo de monedas, así como la falta de experiencia de los administradores de estos sistemas y las PYMES en el diseño y uso de las mismas. Asimismo, el esfuerzo que implican los gerentes de estas monedas en capacitación y acompañamiento en el seguimiento de su operación es una limitación para la adopción de estas iniciativas en los negocios.

2.3.5 Descripción de barreras relacionadas con la infraestructura y la tecnología

Es importante señalar que las barreras aparentes en esta clasificación son las relacionadas con la conectividad, no sólo en términos de infraestructura de redes y comunicaciones deficiente (si existe), especialmente en algunas regiones de países en desarrollo, sino también, en el alto costo de los dispositivos (computadores, portátiles, teléfonos móviles, etc.) utilizados para acceder a los servicios ofrecidos por redes de monedas virtuales. Esto significa que el desarrollo de negocios mediados por este tipo de monedas sólo puede hacerse eco en parte de la población, es decir, son soluciones excluyentes desde el punto de vista social (es decir, el que no disponga de la tecnología requerida no puede participar en estas iniciativas).

Por otro lado, aún existe una brecha en las competencias digitales en los actores del sistema, que no permiten explotar los recursos tecnológicos disponibles de manera eficiente.

Muchas acciones emprendidas por los gobiernos para el desarrollo de estas competencias se incluyen en los programas de educación a diferentes niveles. Sin embargo, es importante involucrar a las personas en el "aprender haciendo" sobre este tema, a fin de acelerar su apropiación.

En este punto, por supuesto, existen las barreras relacionadas con la usabilidad de la moneda, es decir, con utilizar los medios correctos en el contexto correcto y usar además, diseños intuitivos (problemas que los diseñadores de monedas complementarias deben abordar teniendo en cuenta el contexto de las partes interesadas).

La siguiente tabla, resume las barreras identificadas para la adopción de sistemas VCC en los negocios.

Tabla 3. *Resumen de barreras de adopción de monedas virtuales complementarias desde el punto de vista de los usuarios del sistema*

BARRERAS EMOCIONALES				BARRERAS ADMINISTRATIVAS				BARRERAS DE ENTORNO				BARRERAS TECNOLÓGICAS			
Descripción	**Adm**	**PYME**	**Con**	**Descripción**	**Adm**	**PYME**	**Con**	**Descripción**	**Adm**	**PYME**	**Con**	**Descripción**	**Adm**	**PYME**	**Con**
1. Miedo a la aceptación	1	1	3	**12.** Esfuerzo en la administración de la moneda y la viralización de su uso	12	12	19	**20.** Regulaciones legales	5	21	5	**23.** Conectividad	5	23	23
2. Miedo a la opacidad	2	3	4	**13.** Baja fidelización inicial de clientes	13	13		**21.** Bajo nivel de competencias ciudadanas	20	22	21	**24.** Bajo nivel de competencias digitales	16	24	24
3. Hábito de utilizar la moneda de curso legal como único medio de pago		4	5	**14.** Aseguramiento contable	14	14		**22.** Movilidad restringida	21		22		23		
4. Egoismo		5	6	**15.** Alta complejidad en la comunicación del servicio	15	18			22						
5. Seguridad y privacidad		6	7	**16.** Usabilidad: uso de los medios correctos en el contexto correcto	16	19									
6. Independencia			8	**17.** Usabilidad: diseño intuitivo	17										
7. Miedo a lo desconocido			9	**18**. Uso marginal											
8. Miedo a experiencias pasadas			10	**19**. Tiempo de recuperación de la moneda (cambio a moneda fiduciaria)											
9. Fidelidad a una marca o comercio			11												
10. Sospechas en torno a los servicios y productos															
11. Miedo al engaño															

Adm: Administradores de la moneda **Con**: consumidores o clientes en la red comercial mediada por la moneda

Nota: los números en las columnas indican el número que describe la barrera, por ejemplo, el número 1 en Adm bajo las barreras emocionales significa que la barrera corresponde a “Miedo a la aceptación”, igual para las PYMES y así sucesivamente.

2.3.6 Recomendaciones para reducir las barreras de adopción de las monedas sociales, complementarias y virtuales

Según afirma Rogers (2011) si se quiere mejorar la tasa de sostenibilidad de las monedas sociales/complementarias/virtuales, se deben aumentar los niveles de competencia de los actores del sistema tanto para el diseño de las monedas, como para su uso. Estas posibles soluciones pueden orientarse a la búsqueda de mejores formas para transferir conocimiento sobre el tema y a una educación efectiva que le apunte a una mejor comprensión de las reglas de juego de cada aplicación, para los modelos de negocio mediados por este tipo de monedas.

Con respecto a la motivación o fidelidad de los usuarios, aún hay mucho trabajo por hacer para encontrar formas de alcanzar un mayor impacto al utilizar monedas complementarias. Tal vez, los nuevos avances tecnológicos son la clave para hacer de este gran salto una realidad. Es por eso que el uso de dispositivos móviles, la creación de promociones y el desarrollo de fuertes estrategias de mercadotecnia, deberían aplicarse en paralelo con nuevas mecánicas de juegos, para involucrar y crear sentido de pertenencia y compromiso en los diferentes miembros del sistema, no solo para usar el sistema en sí, sino también para difundir la información "voz a voz" con familiares, amigos y colegas.

Para los sistemas de monedas complementarias latinoamericanos, una posible recomendación sería la creación y diseminación de guías prácticas o tutoriales para los miembros. Al hacerlo, se les permite en primer lugar obtener un marco de entendimiento común para el tratamiento contable a utilizar para registrar todas las transacciones (incluidas las que utilizan las monedas complementarias), así como una herramienta diaria para obtener información sobre posibles cambios en las reglamentaciones, estándares y cualquier otro tema que pueda afectar los procesos contables, financieros o tributarios de las diferentes PYMEs que formen parte de la red.

En cuanto a la barrera de la usabilidad para la moneda complementaria, en términos de utilizar los medios correctos en el contexto correcto, se debe continuar trabajando con base en diferentes esquemas de presentación tales como tarjetas prepago, billetes, monedas y otros mecanismos que estén protegidos y respaldados de manera segura según la naturaleza del sistema de moneda complementaria y que evolucionen en concordancia con el contexto del área local.

Por ejemplo, Walter Smets (fundador de la moneda RES), expresa bajo interés, por el momento, en el pago mediante dispositivos móviles, para su moneda. Según su estudio y comprensión del mercado local en Lovaina (Bélgica), ya que es una tecnología demasiado novedosa, la aceptación es baja y los usuarios necesitarían de mucha capacitación para aprender a utilizarla, lo que supondría una gran inversión. Igualmente afirma que es una iniciativa que se incluiría más adelante cuando su público esté preparado[19].

Para las soluciones de inclusión social en contextos donde las competencias ciudadanas son bajas o no se han desarrollado en la población, es obligatorio diseñar estrategias para consolidarlas, mediante el aprendizaje formal y no formal (aprendizaje permanente) tal como se consigna en la Declaración de París sobre la promoción de la ciudadanía y los valores comunes de libertad, tolerancia y no discriminación, que en 2015, fue adoptada por los Ministros de Educación Europeos, y que también se ha extendido a los países en desarrollo como Colombia y Ecuador en América Latina.

La solución para reducir la barrera de la usabilidad de las aplicaciones de monedas virtuales podría estar orientada al diseño de la interfaz del usuario que debería seguir las leyes Gestalt (diseño intuitivo) (Shimpeno and Ezer, 2014) y el principio KISS (diseño simple). Por otra parte, se requiere el uso de estrategias que muestren claramente a las partes interesadas los beneficios para gestionar transacciones seguras, más rápidas y sin comisiones.

[19] Deliverable 2.3: Report on complementary currencies research. Vircoin2SME project

Para mejorar las estrategias de conectividad de las regiones de cada país, la academia puede aportar conocimiento participando en la realización de proyectos de planes estratégicos de incorporación de TIC y orientando la toma de decisiones en aspectos relacionados con el dimensionamiento de la infraestructura de conectividad requerida de acuerdo con el contexto.

La siguiente Tabla resume las recomendaciones antes mencionadas para reducir las barreras de adopción de las monedas complementarias virtuales.

Tabla 4. *Recomendaciones para reducir las barreras de adopción de monedas virtuales complementarias*

ACCIONES PARA REDUCIR BARRERAS EMOCIONALES	ACCIONES PARA REDUCIR BARRERAS DE ADMINISTRACIÓN	ACCIONES PARA REDUCIR BARRERAS DEL ENTORNO	ACCIONES PARA REDUCIR BARRERAS TECNOLÓGICAS
1. Diseñar módulos para mostrar a los usuarios (comerciantes y consumidores) la manera correcta de utilizar el dinero de curso legal y las VCC, entendiendo las reglas del juego en términos de administración de los modelos de negocio.	1		
2. Producir conocimiento relacionado con los principios básicos de las monedas complementarias dando la oportunidad a la comunidad de aprender haciendo.	2		
3. Entender la misión para utilizar una moneda complementaria	**7.** Promover el uso de las criptomonedas para reducir el tiempo de recuperación del dinero	**10.** Diseñar y poner en marcha estrategias de inclusión social mediante el uso de monedas sociales locales (redes de trueque o bancos de tiempo), especialmente para contextos Latinoamericanos.	**11**. Evaluar apropiadamente el diseño de las interfaces de las aplicaciones móviles para las monedas virtuales considerando las reglas GESTALT y el principio KISS (diseño simple e intuitivo)
4. Promover el respeto por la marca o el comercio	**8.** Encontrar y comunicar argumentos racionales para el beneficio del uso de VCC		
5. Alentar el uso estratégico de las VCC aplicando mecánica de juegos para vincular y motivar a los miembros de la red (fidelizar clientes)	**9.** Encontrar analogías de retroalimentación de los clientes sobre la operación de la moneda basada en la compensación de esfuerzos - estrategia didáctica		
6. Diseñar estrategias específicas de mercadeo orientadas a líderes de opinión y a comentarios de los clientes			

2.4 Criptomonedas

2.4.1 Definición

Las criptomonedas son una evolución de las monedas complementarias virtuales que surgieron con el propósito de realizar transacciones a través de Internet utilizando información digital encriptada mediante códigos *hash* (Lee, 2015). Esto permite transferencias seguras e intercambio de bienes digitales de forma distribuida y descentralizada.

Según Graydon (2014), algunos de los beneficios de su uso para los comerciantes en línea son: monetizar nuevos mercados, tarifas de transacción más bajas que utilizando la banca tradicional, las transacciones se reflejan al instante y no hay devoluciones de cargos y seguro.

En la siguiente tabla se presenta un comparativo con la moneda de curso legal que resalta sus diferencias en temas relacionados con el control, la transparencia, el anonimato, la manipulación y el formato que manejan.

Tabla 5. *Comparativa en atributos de la moneda de curso legal con la criptomoneda*

Moneda	**Control**	**Transparencia**	**Anonimato**	**Manipulación**	**Formato**
Curso legal	*Centralizado*: existen unos pocos actores como los bancos centrales, que tienen el control absoluto sobre ella.	*Poco transparente*: el control de cuánto dinero tiene cada individuo está en mano de entidades privadas.	*No existe*, puesto que todas las transacciones y datos de los cuentahabientes los conoce el banco.	*Es posible*: las entidades que controlan su uso podrían alterar algún importe de una cuenta en sus bases de datos sin informar a nadie.	Físico a manera de billetes y monedas
Criptomoneda	*Descentralizado*: No existe una entidad o individuo que pueda alterar su precio o que pueda emitir más moneda al mercado.	*Transparente*, puesto que existe un libro de transacciones global que cualquiera puede descargar y consultar, cualquier individuo puede ver el saldo de todas las carteras y cuales han sido los movimientos previos para conseguir esa suma. Esta información no se puede falsear, por lo que favorece la transparencia en casos de corrupción.	*Si existe*, puesto que cualquiera puede crear una cuenta y empezar a operar en el mercado, sin necesidad de entregar sus datos personales a nadie. Además, crear una cuenta es totalmente gratuito, y se puede hacer desde cualquier lugar que disponga de conexión a Internet.	*Casi imposible*: es decir, ningún individuo puede alterar el importe de una cartera de forma fraudulenta. Esto depende de los modelos matemáticos que encriptan la información que hoy por hoy son robustos.	Digital

Fuente: adaptado de: https://quesoncriptomonedas.org/por-donde-empezar/

2.4.2 Características

Las características más comunes de las criptomonedas son: validez, identidad, aceptación o reconocimiento, movilización de la economía, cantidad limitada en circulación, almacenamiento, actualización y creación de nuevas unidades. A continuación, se describen estas características con más detalle.

Validez

El funcionamiento de todas las criptomonedas se soporta principalmente en dos elementos: la *tecnología Blockchain* que es la infraestructura robusta e innovadora que sustenta su operación, y *la comunidad* que participa tanto en la ejecución de transacciones como en la validación de las mismas.

Identidad

Tanto los validadores como los usuarios que realizan transacciones están identificados con un código *hash* único. El uso de este identificador viene determinado por una clave pública (el hash) y una clave privada definida por el usuario en el momento de crear su identidad. Si se pierde este identificador o el acceso a él, se pierden todos los recursos que han sido administrados por ese usuario.

La mayoría de las criptomonedas usan billeteras, que es un software que almacena direcciones y claves secretas. Las direcciones permiten recibir nuevas unidades de monedas y las claves permiten aprobar transacciones que transfieren monedas a otras partes. Para la mayoría de las criptomonedas, sus fondos son tan seguros como mantener su billetera. Las billeteras que se mantienen en línea con intercambios de criptomonedas tienden a ser objetivos principales para los piratas informáticos, por lo que esta práctica debe evitarse tanto como sea posible (Block Fortune, 2017).

Aceptación o Reconocimiento

Las criptomonedas han tenido tanta aceptación que hoy en día es un negocio que mueve millones de dólares generados por el cambio de estas, a moneda de curso legal. Este suceso es comparado por los expertos con la revolución que generó la llegada del Internet al mundo, sin embargo, el proceso de permitir el cambio a moneda de curso legal ha hecho que algunas personas inviertan en criptomonedas con fines de especulación más que para la adopción de las monedas como medio de intercambio de productos y servicios (Block Fortune, 2017).

La tecnología blockchain que se introducirá en la sección 2.4.4, como herramienta que ofrece seguridad, descentralización, escalabilidad y agilidad, ya está siendo utilizada por grandes empresas como Deloitte e IBM, y tanto ha sido su impacto que algunos bancos y gobiernos están analizando la posibilidad de implementarla igualmente, para agilizar sus procesos diarios y permitir una mayor transparencia en sus actividades (Marvin, 2018).

Movilización de la economía

La razón principal de la creación de la primera criptomoneda (Bitcoin) fue la de optimizar las actividades económicas cotidianas, ya que los intermediarios (bancos e instituciones financieras), además de la gran cantidad que son hoy en día, han adquirido mucho poder y obligan a los usuarios a esperar varios días para la realización y validación de una transacción.

Los sistemas basados en el Blockchain permiten quitarle protagonismo a los bancos y darle la oportunidad a muchos más participantes de validar transacciones. Con esto, además de disminuir el tiempo de resolución de la transacción de varios días a unos cuantos minutos, se reduce el costo de la validación.

Cantidad limitada en circulación

Algunas monedas determinan la cantidad total de la misma en circulación desde el momento en que se crean. Sin embargo, sólo se deja disponible una parte, mientras el resto va apareciendo a medida que se validan transacciones. Las monedas que siguen el patrón del Bitcoin van generando nuevas cantidades al realizar cada validación y la velocidad de crecimiento depende de la velocidad de validación de las transacciones.

Almacenamiento o actualización

Toda la arquitectura de las criptomonedas se basa en un sistema distribuido, de manera tal que todos los nodos están en constante sincronización para poder funcionar, pero la información particular del identificador y el acceso de cada usuario se debe cuidar, ya que si se pierde no hay forma de recuperarlo nuevamente.

Creación de nuevas criptomonedas

El Bitcoin fue la primera criptomoneda y por ello, las nuevas utilizan como base su diseño y se enfocan en mejorar algunos aspectos de su funcionamiento, como por ejemplo:

a. El algoritmo de consenso, ya que es bien conocido que el PoW consume gran cantidad de electricidad y desde el punto de vista ecológico, es ineficiente.

b. La cantidad de moneda en circulación, para poder crear más monedas sin límite.

c. La ejecución de contratos inteligentes (como en Ethereum) u otras diferentes métricas como aumentar velocidad, reducir el costo de la transacción y conceder más anonimato a los actores, entre otros (Graydon, 2014).

2.4.3 Limitaciones y oportunidades para su uso

Según el informe de la Comisión de Asuntos Económicos y Monetarios del Parlamento Europeo (Weizsäcker, 2016), la tecnología *Blockchain* se está implementando ampliamente en las aplicaciones de innovación en servicios financieros (FinTech), Internet de las cosas, voto en línea, etc. Sin embargo, tiene las siguientes limitaciones y oportunidades importantes:

Limitaciones

:

1. Existe duda en su utilización masiva, pues la ausencia de entidades de control centralizado se cree que pudiera aprovecharse para el uso criminal de lavado de dinero, financiación del terrorismo, evasión de impuestos, etc.

2. Carece de las estructuras de gobernanza flexible y confiable (especialmente en Bitcoin) que podrían conducir a la incertidumbre para la protección del consumidor.

3. Existe una limitada capacidad de regulación en el campo de las nuevas tecnologías, para la definición de garantías que aseguren un funcionamiento correcto y fiable de las aplicaciones *Blockchain* de cara a su crecimiento.

4. Existe inseguridad jurídica debido a falencias en la legislación o incluso por ausencia de una regulación adecuada. El Parlamento Europeo insiste en la necesidad de aumentar la capacidad regulatoria antes de su posible relevancia sistémica.

5. Dificulta la definición de un valor equivalente entre monedas por la existencia de más de cien de éstas monedas alrededor del mundo con diferente tipificación (criptomonedas, monedas sociales, monedas basadas en el conocimiento, etc.).

6. La mayoría de las monedas complementarias virtuales existentes, no tienen una tabla de conversión con la moneda de curso legal. Este aspecto crea otro problema en cuanto a la identificación de la forma de aplicar las reglas de cambio.

7. Es escasa la literatura científica relativa a la aplicación de esta tecnología.

Oportunidades

1. Contribución al desarrollo económico y al bienestar de los consumidores mediante la reducción de los costos de transacción de pagos con respecto a los sistemas tradicionales (los costos totales podrían ser reducidos hasta en 20.000 millones de euros), así como los costos de financiamiento sin una cuenta de acceso bancario tradicional.

2. Aumento de velocidad para la realización de las transacciones

3. Costos de transacción bajos y alto grado de privacidad (no total anonimato pues tiene la posibilidad de seguimiento en caso de infracción).

4. Combinación de sistemas que facilitan su uso, por ejemplo, micropagos en línea en aplicaciones seguras y fáciles de utilizar.

5. Potencial para acelerar, descentralizar, automatizar y estandarizar los procesos con base en datos que podrían modificar las modalidades de transferencia de la tenencia de activos y los registros.

6. Potencial para proporcionar una mayor eficiencia, velocidad y resistencia a los otros procesos de gestión, compensación y liquidación después de la negociación (que actualmente representan más de 50.000 millones de euros al año para el sector financiero mundial).

7. Potencial de Contratos Inteligentes.

Con la presente investigación se pretende reducir el efecto de las limitantes mencionadas en el informe de Weizsäcker, que también aparecieron durante el estudio de las barreras de adopción de este tipo de monedas y aprovechar ampliamente las fortalezas para motivar el uso de las criptomonedas de manera ágil, confiable, segura y sostenible.

2.4.4 Blockchain

Descripción

Según Preukschat (2017), *blockchain* es la evolución natural del actual *Internet de la Información* (representado por compañías como Google, Facebook, etc.), al *Internet del Valor*, que se basa en los siguientes conceptos clave: nodo, protocolo, red entre pares (P2P), y la diferencia entre sistemas centralizados y descentralizados. Un nodo es un computador o servidor que puede conectarse con otros siguiendo unas reglas de comunicación que están definidas en un protocolo. La red que forman estos nodos interconectados se llama red entre pares. En conjunto estas partes constituyen un sistema descentralizado.

Saqib y Saake (2015), definen el *blockchain* como un sistema de almacenamiento de transacciones digitales realizadas en línea, utilizando un formato más seguro y encriptado que el de cualquier transacción por Internet. Los datos una vez encriptados se almacenan en la base de datos a manera de bloques encadenados con una estructura de árbol de *Merkle*.

Esta tecnología también se conoce como la del libro de contabilidad encriptado e inmutable, lo cual significa que las transacciones realizadas pueden verse en todo momento, pero no alterarse o borrarse debido a que la complejidad de romper la encriptación para realizar alguna de estas acciones puede ser tan compleja y tardar tanto tiempo que al final no vale la pena llevar a cabo dicho ataque. Cualquiera puede verificar que la información está registrada allí (mediante una clave pública), ya que aparece con la firma del propietario pero solo el propietario puede desbloquear lo que está dentro del contenedor (mediante su clave privada) (Mougayar, 2016).

Del libro “Blockchain for a new economy” de Melanie Swan (2015) se han rescatado los siguientes conceptos sobre el *blockchain* y su funcionamiento:

a. Es un tema considerado en el mundo como uno de los más innovadores de los últimos tiempos. Inicialmente fue definido como la base intrínseca del funcionamiento de las criptomonedas, pero ya su aplicación se ha extendido a múltiples ámbitos como el de las votaciones (elecciones) o como justificante de existencia y propiedad de documentos, entre otros.

b. Fue creado inicialmente para ser usado como solución de dos problemas principales, los cuales son el *doble gasto* y el *problema general bizantino de computación*. El doble gasto lo soluciona combinando dos tecnologías, el compartimiento de archivos entre pares y la criptografía basada en la combinación de una clave pública, donde la propiedad de la moneda se guarda en un libro contable público (public ledger) y éste es validado o confirmado por protocolos criptográficos. El problema bizantino se refiere a que no se necesita que las dos partes involucradas en una transacción sean confiables o confíen entre ellas pero si es necesario que confíen en la tecnología basada en *blockchain* con la que se esten realizando las transacciones.

c.. Las primeras versiones de criptomonedas se validaban y almacenaban en el Blockchain utilizando un único protocolo de consenso que era el PoW (Proof of Work - prueba de trabajo) con el que nació el Bitcoin, pero hoy en día existen otras que utilizan diferentes métodos como el PoS (Proof of Stake - prueba de existencia de moneda), el PoA (Prueba de actividad o prueba de autoridad- Proof of Activity o Proof of Authority) o la criptomoneda IOTA (para apoyo al Internet de las Cosas) que no utiliza *blockchain* sino otra manera de validar sus transacciones. Más adelante se introducen estos métodos, porque de la combinación de algunos de ellos nace la propuesta que responde a la pregunta de investigación de esta tesis.

d. Ha ido evolucionando desde su formato inicial el *Blockchain* 1.0, que tenía únicamente una forma básica para soportar monedas digitales y validar sus transacciones, al *Blockchain* 2.0, con la aparición de los Contratos Inteligentes (Cassano, 2014), que permite la creación de aplicaciones descentralizadas y el almacenamiento de su código en el libro contable para ejecutar funciones mediante su llamado por medio de transacciones o paso de mensajes. El *Blockchain* 3.0, hace referencia al uso dado a los Contratos Inteligentes para administrar procesos de interacción humana o máquina-hombre, que pueden ir desde la administración de un sistema de votación u otros sistemas que garantizan la participación de todos los integrantes de una comunidad de forma libre y transparente, o la administración de recursos prevaleciendo el interés del grupo participante y evitando actividades egoístas de algún miembro de dicho grupo.

Andreas Antonopoulos (2014), explica en su libro "Mastering Bitcoin" la manera como se almacenan las transacciones en el Blockchain:

Blockchain es una estructura de datos ordenada en la que cada bloque de transacciones se enlaza primero al bloque anterior y luego se guarda. Esta lista es un árbol binario de *hashes* (códigos encriptados).

El primer bloque recibe el nombre de Génesis y corresponde a la primera transacción de esta estructura. Esto significa que es el único que no tiene ninguna referencia a bloques anteriores. Posteriormente a la generación y guardado de este bloque génesis, los siguientes bloques se guardan siguiendo una estructura hacia arriba de un árbol de Merkle (ver Figura 3), es decir, el bloque génesis se convierte en el bloque Merkle raíz; y si al ir incorporando nuevos bloques, la estructura contiene un número impar de bloques, ésta se completa a sí misma duplicando el último bloque temporalmente. En el caso de existir únicamente una transacción y al agregar un nuevo bloque, el duplicado es reemplazado por el nuevo bloque y así sucesivamente.

Los principales métodos de encriptación son algoritmos con base en el SHA (Secure Hash Algorithm), como el SHA-256 (utilizado por Bitcoin) y el SHA-3 (utilizado por Ethereum), pero también existe el Scrypt que ya está siendo utilizado con fuerza por algunas criptomonedas emergentes como el Litecoin (2018).

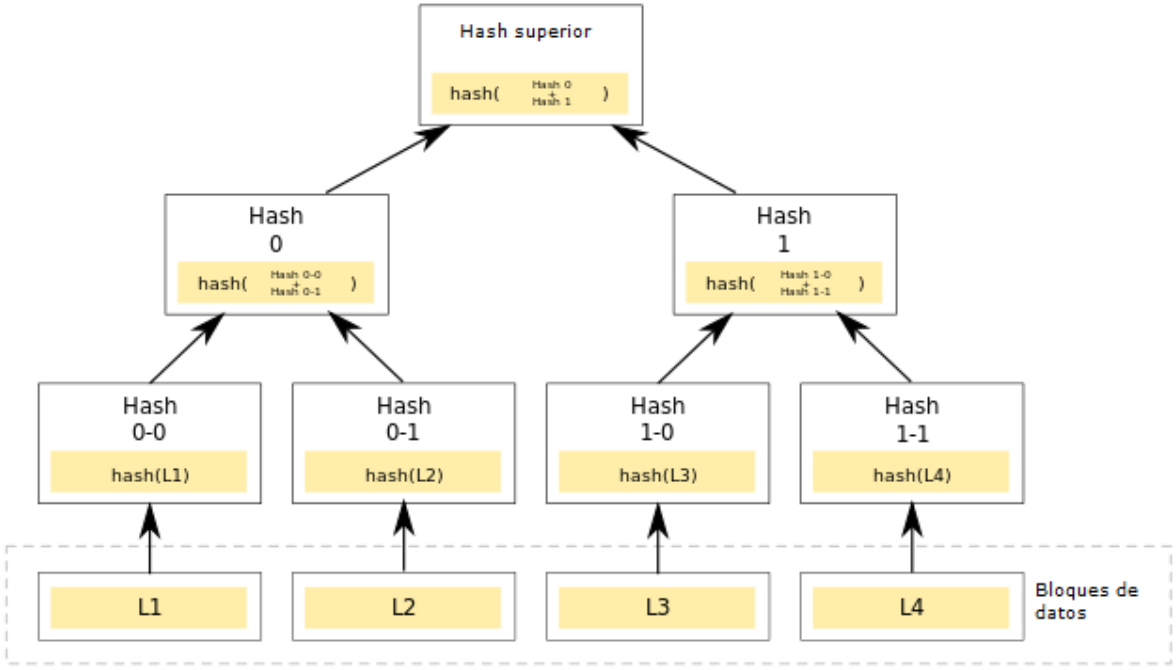


Figura 3. Estructura de un árbol de Merkle

Fuente: De Azaghal - https://commons.wikimedia.org/w/index.php?curid=18157888

Tipos

Existen diferentes tipos de *blockchain* definidos por la manera como se autorizan los validadores de las transacciones según Mattila (2016). Estos tipos son los siguientes:

1. Aquellos con autorización (*Permissioned*): en los que no hace falta incentivar la cooperación en la validación utilizando recompensas, es decir los validadores pueden realizar esta tarea de manera gratuita. Los beneficios más claros de esta arquitectura es que su funcionamiento es más rápido, gestiona la energía más eficientemente y es fácilmente escalable, ya que no implica esfuerzo en hardware necesario para conseguir recompensas que es lo que genera una mayor ralentización y consumo de energía. Por ejemplo, los *blockchain* de ERIS (Eris, 2016) e Hyperledger (Cachin, 2016) son de este tipo.

2. Aquellos sin autorización explícita (*Permissionless*): que son resistentes a la censura y a la manipulación indebida. Por ejemplo, los *blockchain* de Bitcon (Nakamoto, 2005) y Ethereum (Buterin, 2015) son de este tipo.

Sin embargo, hay que tener en cuenta que cada tipo de *blockchain* se diseña para un determinado objetivo en el caso de las criptomonedas o para uso genérico. Por ejemplo, en la categoría de propósitos generales estarían los *blockchain* de Ethereum y Eris, y en la de propósitos especiales estarían el *blockchain* de Bitcoin y el de Hyperledger.

Ethereum también define otros tipos de *blockchain* según su privacidad así:

1. *Blockchain privados*: que se utilizan en entornos controlados y donde cada nodo debe definir en su configuración inicial los otros nodos que participan en su *blockchain*. Su uso se hace en aplicaciones descentralizadas o en criptomonedas en las que solo pueden participar determinados nodos.

2. *Blockchain públicos*: que se utilizan normalmente para crear nuevas criptomonedas y permiten que nuevos nodos se puedan integrar al mismo sin mayor complejidad, es decir, no hay restricción de acceso al *blockchain*.

Por otra parte, según lo que se ha podido observar en la presente investigación, también es posible definir otros tipos de *blockchain* de acuerdo con la forma de almacenar los bloques validados. La más conocida y común es mediante los árboles de Merkle, pero también existe el Tangle (Popov, 2017), que consiste en que los bloques se guardan referenciando a dos transacciones previamente validadas y no necesariamente se sigue un árbol binario pero siempre debe haber una ruta entre el primer bloque guardado o bloque génesis y cualquiera de los últimos bloques validados (ver Figura 4).

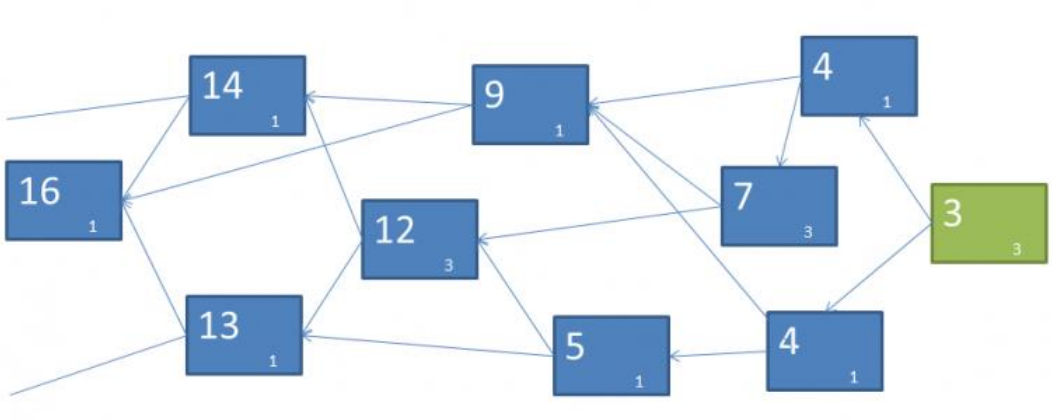


Figura 4. Estructura de un Tangle

Fuente: https://iotfutura.com/arquitecturas-iot/iota-the-tangle

Bitcoin

Fue la primera versión de criptomoneda existente en el mundo, por tanto, su *blockchain* hace parte de la primera generación es decir, el *blockchain* V1.0. Se creó con el objetivo de disminuir los costos económicos, de tiempo y esfuerzo requeridos para poder realizar transacciones financieras. En consecuencia, a partir del Bitcoin se define una moneda electrónica como una cadena de firmas digitales en la que al transferirse la propiedad de una cierta cantidad de moneda se realiza una transacción que es firmada por una función criptográfica, la cual da como justificante un código *hash* como prueba de dicha transacción. Esto quiere decir que esta moneda transfiere la confianza implícita entre las partes a una función criptográfica para justificar que una transacción se realiza correctamente.

Para realizar una transacción con Bitcoin se requieren los siguientes componentes (Nakamoto, 2005):

1. Un servidor de *Timestamp*: que se encarga de obtener el *hash* de un bloque y publicarlo en la red de nodos como prueba de su existencia en un determinado momento en el tiempo. Este *timestamp* del bloque, incluye previos *timestamp* generados por publicar otros bloques creando así una cadena de bloques.

2. El protocolo Proof of Work: que es la acción necesaria para que el servidor de *timestamp* funcione correctamente. Ejecuta el método criptográfico SHA-256 que genera una cantidad de ceros al inicio del *hash* resultante. La cantidad de ceros viene definida como requerimiento del método de encriptación (también recibe el nombre de *challenge*).

3. La red de Bitcoin que se genera teniendo en cuenta los siguientes pasos:

a. Una nueva transacción es publicada a todos los nodos
b. Cada nodo agrupa nuevas transacciones en un bloque.
c. Cada nodo debe realizar un esfuerzo para encontrar la dificultad definida por el Proof of Work para ese Bloque.
d. Cuando el nodo encuentra la dificultad del PoW, publica el nodo a toda la red
e. Los nodos aceptan el bloque si todas las transacciones incluidas son válidas.
f. Los nodos aceptan el bloque al comenzar a trabajar en el siguiente bloque.
g. Los nodos siempre toman la cadena más larga como la correcta.

4. Un incentivo: que es la recompensa que se le da al dueño de la primera transacción validada del bloque. Se compensa el esfuerzo realizado por la validación y se motiva a realizar un trabajo honesto (sólo se recibe si hay un esfuerzo en pro del buen funcionamiento del sistema).

5. Un reclamo de espacio libre en disco: las transacciones son guardadas en el blockchain siguiendo la estructura de un árbol de Merkle, por lo que para evitar que se consuma mucho espacio en disco se van compactando los bloques antiguos a medida que se van guardando los nuevos.

6. La Privacidad: el sistema tradicional bancario asegura la privacidad limitando el acceso a la información de las acciones realizadas únicamente entre las partes involucradas y un tercero de confianza (organismo de control), pero si en algún momento debe publicar determinados movimientos, cualquier persona puede enterarse de los mismos. La privacidad en el blockchain es protegida reemplazando la identidad de las partes involucradas por *hashes*, por tanto, en cualquier momento se pueden ver los movimientos realizados sin develar las identidades de las partes involucradas.

Ethereum

Se creó con el objetivo de mejorar la implementación de meta-protocolos en el blockchain. Permite la creación de aplicaciones basadas en consenso (descentralizadas) que ofrecen escalabilidad, estandarización, facilidad de desarrollo e interoperabilidad, sin aumentar la complejidad.

Se concibió como una versión mejorada del Bitcoin, con el fin de superar las limitaciones de su lenguaje de programación, proporcionando características avanzadas tales como protección del *blockchain*, límites de retiro, contratos financieros, mercado de juegos de azar y similares a través de un lenguaje de programación generalizado (Criptonoticias, s.f.).

Ethereum se basa en un *blockchain* que permite mediante el lenguaje de programación Solidity (Solidity, s.f.), resolver cualquier problema computacional añadiendo reglas complejas. Para este caso específico, las aplicaciones son a manera de Contratos Inteligentes o *Smart Contracts* (Delmolino, Arnett, Kosba, Miller y Shi, 2015), que son programas que ejecutan acuerdos establecidos entre dos o más partes, cuando se da una condición prevista con anterioridad. Es decir, son contratos que se ejecutan y se hacen cumplir a sí mismos de manera automática y autónoma.

El *blockchain* de Ethereum es de segunda generación (*blockchain* V2) y a diferencia del *blockchain* de Bitcoin, almacena en sus bloques un número identificativo de los mismos, su estado más reciente y la dificultad actual para obtener un nuevo bloque.

Los componentes de Ethereum son (Buterin, 2015):

1. Cuentas: identificadas con una dirección de 20-bytes de longitud que contiene las transiciones de estado con otras cuentas. Utiliza las siguientes variables:

Nonce: que es el contador que asegura que cada transacción se procese una sola vez

Balance: que es el total de *ethers* que posee la cuenta en el momento

Smart contracts: que son los contratos inteligentes creados por el usuario (si existe alguno)

2. Almacenamiento: vacío por defecto

3. Ether: es la moneda interna utilizada para pagar las transacciones.

4. Transacciones: son acciones realizadas por cuentas externas (relacionadas con las personas) que pueden contener datos para actualizar un estado de un contrato o una cantidad de moneda específica. Normalmente conlleva un costo el poder realizarlas.

5. Mensajes: son acciones que pueden ser ejecutadas por cuentas externas o contratos y pueden contener datos. Normalmente se utilizan para comunicación o consultas de información contenida en contratos, en muchos casos no conllevan costo.

Hyperledger

Es una plataforma *blockchain* tolerante a fallos, creada para la industria bajo licencia *Open Source*. Permite también ejecutar contratos inteligentes pero a diferencia del Ethereum, estos están programados en el lenguaje Go (Golang, s.f.) y cada nodo es ejecutado mediante un contenedor llamado *Docker*.

Utiliza el consenso Bizantino (Sousa, Bessani y Vukolic, 2018) que consiste en un sistema que evita fallos en la validación de transacciones mediante un modelo de replicado de una máquina de estado (MRME). Allí, cada nodo ejecuta los mismos procesos y en caso que alguno presente fallo, puede ser corregido por otras réplicas.

Fundamentalmente se puede interactuar con su *blockchain* mediante 3 tipos de transacciones (Cachin, 2016):

1. Transacción de despliegue: que se utiliza cuando se tiene construido un contrato inteligente mediante el lenguaje Go y se desea que esté disponible para ser usado en cualquier momento, por tanto, este código se instala en los nodos.

2. Transacción de invocación: que utiliza parte del código de una transacción de despliegue, por lo que puede modificar su estado. Este tipo de transacción únicamente notifica si ha fallado o ha sido ejecutada con éxito.

3. Transacción de consulta: que se utiliza para solicitar parte de la información del estado de alguno de los contratos guardados anteriormente.

Hyperledger también se basa en la asignación de certificados para poder realizar diferentes acciones sobre la red de nodos y así evitar algunos tipos de ataques. Por ejemplo, si un nodo quiere adherirse a la red, realiza una petición, si ésta se acepta, se le asigna un certificado de registro y finalmente es incorporado a la red. Si se quiere realizar una transacción, se debe explicar su tipo y al obtener el certificado de la misma, ésta se puede llevar a cabo.

Algoritmos de consenso

Los algoritmos de consenso son una parte vital de la tecnología *blockchain* y se implementan para asegurar que un grupo de personas acepte que todas las transacciones realizadas sean válidas y auténticas. Los mecanismos utilizados aseguran las transacciones para evitar ataques o problemas como el del *doble gasto*, que es algo muy parecido a enviar un activo digital a alguien cuando no se utiliza el original sino una copia y resultan dos personas obteniendo el mismo activo.

Dado que las criptomonedas están descentralizadas, la red de cada una de ellas debe encontrar su propia manera de verificar todas las transacciones en su cadena de bloques y prevenir el fraude. Los mecanismos de consenso garantizan que todas las transacciones en el Blockchain sean auténticas y que las monedas pasen de la billetera A a la billetera B con transparencia y seguridad. Estos algoritmos igualmente realizan el proceso de compensar el esfuerzo llevado a cabo por las personas que verifican dichas transacciones (Zwanenburg, 2018).

Cada criptomoneda tiene su propio algoritmo de consenso escrito en su código. Este algoritmo puede ser uno que ya exista, como el del Bitcoin o Ethereum, o, uno nuevo como el del IOTA, entre otros. En la presente investigación por ejemplo, se escribe un nuevo algoritmo de éstos como producto de la combinación de conceptos de dos algoritmos existentes. A continuación se describe el funcionamiento de algunos de los algoritmos de consenso más conocidos.

Proof of Work (PoW)

Este protocolo se desarrolló para impedir ataques de denegación de servicio y abusos como el *spam* en una red. Estos casos, al requerir algo de trabajo del solicitante del servicio, generalmente significaban incremento del tiempo de procesamiento en un computador (Dwork & Naor, 1993). El Bitcoin fue la primera criptomoneda que implementó este protocolo como algoritmo de consenso para validar transacciones con criptomonedas. La validación se realiza ejecutando algoritmos matemáticos complejos como el SHA-256 y se autoregula a partir de la velocidad en que se validan las transacciones, es decir, a mayor velocidad de validación, mayor será el nivel de dificultad incluido en la misma para alcanzar el código *hash* que permitirá crear un bloque en el blockchain (Nakamoto, 2005).

En el blockchain, cada bloque contiene el *hash* del bloque anterior y esto es lo que lo protege de manipulación ya que intentar cambiar alguno, implicaría regenerar todos los sucesores y rehacer el trabajo que contienen y esto sería prácticamente imposible porque sólo podría hacerse creando un nuevo bloque que contenga el mismo predecesor y por la velocidad con que se generan los bloques, el predecesor en un momento dado sería diferente al que se estaría intentando regenerar (BitcoinWiki, 2015).

Proof of Stake (PoS)

Este protocolo, al igual que la mayoría, ha surgido con el fin de corregir los problemas del alto consumo de energía utilizado por el PoW para validar transacciones. Permite la incorporación de validadores al proceso, dependiendo de la cantidad de ahorros o moneda que tengan en sus billeteras, es decir, que si se mantiene una cierta cantidad de moneda en la billetera durante un mínimo de tiempo, se puede ser validador. Sin embargo, con esta estrategia pueden surgir entre otros, los siguientes problemas:

1. Promoción: se promueve la acumulación de la moneda y, por tanto, la velocidad de uso de la misma disminuye.

2. Distribución: como las recompensas van para los validadores que tienen cierta cantidad de moneda acumulada, no está clara la manera de distribuir las monedas inicialmente.

3. Monopolización: aquellos con una cantidad significativa de monedas cosecharían la mayoría de las monedas futuras.

4. El ataque del 51%: igual que con el PoW cuando un validador tiene una buena capacidad de potencia de cálculo y puede acaparar el 51% de las validaciones. En PoS sucede cuando un validador tiene un 51% de peso de moneda.

El PoS implica un importante ahorro en el consumo de energía, y una mejor alineación de los incentivos entre los validadores y las partes interesadas (porque los validadores son ahora las partes interesadas) (Ren, 2014).

Con base en lo anterior, la evolución de PoS puede ser entendida por cada moneda que intenta resolver los problemas mencionados a su manera como es el caso del Peercoin (2015) que realiza actividades de validación híbridas combinando PoW y PoS; el Blackcoin (2015) que utiliza un PoS puro y mejorado, y el Ether (2018) que intenta implementar un PoS con tolerancia a fallos bizantinos.

Según King y Nadal (2012), el PoS requiere que se tenga almacenada la cantidad de moneda adquirida durante un mes para poder ser utilizada en las validaciones. Para evitar que se guarde más tiempo de lo indicado, han implementado una tasa de oxidación de 1 céntimo por unidad de moneda consumida.

Proof of Stake on Velocity (PoSV)

Este protocolo comparte la misma base del PoS con la diferencia de un menor tiempo en propiedad exigida de moneda para poder validar, por lo tanto, la velocidad de uso aumenta ya que los validadores no necesitan mantenerla almacenada por tanto tiempo como en su predecesor.

Según Ren (2014), la mayoría de los inconvenientes de PoW y PoS no se deben a fallas en los diseños técnicos sino a la desconexión que tienen de los aspectos económicos y sociales de una moneda real. Afirma igualmente, que muchas de las criptomonedas se crean como productos tecnológicos y no como monedas en todo el sentido de la palabra.

PoSV está diseñado para fomentar tanto la propiedad de la moneda como la actividad para su uso (velocidad), los dos criterios principales de una moneda social. El protocolo se desarrolló específicamente para la moneda social digital Reddcoin (2015) que es accesible al público en general; su filosofía es integrarse fácilmente a las redes sociales para que el envío y recepción de consejos y micro-transacciones sean baratos, rápidos y gratificantes para todos.

Proof of Authority (PoA)

En Marzo de 2017, un grupo de desarrolladores liderados por Gavin Wood (Wood, s.f.), uno de los fundadores de Ethereum, propuso este algoritmo para un *blockchain* basado en Ethereum con el fin de solucionar el problema de los ataques *spam* en la red de prueba *Ropsten* de Ethereum.

El protocolo PoA es un protocolo *Proof of Stake* optimizado que aprovecha la identidad como herramienta para validar. Esta identidad es dada por un grupo de validadores denominados autoridades (*Authorities*) que han sido aprobados previamente para validar transacciones y bloques dentro de la red respectiva. Se plantea que este grupo no sea mayor de 25 validadores para garantizar la eficiencia y la seguridad de la red. Inicialmente, el primer rol lo adquiere el creador de la cadena, el cual define los primeros *Authorities* a participar y éstos, votando por otros participantes, podrán permitirles incorporarse como validadores (Curran, 2018).

Las características principales de una red de PoA son:

1. Bajo requerimiento de potencia de cómputo
2. Ningún requisito de comunicación entre nodos para llegar a un consenso
3. Continuidad de la red independiente del número de nodos genuinos disponibles (ya que están aprobados previamente a través de una verificación cruzada en el dominio público).

PoA está diseñado para ser menos intensivo en computación que los modelos de PoW y además, elimina la preocupación principal del modelo PoS referente a que aunque las apuestas para validar entre dos partes puedan ser iguales, su valor para cada parte puede variar significativamente dependiendo de sus tenencias (la cantidad de moneda que se tenga en posesión en un momento dado).

Existen 3 requisitos básicos para convertirse en un validador PoA, con la idea de estructuraar incentivos para impulsar un comportamiento honesto. Estos son:

1. Sus identidades deben confirmarse formalmente en el Blockchain con la capacidad de realizar referencias cruzadas de estas identidades a través de datos confiables disponibles en el dominio público (como una base de datos de notarios públicos).

2. La elegibilidad para convertirse en un validador debe ser difícil de obtener para garantizar que la posición prospectiva a largo plazo del validador sea un claro incentivo, tanto financiero como de reputación, para seguir siendo un validador honesto.

3. Debe existir una completa uniformidad en el proceso para el establecimiento de validadores. Hay algunas plataformas que implementan variaciones ligeramente diferentes de los requisitos anteriores que se centran en proporcionar un incentivo financiero para que el validador permanezca como parte de la red a largo plazo, y la reputación, como el "desincentivo" para actuar erróneamente. Cualquier validador que actúe de manera maliciosa puede eliminarse fácilmente del proceso de validación y reemplazarse. El resultado final para ese validador sería un golpe público a su reputación, así como una pérdida de ganancias financieras futuras. El uso de la reputación a través de la identidad es especialmente relevante para los tiempos contemporáneos.

Al mismo tiempo, el caso de uso de PoA se considera en gran medida como el más efectivo para las cadenas de bloques autorizadas (privadas). Por ejemplo, una red de bancos verificables donde cada uno actúe como su propio validador. Se necesita una mayoría para confirmar el estado de la cadena de bloques y ofrecer una mayor eficiencia en la verificación de la transacción y el consenso, sin tener que descartar una cantidad sustancial de influencia, privacidad o poder en el proceso (Curran, 2018).

Esta tesis hace uso de este protocolo de consenso para generar el suyo propio y resolver el problema de investigación planteado.

Proof of Space

Este protocolo es una variación del protocolo *Proof of Stake*, pero el recurso base con el cual se valida es el espacio en disco disponible. La validación de transacciones se realiza en dos fases: la fase de inicialización, que permite llenar una cantidad mínima de datos en el disco duro y la fase de ejecución, que lleva a cabo una validación PoS que al finalizar determina si fue exitosa o no según la cantidad de espacio en disco ocupado, es decir, una transacción fallida ocupará 8 veces más de disco que una transacción realizada de manera exitosa (Dziembowski, Faust, Kolmogorov y Pietrzak, 2015).

Este tipo de validación puede significarle a los validadores una menor dificultad en el procesamiento de las transacciones, pero a su vez dar origen a los siguientes problemas:

1. Generación de una cantidad importante de datos sin sentido para justificar su funcionamiento
2. Exposición a daños en el hardware de almacenamiento, por lo que un ataque a éste, podría generar no solo un bloqueo a las validaciones sino también daño a la información sensible guardada en dichos dispositivos.

Proof of Activity

Este protocolo es una extensión del *Proof of Work* pero reduciendo el consumo de energía involucrado en la validación de transacciones. Los nodos de una red *Proof of Activity* deben realizar verificaciones más complejas que las llevadas a cabo por los nodos de *Proof of Work* para crear bloques en el Blockchain y los desarrolladores argumentan que dicho trabajo extra trae consigo algunos beneficios particulares (Bentov, Lee, Mizrahi y Rosenfeld, 2014).

La subrutina principal que incorpora *Proof of activity* se llama *follow-the-satoshi*, mediante la cual se transforman en un *Satoshi* (unidad más pequeña de la criptomoneda), algunos valores pseudoaleatorios generados que se seleccionan uniformemente entre todos los *Satoshis* que han sido acuñados hasta el momento. Para ello, se elige un índice pseudoaleatorio entre cero y el número total de *Satoshis* existentes hasta el último bloque, para inspeccionar el bloque en el que se encuentra el *Satoshi* acuñado y cada transacción que transfirió este *Satoshi* a una dirección posterior hasta llegar a la dirección que controla actualmente dicho Satoshi. Este proceso se puede considerar como la selección de un actor pseudoaleatorio de manera uniforme que estará involucrado en la validación de las transacciones.

El protocolo *Proof of Activity* recompensa a los interesados que participan y mantienen la red, en lugar de castigar a grupos de interés que no participan.

Proof of Existence

Este protocolo es diferente a los antes mencionados y a la filosofía de un algoritmo de consenso como los comúnmente utilizados para validar criptomonedas. Funciona como un servicio en línea que verifica la existencia de archivos en un momento específico mediante transacciones con marca de tiempo en el blockchain de Bitcoin. Fue lanzado en 2013 como un proyecto de código abierto.

Esencialmente, es un servicio público notarial en Internet, una forma económica de utilizar el poder de cómputo distribuido de Bitcoin para permitir que las personas verifiquen que un documento existió en un determinado momento. Los algoritmos se pueden utilizar para crear un compendio o una cadena criptográfica que sea representativa de un dato. El resumen creado por una función *hash* se basa en las características de un documento. No hay dos compendios iguales a menos que los datos utilizados para calcular los compendios sean los mismos (Kirk, 2013).

Según Trew, Brandon y Dorier (s.f), ya se utilizan de manera informal, soluciones mediadas por *blockchain* para representar un tipo de *copyright* digital. Los usuarios generan un *hash* (que es un resumen criptográfico de un archivo que sirve como una huella digital única) y lo cargan en el Blockchain. Debido a que cada bloque tiene marca de tiempo y cada *hash* representa un archivo específico (cambiar incluso un solo carácter de un documento o un píxel de una imagen da como resultado un *hash* completamente diferente). Esto prueba más allá de cualquier grado razonable de duda, que el archivo existió en el momento en que se cargó el *hash*. Este es un enfoque que puede extenderse desde derechos de autor y patentes a contratos de todo tipo, e incluso a establecer la condición física de una propiedad o un automóvil de alquiler antes de que sea utilizado por un cliente.

Ataques

La implementación de la tecnología *blockchain* en las criptomonedas hace casi imposible el éxito de un ataque directo utilizando métodos de "desencriptado", porque puede llevar mucho tiempo hacerlo, ya que en el momento en que se tenga la información legible y se quiera continuar la cadena, se habrán realizado muchas transacciones más sobre la cadena principal y se necesitaría volver a comenzar el ataque, lo cual no quiere decir que se deje de intentarlo pues estos ataques han evolucionado también y se aprovechan de la misma tecnología existente para funcionar.

Existen igualmente ataques a las billeteras y las que están en línea son las menos seguras porque se exponen a la denegación del servicio (DDoS) que es un ataque enfocado hacia la conexión que tiene un nodo con respecto al resto (Vyas and Lunagaria, 2014).

El doble gasto de los *tokens* de diferentes monedas, muchas veces se logra alterando el *timestamp* del validador. Esto quiere decir que modificando la fecha del dispositivo se puede lograr que el sistema piense que todavía tiene recursos que previamente se habían gastado, este ha sido otro tipo de ataque.

El PoW al ser el primer sistema de validación y el más usado en criptomonedas, es el objetivo de muchos ataques, por lo que otros sistemas de validación comenzaron a ser también importantes no solo para disminuir el esfuerzo necesario para validar sino para mitigar dichos ataques.

El ataque más conocido para PoW consiste en que si un validador o conjunto de validadores posee el 51% o más de la potencia de validación puede poner en peligro el funcionamiento de la criptomoneda tomando decisiones sobre las transacciones que se estén validando o se vayan a validar, por ejemplo, se puede decidir sobre cual transacción validar o no, realizar cambios sobre ellas o simplemente reversar validaciones que ya hayan sido aceptadas. Por otra parte, cualquier problema técnico o ataque a este validador puede "tumbar" todo el sistema.

King y Nadal (2012) muestran que el *Proof of Stake* propone un sistema basado en la validación de transacciones a partir de una cierta cantidad de tiempo en propiedad de una cantidad de moneda. Este tipo de validación transfiere el problema de tener potencia de cómputo para validar, al de tener más tiempo en propiedad de cierta cantidad de moneda, permitiendo la acumulación de la misma y disminuyendo su velocidad de uso. Esto quiere decir que no solo puede salir más caro el ataque por la cantidad de moneda necesaria para realizarlo sino que también el ataque puede afectar al mismo atacante.

Li, Andreina, Bohli y Karame (2017) mencionan que existe un tipo de ataque que afecta al PoS llamado "*Nothing at Stake"*, en el cual se intenta generar que haya bloques en conflicto para forzar la creación de varias separaciones de las cadenas de bloques (o fork) y poder recibir más incentivos por validar en dichos fork. La principal consecuencia de este ataque, es que ralentiza el sistema y aprovecha la debilidad del *blockchain* de solucionar el doble gasto de una misma cantidad de moneda.

El *long range* es un ataque del *Proof of Stake* hacia el historial del *blockchain* para poder aprovechar el doble gasto de una misma cantidad de moneda. Este ataque en teoría, se realiza, cuando se tiene la mayoría de *tokens* en la red, pero realmente lo que aprovecha es que si en algún momento se ha tenido en propiedad más del 30% de la cantidad de moneda circulante, el atacante podría crear un fork tomando como base algún bloque creado anteriormente. En resumen, el atacante aprovecha cuentas viejas que en ese momento tengan balance 0 (teniendo en cuenta que éstas no están tan protegidas como las que tienen balance > 0) para ejecutar sus acciones.

El *Proof of Authority* a pesar de ser un sistema bastante más ligero que el PoW no quiere decir que sea más seguro, pues para ser validador con este método, se debe ser aceptado por otros validadores y teniendo en cuenta que para ingresar se tiene que estar plenamente identificado por un *hash* o address validado por el *blockchain* de la moneda que lo implemente, debería ser difícil ser aceptado como validador, pero, a pesar de ello, se puede correr el riesgo de atacar el método para ser elegido.

3. MARCO METODOLÓGICO

3.1 Definición del problema

El problema a resolver en esta investigación se enuncia de la siguiente manera:

¿Lo*s nuevos diseños de monedas complementarias, teniendo en cuenta el desarrollo de las redes sociales y las tecnologías blockchain y de contratos inteligentes, cómo pueden mejorar los negocios?*

Y su solución se lleva a cabo mediante un diseño experimental cuantitativo con razonamiento deductivo. En el desarrollo se tienen en cuenta las siguientes concepciones:

1. En los últimos treinta años, se han desarrollado más de 4000 sistemas monetarios complementarios en más de 50 países alrededor del mundo con una variedad de modelos y esquemas de aplicación. Desafortunadamente, debido a las numerosas barreras y limitaciones identificadas y estudiadas por el proyecto VirCoin2SME, un gran porcentaje de estos sistemas ha fracasado en su intento de crear prácticas alternativas que apunten a apoyar las economías hacia ecosistemas sostenibles mientras mejoran las interacciones sociales entre los diferentes actores involucrados.

2. En la era de la disrupción del *blockchain*, de los nuevos avances tecnológicos en redes sociales y del dinero social, las monedas sociales virtuales-complementarias esperan alcanzar los más altos niveles no solo de desarrollo e innovación, sino también de usabilidad, seguridad y confianza. Sin embargo, antes de cumplir con este gran salto exitoso, es necesario comprender y enfrentar las barreras para su adopción y replicación, y explorar las posibilidades de superarlas y mejorar el desempeño de estas monedas cumpliendo con los objetivos, características y mecanismos que permitan crear en el largo plazo modelos de negocio que impulsen el desarrollo socio-económico dentro de un contexto local, regional o global.

3. La existencia de numerosas tecnologías y métodos de validación de criptomonedas, lo cual dificulta la definición de una equivalencia entre unas y otras, y por tanto imposibilita su intercambio. Es importante el aprovechamiento de esa diversidad, para generar soluciones lo suficientemente robustas en términos de seguridad y rapidez que permitan cierto nivel de personalización para generar nuevos métodos de validación implementables sobre una tecnología común.

4. Las tecnologías basadas en *blockchain*, específicamente el *Ethereum* que permite diseñar nuevas aplicaciones mediante su propio lenguaje de programación y probar nuevos métodos de validación sobre éstas de manera abierta y libre.

5. En términos de protocolos de consenso por ejemplo, el problema que tiene el tradicional método *Proof of Work* es que a medida que se incremente la cantidad de validaciones a realizar y se requiera más velocidad, se aumenta la dificultad y hace que no cualquier usuario pueda participar en el proceso. Esto puede dar origen a que después de cierto tiempo, el poder de mercado que tienen los bancos sobre las monedas de curso legal, se traslade a los validadores de criptomonedas con suficiente potencia de validación.

6. La implementación de otros sistemas de validación independientes, puede que solucione determinados problemas, pero al final, afecta la velocidad de uso de la moneda o facilita la generación de otros ataques. Por tanto, el autor de esta tesis piensa que un método de validación híbrido (resultado de la combinación de dos conceptos existentes) podría aprovechar las bondades de los protocolos participantes mitigando entre sí, los posibles ataques que puedan surgir entre ellos, y realizar eficientemente la tarea para la cual fue concebido, esto es, motivar el uso y apropiación de las criptomonedas en los negocios, reduciendo las barreras para su adopción (por ejemplo: confianza, baja liquidez, poca oferta, costo incurrido en la validación, esfuerzo para su mantenimiento y viralización, y gestión centralizada entre otras) e impulsar con ello el desarrollo socio-económico en contextos locales, regionales o globales.

7. El método del estudio de caso utilizado para conocer el funcionamiento de la moneda digital RES y la moneda virtual Eurakos (durante la participación del autor de esta tesis en el proyecto Vircoin2SME), permitió adquirir conocimiento para orientar el diseño de la criptomoneda EurakosNext que aprovecha los beneficios de la tecnología *blockchain* y ofrece la oportunidad de innovar en los procesos de validación de transacciones, con el fin de motivar su adopción en contextos económicos locales de micronegociaciones, de manera segura, fácil y sostenible reduciendo las barreras relacionadas con el factor humano (emociones y confianza), la gestión de la moneda por parte de las PYMES y los consumidores y el contexto en donde se utilizan especialmente cuando el acceso a la tecnología está garantizado a la población.

8. El círculo de la ingeniería de la usabilidad (analizar contexto, conceptualizar, diseñar la solución, probarla y ajustarla para liberarla como final o para retroalimentar el ciclo), que se enfoca en orientar el procedimiento para generar soluciones software con: facilidad de uso (múltiples formas de intercambiar información entre el usuario y el sistema) y, facilidad de aprendizaje para nuevos usuarios que garantizan interacción efectiva, máximas prestaciones y la satisfacción del usuario incluyendo el soporte al mismo para el alcance de las metas (Rosson y Carroll, 2001).

3.2 Objetivos

3.2.1 Objetivo general

Diseñar un protocolo de consenso que inspire seguridad y confianza en los usuarios de criptomonedas para el fomento de su uso y apropiación, y que impulse el desarrollo socio-económico en contextos locales, nacionales e internacionales.

3.2.2 Objetivos específicos

1. Definir el contexto económico para la creación de iniciativas sostenibles basadas en el uso de monedas sociales-complementarias.
2. Identificar las barreras de adopción de monedas sociales-complementarias y posibles soluciones para su reducción desde el punto de vista de los usuarios.
3. Analizar desarrollos tecnológicos existentes en torno al concepto, estructura y uso de criptomonedas, para la identificación de variables que permitan soluciones que amplíen su rango de uso de manera directa, natural y segura en comunidades locales, nacionales e internacionales.
4. Diseñar la criptomoneda EurakosNext y el protocolo de consenso *Proof of Reputation* (PoR) para el fomento, uso y adopción de criptomonedas en los negocios, que minimicen algunas de las barreras identificadas en 2) en contextos económicos locales, regionales y nacionales.

4. DISEÑO DE LA INVESTIGACIÓN

4.1 Introducción

Para llevar a cabo un proceso de innovación en la manera como se validan las transacciones con criptomonedas y para fomentar su uso de manera natural y segura en contextos económicos cotidianos, se presenta a continuación la propuesta de dos productos importantes generados en esta tesis como son: el diseño de la criptomoneda EurakosNext como una evolución de la moneda virtual Eurakos introducida en la sección 2.2.4, y, el diseño del protocolo de consenso *Proof of Reputation (PoR)* como una combinación de conceptos *del Proof of Work* y el *Proof of Authority*. El uso del *Proof of Reputation (PoR)* permite reducir las barreras de adopción de las criptomonedas en aspectos relacionados con la velocidad de la validación de sus transacciones, el esfuerzo para su mantenimiento y viralización y, la centralización de su gestión, entre otras.

4.2 Criptomoneda EurakosNext

4.2.1 Descripción

EurakosNext es una de las contribuciones de esta tesis doctoral y está orientada a permitir que cualquier integrante de su red, desde su rol, administre su propia moneda de manera rápida, segura, confiable y entretenida. Se parte del modelo de comportamiento de la moneda virtual Eurakos (expuesto en la Figura 2) y evita depender de una organización central para gestionar la confirmación de las transacciones o la aceptación de los participantes en dicho proceso.

Esta moneda permite hacer acuerdos entre dos participantes para realizar negociaciones que ayuden a los clientes a obtener más productos con la misma cantidad de dinero y motivar a los proveedores a obtener más clientes. Funciona mediante el uso de *contratos inteligentes* a través de la tecnología *blockchain* de Ethereum.

El diseño de un contrato EurakosNext tiene en cuenta los siguientes elementos:

a. Los participantes y sus roles
b. Las funciones de cada rol
c. Cláusulas para validar la participación de los grupos de interés
d. Cláusulas a ser ejecutadas cuando el contrato termina y alguien no cumple con algún compromiso.
e. Cláusulas que permiten el uso del mediador si alguna de las partes interesadas no está de acuerdo con alguna acción ejecutada.
f. Cláusulas para activar y definir la mediación y las acciones permitidas en este proceso.
g. Fecha de expiración del contrato o tiempo de duración del mismo.
h. Presupuesto de la cesta de productos
i. Contabilidad asociada a un contrato

El ecosistema de la moneda, implementa un concepto similar al de *Ethereum* con la diferencia de que las acciones o eventos están pre-establecidos. La inclusión de más acciones y eventos se lleva a cabo de acuerdo con las interacciones de los usuarios (consumidores y proveedores).

La tecnología *blockchain* se utiliza, al igual que en muchas criptomonedas, para almacenar y validar transacciones y contratos con un criterio específico y un método de encriptación particular. Esta investigación introduce nuevos criterios para realizar tareas que tienen como objetivo, mejorar el procesamiento y el tiempo de respuesta y, facilitar la participación de los usuarios en los procesos de validación. En la sección 4.3 se introduce el algoritmo de consenso desarrollado.

Actualmente, la estructura de EurakosNext (ver Figura 5) sigue el modelo cliente-servidor y tiene diferentes componentes de acuerdo con su funcionalidad y el dispositivo que debe ejecutarlos. Estas componentes son:

a. El *blockchain* que es el libro de contabilidad que permite obtener consenso entre pares sobre qué transacciones y contratos son válidos.

b. El servidor *Ethereum* propio que está constituido por:

1. Un *nodo Ethereum* que interactúa con la cadena de bloques para leer y escribir "bloques" o información relacionada con contratos, transacciones o prosumidores (proveedores y consumidores).
2. Un *módulo de procesamiento* para crear acuerdos basados en contratos que permiten que el módulo del prosumidor complete la información necesaria para almacenarlos.
3. Un *módulo de conversión* (a futuro), que tendrá los componentes necesarios para ayudar a la conversión de otras monedas virtuales a Eurakos con el fin de facilitar el trabajo en el entorno.
4. Un *túnel de servicios web* que permite la interacción con interfaces móviles relacionadas. Este componente trabaja con el módulo de procesamiento para preparar contratos y transacciones que se almacenarán o leerán en la cadena de bloques con la ayuda del nodo Ethereum.
5. Una *interfaz web* para permitir la accesibilidad en línea al entorno. Este componente interactúa con el módulo de procesamiento de la misma manera que lo hace el túnel de servicios web.
6. El módulo prosumidor, que incluye una interfaz móvil (web móvil o aplicación de cliente móvil) para obtener la información relacionada con cuentas, contratos y transacciones; luego procesa la información para enviarla de manera segura utilizando el túnel de servicios web para manejar el almacenamiento y la recuperación en el *blockchain*.

7. Las billeteras, que permiten a los usuarios conocer el estado de sus transacciones realizadas con esta moneda. Estas se pueden almacenar en cualquier dispositivo elegido por el usuario. Si se utiliza más de un dispositivo, el usuario debe estar pendiente de mantenerlos sincronizados para manejar la información actualizada.

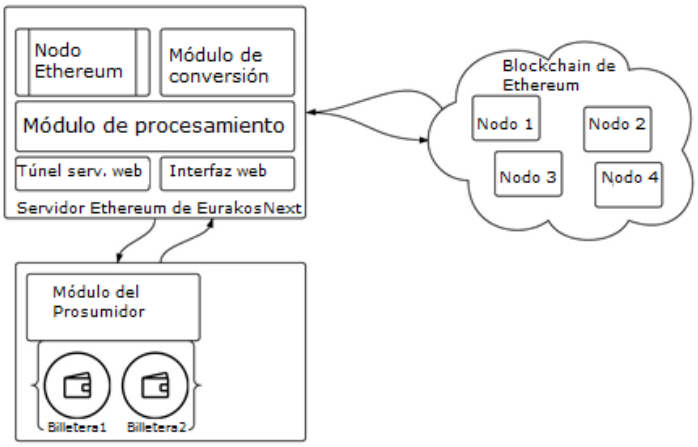


Figura 5. Estructura global de la criptomoneda EurakosNext

Fuente: el autor

4.2.2 Validación de transacciones y contratos

Las recompensas para los validadores se obtienen de las comisiones aplicadas. Al realizar una validación de transacción, realizar un acuerdo o validar una cláusula del contrato, se les paga una comisión de aproximadamente 0,00075€ (valor definido tentativamente).

La recompensa a pagar cuando se realiza una validación de una transacción, una cláusula o un contrato, se define mediante las siguientes ecuaciones:

$$Pay(Tx) = W_{Tx} * Rw \quad (1)$$

$$Pay(Cl) = W_{Cl} * Rw \quad (2)$$

$$Pay(Ctk) = W_{Ctk} * Rw \quad (3)$$

Donde:

Rw = Comisión base de la recompensa

Tx = Transacción

Cl = Cláusula

Ctk = Contrato /Acuerdo

W = Peso asignado de acuerdo con la dificultad de la información a ser validada

W_{Tx} = Peso asignado a las transacciones

W_{Cl} = Peso asignado a las cláusulas

W_{Ctk} = Peso asignado a los acuerdos

Pay(Tx) = Función que define el pago a realizar cuando se ha validado una transacción.

Pay(Cl) = Función que define el pago a realizar cuando se ha validado una cláusula.

Pay(Ctk) = Función que define el pago a realizar cuando se ha validado un contrato.

$W_{Tx} = 1$

$W_{Cl} = 2$

$W_{Ctk} = 4$

Rw = 0,00075 €

El esfuerzo realizado por los validadores se representa en el peso establecido en las fórmulas. Si en el futuro el esfuerzo está definido por una fórmula, el peso puede ser reemplazado por esa fórmula.

Utilizando *Solidity*, el lenguaje de programación para contratos inteligentes basados en Ethereum, el flujo de trabajo de la moneda EurakosNext (el mismo de la moneda virtual Eurakos presentado en la Figura 2), requeriría de las siguientes funciones que a su vez son contratos:

User.sol

Es el contrato que representa al usuario y que en resumen se puede definir con una dirección en donde se almacena su dinero, y que contendrá un saldo en Eurakos y podrá enviar y recibir determinados montos. En términos técnicos, este contrato equivale a la billetera del usuario y tiene la siguiente apariencia en código:

```
    // Variables
    address userId (hashId)
    uint balance;
    mapping(string => address) walletList;
function public setAccount(address newAccount)
function public getAccount() returns (uint)
function addWallet(string newWallet, address walletId)
function receiveAmount(uint amount)
function sendAmount(uint amount)
```

Agreement.sol

Es un contrato que maneja los acuerdos de la negociación. Su código tiene la siguiente apariencia:

```
    // Variables
    mapping (address => uint) public balances;
    // mapping includes the address of client and provider
    address consumer (hash id)
    amount of currency to spent
    string startDate
    string endTime
function getRelatedClause()
function executeClauses()
function closeContract()
```

Clause.sol

Es un contrato que contiene cada cláusula del acuerdo y está conformado por:

1. Una dirección relacionada con el contrato, en la que los usuarios ingresan el monto acordado en el contrato.
2. Una dirección del usuario (prosumidor) al que se le aplica la cláusula.
3. Un par de variables (coinAmount y productQuantity) que pueden ir vacías o no dependiendo de lo que el usuario haya prometido en el contrato

Y, otras componentes específicas para su funcionamiento.

El código tiene la siguiente apariencia:

```
        address agreeOwner;
        address userFrom;
        string productIn;
        uint dateInit;
        uint timeLong;
        uint productQuantity;
        bool isProductRequired;
        uint coinAmount;
        bool isAmountRequired;
        bool isAmountReturnedOnFail;
        bool isProductReturnedOnFail;
        bool end;
        bool success;
        function productNotRequired() {
        isProductRequired = false;
}
Function amountNotRequired() {
        isAmountRequired=false;
}
Function validateClause() {
        …
}
function executePenalty ()
{
        …
}
```

Mediation.sol

Si existiese mediación para hacer cumplir las cláusulas, el código de dicho contrato sería similar a como se muestra a continuación:

```
address contractId
address userId   // mediator
address[] transactionsToRollBack
function executePenalties()
function rollbackPenalties()
```

Coin.sol

Es un contrato que determina el funcionamiento de la moneda a implementar. En la práctica debe existir un contrato de estos por cada moneda que se quisiera utilizar en dicho contrato y manejaría dos procedimientos, el que representaría funcionalmente la moneda con el formato "NombreMoneda.sol" y el que representaría un contrato genérico con un nombre como "CoinHandler.sol" que permitiría administrar el redireccionamiento al contrato de la moneda involucrada en la negociación.

En la Figura 6 se puede observar un esquema de la estructura de estos contratos y sus relaciones.

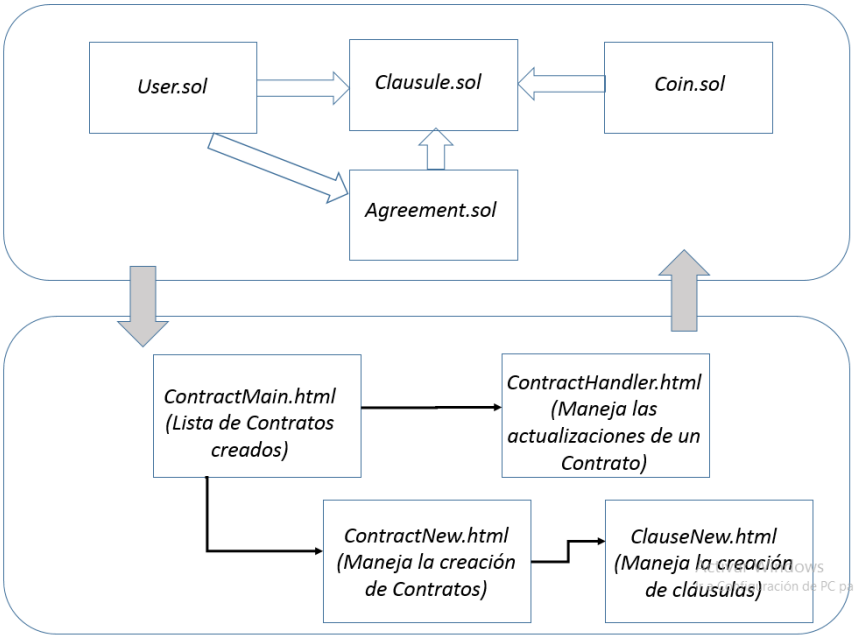


Figura 6. Estructura de contratos en *Ethereum* para la evolución de un contrato digital *Eurakos* a un Contrato Inteligente de *EurakosNext*

Fuente: el autor

La Figura 7 muestra lo que sucede en *background* cuando se acepta un contrato. De acuerdo con esto y con el flujo de operación de *Eurakos*, cuando se acepta la oferta, se inicia el proceso para guardarla utilizando el entorno Ethereum. El contrato se llena con base en la información obtenida de la negociación, al igual que se definen las cláusulas teniendo en cuenta algunos procesos preestablecidos por defecto, las fechas de inicio y finalización del contrato, la cantidad de dinero que gastará el consumidor, los participantes y el saldo del contrato.

La cuenta se registra con su ID de *hash* y el saldo, y luego el contrato está listo para validarse y almacenarse en el *blockchain*. Para las pruebas iniciales de funcionamiento del proceso de validación, se utilizó como algoritmo de consenso el *PoW*, y el proceso producto de esta tesis, el *PoR*, para disminuir el consumo de recursos y los tiempos de respuesta en la generación de bloques, dando oportunidad inclusiva a los validadores de acuerdo con su comportamiento en la red. En la sección 4.3 se presenta el protocolo propuesto.

Después de guardar el contrato en el *blockchain*, ambos participantes pueden comenzar a hacer transacciones. Un contrato inteligente en esta propuesta, comienza con la transacción que define la cantidad de moneda que el consumidor quiere gastar.

El proceso de almacenamiento de la transacción es similar al proceso de almacenamiento del contrato (ver Figura 8). Cada transacción contiene un registro del contrato relacionado (dirección del contrato o ID de *hash* del contrato) y se valida utilizando el mismo algoritmo de consenso. Esta transacción actualiza el saldo de la cuenta del contrato de 0 a la cantidad de dinero acordada en el contrato y la billetera de cada participante se ve afectada. Si el consumidor prometió 5,5€ , su billetera debería mostrar una disminución de 5,5€ y el proveedor recibiría esos 5,5€ . La transacción incluye la dirección del consumidor (id. *hash* del cliente), la dirección del contrato (id. *hash* del contrato), la cantidad de moneda a enviar, la fecha en que se realiza la transacción y la dirección del proveedor (id. *hash* del proveedor) como destinatario de la cantidad de la moneda.

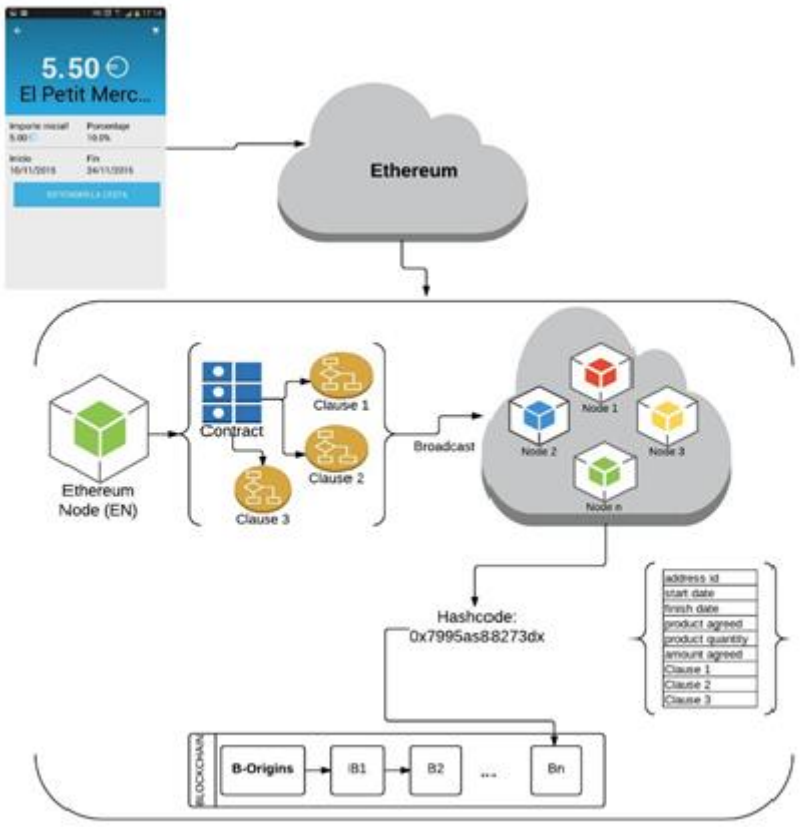


Figura 7. Proceso para guardar un contrato *EurakosNext* en el *blockchain*

Fuente: el autor

Mientras el contrato esté activo, el consumidor puede mantenerse en contacto con el proveedor y utilizar los servicios que desee de acuerdo con la cantidad de dinero acordada previamente. Para cada servicio, el proveedor debe registrar la factura correspondiente y el consumidor debe aceptarla, si todo está correcto; luego, cada transacción debe publicarse por difusión en la red de Ethereum para su validación y posterior almacenamiento en el *blockchain*. La transacción se almacena como una identificación con *hash*, pero contiene el *hashId* del contrato, la cantidad de moneda acordada, la dirección del participante que realiza la transacción y la fecha.

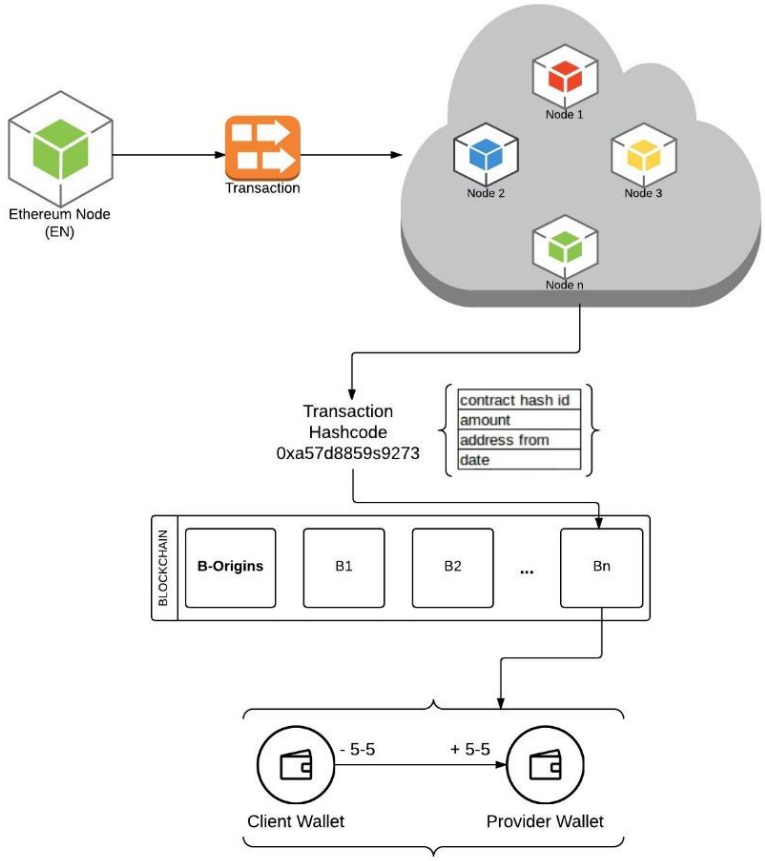


Figura 8. Proceso para guardar una transacción de *EurakosNext* en el *blockchain*

Fuente: el autor

Si el saldo llega a cero (0) antes de la fecha límite y no hay quejas de nadie, el contrato puede cerrarse. Si la fecha actual es la fecha límite, las cláusulas se revisan para decidir si se debe aplicar una multa. Si alguna parte ha fallado el compromiso, es posible intentar resolver el problema en el momento y si no hay acuerdo, se aplican las sanciones correspondientes. Si alguna de las partes no está de acuerdo con la solución, puede solicitar una mediación. Después de que se cierra un contrato, no es posible llevar a cabo más operaciones.

A continuación se presenta el proceso realizado para crear un modelo de contrato Eurakos en Ethereum:

En la Figura 9, se muestra la interfaz de programación de los contratos de EurakosNext, utilizando la herramienta MIX de Ethereum. En la parte izquierda se puede observar la estructura de archivos de la aplicación (nótese que hay un archivo con extensión .sol por cada contrato), y en la parte derecha la zona de programación y compilación de contratos.

Los archivos html corresponden a aquellos que despliegan la interfaz web, la que permite al usuario interactuar con el contrato. Esta interacción se programa mediante el uso de funciones JavaScript o por medio de *Webservices* RESTful. Recordar que para que el contrato se despliegue en un entorno web, se debe tener instalada la plataforma Ethereum al igual que un servidor de páginas web como el Apache Tomcat. Para este caso en particular, se utilizó tecnología JSF (*JavaServer Faces*) para las interfaces web y clases *BackBeans* para la comunicación con Ethereum.

La Figura 10 presenta un contrato en ejecución que captura los elementos que permiten construir una cesta de productos.

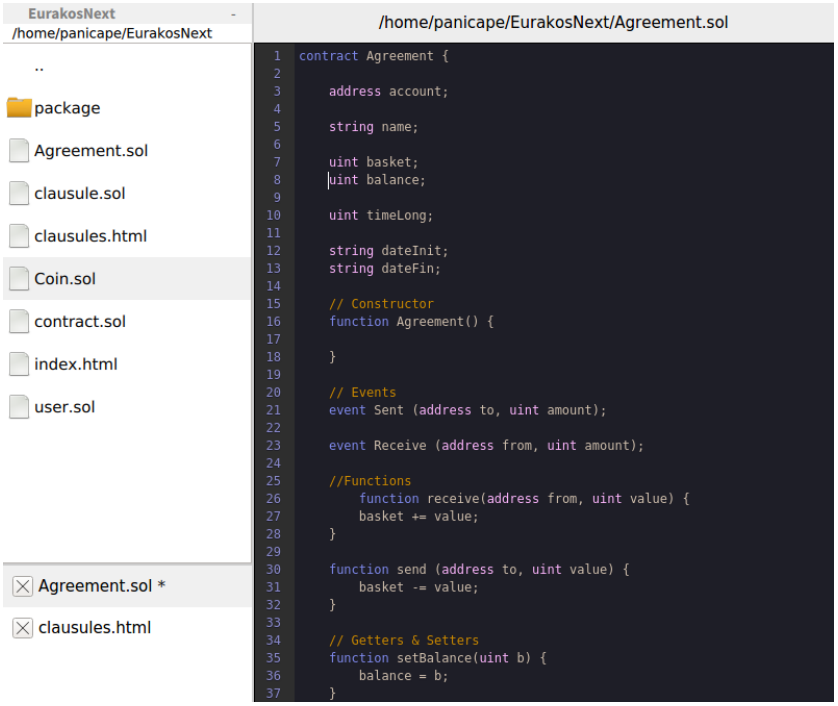


Figura 9. Entorno de programación de contratos en Ethereum utilizando la herramienta de desarrollo MIX

Fuente: el autor

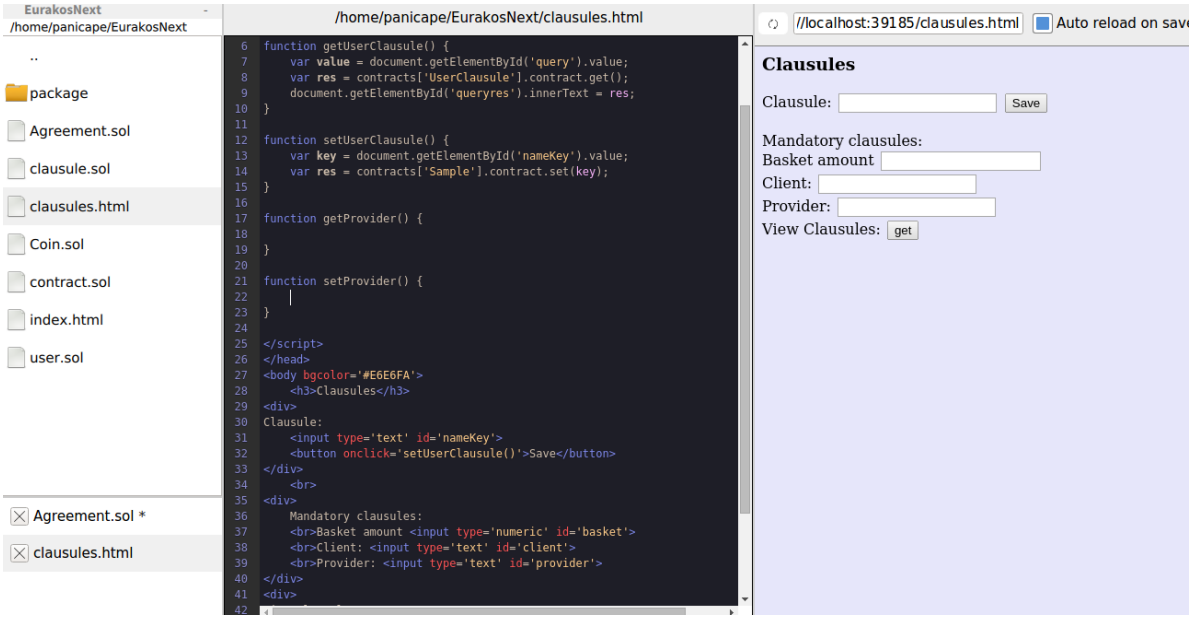


Figura 10. Interfaz de captura de elementos de una cesta Eurakos

Fuente: el autor

4.3 Protocolo de consenso para EurakosNext

4.3.1 Introducción

Uno de los objetivos de esta investigación, es optimizar los recursos tecnológicos disponibles para mejorar los tiempos de respuesta, garantizando la seguridad de las transacciones y alentando la participación de los usuarios de criptomonedas en estas tareas, de la manera más sencilla posible. En ese sentido, se ha orientado el diseño del protocolo de consenso para la validación de las transacciones y contratos de la criptomoneda EurakosNext.

Inicialmente se partió de un escenario de simulación creado con dispositivos móviles que interactuaban con un servidor, para ayudar en las tareas de sincronización de la información y la asignación de roles en el *blockchain* de *Ethereum*. El consenso de validación fue diseñado utilizando los siguientes dos niveles (basado en dos protocolos de consenso conocidos: *Proof of Work y Proof of Stake):*

1. *Proof of Work*, que implementa la moneda y verifica si la transacción es válida y la almacena en el *blockchain*.
2. *Proof of Stake*, que valida el contenido de los acuerdos cuando éstos han finalizado o porque se han cancelado.

Según los resultados de las mediciones llevadas a cabo, se pudo concluir que estos dos procesos en conjunto, aumentaban el costo de las comisiones y ralentizaban la validación como tal. Estas demoras aparecieron a causa de los recursos físicos y de moneda, involucrados en los cálculos de identificación del código *hash* (se midieron demoras de 1 segundo para la obtención de resultados).

En la búsqueda del mejoramiento de estos procesos, se optó entonces por diseñar un protocolo único que corrigiera los problemas encontrados y le facilitara a las comunidades de la moneda, vincularse de una u otra manera a la validación de transacciones y contratos de EurakosNext en diferentes etapas y recibir a cambio una compensación a su esfuerzo, según el nivel de dificultad involucrado; fue así como surgió entonces la propuesta del PoSb (Prueba de participación según la pertenencia – *Proof of Stake on belonging*) que se introduce a continuación.

4.3.2 Protocolo Proof of Stake on belonging PoSb

Descripción

Este protocolo se basa en el Algoritmo de Colonias de Abejas Artificiales ABC (Gajendra, 2014), que organiza los roles de los participantes en la validación, siguiendo diferentes comportamientos inteligentes de los enjambres de abejas. Esto significa que un validador puede desempeñar un papel sólo si es parte de un enjambre.

El ABC es un algoritmo cooperativo que intenta simular la forma como se realiza el comportamiento de recolección de alimentos. Pretende ser una manera de manejar procesos complejos segmentándolos en partes más pequeñas para que sean procesadas por muchos participantes.

En la naturaleza, las abejas tienen su propia estructura jerárquica para sobrevivir. Esto significa que tienen algunas funciones que incluyen diferentes actividades que deben llevarse a cabo, y por esta razón, es necesario describirlas. La estructura del algoritmo ABC aplicada al EurakosNext está compuesta por:

a. Dos actividades principales:

1. Búsqueda de fuentes alimenticias (transacciones o contratos)
2. Explotar recursos (validar transacciones o contratos y almacenarlos en el *blockchain*)

b. Tres roles:

1. *Scouts*: tienen la responsabilidad de buscar continuamente los recursos disponibles para ser explotados. En la naturaleza, los recursos a encontrar son los alimentos. Con EurakosNext, estos pueden ser físicos o también tareas pendientes de validación.
2. *Workers*: publicar tareas para validar, dividir tareas complejas en tareas más pequeñas si es necesario.
3. *Onlookers*: explotar recursos

La siguiente Figura, muestra un aspecto de la organización del trabajo por parte del Algoritmo ABC.

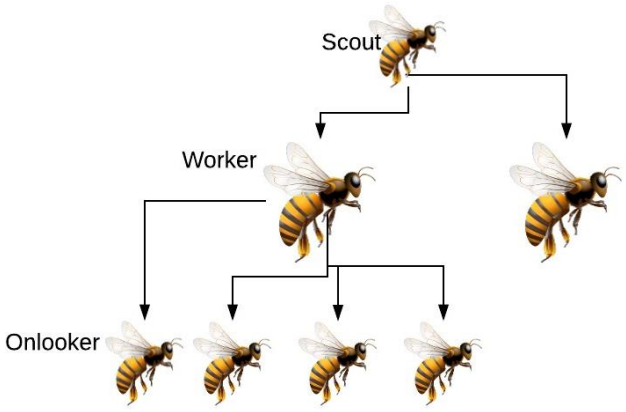


Figura 11. Esquema de trabajo de los participantes en un proceso de validación mediante el Algoritmo ABC

Fuente: Carrillo, De la Rosa y Peña (2017)

Funcionamiento

Usando el algoritmo ABC se llevan a cabo las siguientes tareas en EurakosNext:

1. Almacenar transacciones o contratos: que sigue el proceso de validación del PoW de Ethereum. Comienza a aplicar el algoritmo de encriptación SHA-256 para obtener un *id* de *hash* como resultado y luego publicarlo en la red de Ethereum para ser confirmado por un mínimo de 15 validadores o mineros. En EurakosNext, esta tarea se realiza en un grupo de minería porque muchos participantes la ejecutarán desde sus propios terminales móviles.

Un grupo de mineros es un grupo de participantes que une recursos para mejorar su capacidad de validación y tratar de obtener más recompensas por ello. En este caso, un grupo minero ayudará a replicar la misma capacidad de un nodo Ethereum normal, por esta razón, un *worker* debe analizar el esfuerzo necesario para definir el número de *onlookers* requerido para validar transacciones o contratos.

2. Validar el contenido del contrato: utiliza el algoritmo de consenso *Proof of Stake*, lo cual significa que los validadores se unirán al proceso de minería sólo si pueden demostrar la propiedad de una cantidad de moneda específica.

La validación del contenido implica el cálculo de todas las transacciones de un contrato de acuerdo con cada participante, lo cual hace que los mineros ejecuten las cláusulas o recompensas del contrato de acuerdo con el código escrito en el contrato relacionado. Los *Scouts* deben definir los contratos a validar y los trabajadores (*Workers*) distribuir las tareas de acuerdo con los mineros disponibles.

Articulando los conceptos anteriores, los contratos EurakosNext contarían con los siguientes elementos:

a. Las funciones Init() y kill(), para inicializar los requisitos de cada contrato y deshabilitar el acceso a él respectivamente. La función kill() es un método que deshabilita el acceso al contrato que lo implementa. Se recomienda su uso como mejores prácticas en el desarrollo de la programación en lenguaje Solidity, pero es necesario tener cuidado porque después de ejecutarla ya no es posible interactuar con el contrato que la implementa.

b. Eurakos, que es responsable de la administración de la moneda Eurakos. Por lo general, los contratos que representan y administran una moneda se denominan *tokens*, y sus tareas consisten principalmente en administrar el saldo general de las cuentas y confirmar que no se transfiera nada más que lo solicitado. Este contrato también tiene una función denominada askReward (), que se llama para recompensar el esfuerzo realizado por los validadores.

La codificación de estos contratos se puede observar en el ANEXO 1.

c. Product: este contrato contiene una representación básica del producto ofrecido por un proveedor; facilita el seguimiento que da a cada parte en acuerdos entre cliente y proveedor. (Ver código en ANEXO 2).

d. Basket: este contrato es la representación de una cesta Eurakos. Establece el sector en el que un cliente desea gastar una cierta cantidad de dinero y la cantidad en sí (en el inicio). El método isValidBasket () permite a los validadores determinar si la cesta está bien formada y si se ha realizado la transferencia de fondos para gastar. Los proveedores que desean que los clientes consuman en su establecimiento, ofrecen una cantidad adicional de Eurakos para gastar en su tienda (utilizando el método addOffer ()) y, posteriormente, la decisión del cliente elige al proveedor para las provisiones de gastos (setChoosenOffer ()). El código de este Contrato se puede observar en el ANEXO 3.

e. Agreement: este contrato contiene una estructura que representa el acuerdo entre un cliente y un proveedor, donde las condiciones iniciales se toman de la cesta creada previamente por el cliente; valueToSpend es el valor para gastar definido en la cesta, propietario es el ID del cliente (es una dirección asignada por Ethereum), el proveedor es el ID del proveedor al que se aceptó la oferta (también es una dirección asignada por Ethereum). El método doClientPay () con el cual el cliente paga lo que consume, disminuye la cantidad establecida inicialmente para gastar y al alcanzar este valor no es posible seguir gastando y el Contrato estará pendiente de validación. Los métodos cancelAgreement() o finishAgreement() determinan las acciones finales del acuerdo, o si una de las partes cancela el acuerdo. Ver código en el ANEXO 4.

f. SwarmValidators: Este contrato se encarga de asignar el rol de cada validador; la idea es que en el futuro un validador pueda pertenecer a más de un enjambre, en función de su ubicación geográfica, pero solo puede estar activo en un solo enjambre al mismo tiempo. Dependiendo del rol, el método actionSelection () asigna una acción predeterminada, disminuyendo la cantidad establecida inicialmente. Por lo tanto, si se alcanza este valor, no puede continuar gastando y el Contrato estará en espera de ser validado. Los métodos cancelAgreement () o finishAgreement () determinan la acción a realizar al final del acuerdo o si se cancela.

Si el rol es el de un *Scout*, se llama el método scoutAction () para buscar dispositivos cercanos que contengan elementos para validar y agregarlos a una lista pendiente de elementos de validación del Contrato. Estos artículos pueden ser una cesta, un acuerdo o un nuevo producto del proveedor.

Si el rol es de *Worker*, se llama el método workerAction () para distribuir los elementos que se validarán entre los *Onlookers* existentes.

Si el rol es de *Onlooker* se llama el método onlookerAction() para llevar a cabo la validación del elemento en función de su naturaleza.

El ANEXO5 muestra la codificación de este contrato.

Observaciones relativas al proceso

La utilización del Algoritmo de las abejas ABC fue una alternativa propuesta por esta investigación, para generar un protocolo de consenso que permitiera validar las transacciones de la criptomoneda EurakosNext, tratando de superar los inconvenientes presentados con algunos de los protocolos de consenso existentes, que dejan latente aun una que otra barrera de las identificadas para la adopción de este tipo de monedas de manera natural.

Las ventajas ofrecidas por el Algoritmo ABC frente al problema planteado fueron las siguientes:

1. La cooperación entre los miembros de un enjambre, permite dividir la ejecución de la validación de transacciones en pequeñas partes a partir de las capacidades reales de cada nodo.
2. Cada nodo es consciente de que si realiza su tarea correctamente le permite a otros nodos también participar en la ejecución de tareas correctamente. Esto no sucede con otros de los protocolos examinados, pues los validadores se vinculan al proceso sólo con el ánimo de recibir una compensación económica.

Y, como desventaja se puede mencionar que para utilizar este tipo de algoritmo cooperativo para la implementación de un protocolo de consenso, habría que partir de cero para diseñar el proceso de descubrir nuevos validadores, asignar tareas y cambiar la manera en que los validadores se comunican para determinar si han completado sus tareas o si han cambiado de rol y eso implica demasiado trabajo y desperdicia los recursos que ya han sido desarrollados por otros investigadores y de una u otra manera se han incorporado a las plataformas o entornos que están dispuestos a los usuarios con libre acceso para realizar implementaciones personalizadas de criptomonedas.

Ethereum ofrece un sistema de contratos inteligentes que permite diseñar nuevas criptomonedas y configurar su *blockchain* para tal fin, entonces se optó por conocer a fondo los beneficios de algunos de los protocolos de consenso en funcionamiento y presentados en la sección 4.3, para solucionar el problema de reducir parte de las barreras de adopción de las criptomonedas en aspectos relacionados con: En este caso, se pensó que la mejor opción podría ser desarrollar un sistema híbrido para aprovechar los beneficios de algunos protocolos y mitigar los ataques de otros.

El *Proof of Authority* por ejemplo, permite simplificar el esfuerzo que necesita un nodo para validar una transacción (mediante un sistema de roles) en comparación con el PoW y el PoW puede apoyar al *Proof of Authority* para detectar si algún validador se ha saltado el método para beneficio propio mediante su propia ejecución y así detectar que está realizando validaciones usando su propia máquina.

A continuación se presenta el protocolo de consenso *Proof of Reputation* PoR (prueba de reputación), resultado de esta decisión y con el cual se da cumplimiento a parte de los objetivos propuestos en esta tesis, que enfoca sus esfuerzos en la generación de soluciones que reduzcan las barreras de adopción de criptomonedas, en términos de: los factores emocionales - generando confianza entre los miembros de la comunidad asociada a la moneda; la gestión de la moneda – quitando a los usuarios dicha responsabilidad y delegándola a la tecnología; y la infraestructura y entorno donde operan las monedas - seleccionando de acuerdo al contexto, la mejor apariencia de la moneda.

4.3.3 Protocolo de consenso Proof of Reputation (Prueba de reputación) PoR

Descripción

Este protocolo se diseñó sobre una plataforma *Ethereum* porque ofrece la facilidad de generar nuevas criptomonedas aprovechando su arquitectura descentralizada, y se programó en lenguaje Solidity.

Consiste en un híbrido que aprovecha los beneficios de los procesos de validación del *Proof of Work* y del *Proof of Authority*, así como su buena gestión de la seguridad, entendida esta, no como que esté exento completamente de los ataques propios de cada uno de estos protocolos, sino que articulados entre ellos se mitiguen los ataques y además se limite el número de veces que se ejecute PoW, para generar confianza en el validador por medio de un esfuerzo y después se pueda seguir validando y aumentando su reputación por medio de la ejecución del PoA.

Proof of Reputation reemplaza el sistema de votación utilizado por *Proof of Authority (PoA)* para agregar nuevos autorizados *(authorities)* a validar con PoW, esto quiere decir, que en vez de que los autorizados existentes voten para aceptar nuevos validadores, éstos deben realizar un mínimo de validaciones con PoW para poder ser agregados a la lista de autorizados; posteriormente las siguientes transacciones se validan con PoA (asignándole la reputación respectiva al validador si las validaciones se realizaron satisfactoriamente). Si los validadores fallan en algún momento, deben ejecutar de nuevo una validación con PoW para confirmar que siguen siendo confiables.

El PoA normalmente comienza teniendo un único validador o *Authority* con una cantidad de moneda disponible limitada desde el principio. La incorporación de nuevos validadores se realiza por votación de todos los que se encuentran disponibles, por lo que al principio únicamente votará un validador por la incorporación de otro y al entrar el segundo, los dos deberán decidir si el siguiente ingresa y así sucesivamente, por tanto, cada vez será más difícil registrarse como validador.

Un validador puede perder su reputación si falla alguna validación y, por tanto, deberá volver a comenzar a validar con PoW para recuperar la reputación que le permitirá seguir validando.

La siguiente expresión define el modelo general del protocolo PoR:

$$x' = A \vee B \quad (5)$$

Donde:

x´: es el resultado de la validación de la transacción actual

$$\mathbf{A} = SHA3(nonce, challenge) \quad (6)$$

□R > 0 ∧ □R < (x+i) (ejecución del PoW)

$$\mathbf{B} = SHA3(isAuthority(address), address) \quad (7)$$

□R >= (x+i) (ejecución del PoA)

Siendo,

$$\mathbf{R} = (R + 1) * Tx_S \quad (8)$$

Donde,

R: es la reputación del validador

$\boldsymbol{Tx_S}$: es el estado de la validación de la transacción que toma los valores **1** si salió exitosa o **0** si resultó fallida.

Cada vez que se valida correctamente una transacción, esta reputación se incrementa. Por tanto, si la reputación inicia con un valor de 0, y la primera transacción validada resulta exitosa, la reputación actualizada sería igual a 1 que corresponde al reemplazo de datos así, de la Ecuación (8):

R = (0 + 1)*1

En caso contrario, si la validación no hubiera sido exitosa, automáticamente la reputación tendría un valor de 0 ya que Tx_S tomaría el valor de 0 por dicho fallo.

x: el número de validaciones iniciales a realizar con PoW para ganar reputación

i: un contador de validaciones a realizar con PoW por penalización en fracaso (validación no exitosa)

SHA3(nonce, challenge): es el método de criptografía que aplica el PoW basado en una entrada (*nonce*) y un *hash* objetivo a solucionar (*challenge*).

nonce es la entrada a guardar que al encriptarse debe cumplir con los requisitos del *challenge* para poder ser almacenada en el *blockchain*.

El *challenge* es un objetivo impuesto al *nonce* encriptado, el cual debe cumplir una cantidad específica de ceros al inicio del *hash* para ser aceptado como válido y posteriormente ser guardado en el *blockchain*. Este valor va cambiando cada vez que se llama el algoritmo de validación, por tanto, al validar una transacción se puede pedir 3 ceros al inicio del *hash*, en el siguiente pedir 5 ceros y así continua tomando un valor aleatorio a través de todas las validaciones.

***SHA3(isAuthorithy(address), address)*:** es el método de criptografía que aplica el PoA cuando se valida cada transacción. Este método requiere como entrada una bandera que diga que un validador es realmente quien és y la dirección identificadora del mismo.

Simulación para verificación del comportamiento del protocolo

El comportamiento del protocolo PoR se refinó mediante simulación utilizando diferentes variaciones de equilibrio de uso del PoW y el PoA. A continuación se define el proceso de la simulación base.

Línea de base

La primera versión del PoR determina que los validadores registrados en el sistema comienzan sin reputación, por tanto, deben ejecutar la primera validación de una transacción usando PoW como requisito para obtener una reputación por encima de 0. Cuando se confirma que la validación se realizó correctamente, las siguientes transacciones se deben continuar validando con PoA para incrementar su reputación de 1 en 1 por cada validación exitosa.

Si la transacción no fue validada exitosamente, la variable que controla la reputación del validador se resetea a cero y se incrementa en 1 el número de transacciones necesarias a validar con PoW, como castigo por haber fallado, antes de poder continuar validando con PoA.

5. RECOLECCIÓN DE DATOS

5.1 Introducción

A partir del modelo general del PoR planteado en la expresión (5) para definir el comportamiento del protocolo PoR, se procedió a establecer el híbrido PoW-PoA de mejor comportamiento a la hora de validar transacciones, y para eso se llevaron a cabo simulaciones con las siguientes características:

1. Se realizó el mismo tipo de transacción múltiples veces y se tomaron los tiempos utilizados en las diferentes validaciones con el fin de determinar si el proceso le representó al validador un mayor esfuerzo.

2. Se introdujeron algunas variaciones en la cantidad de validaciones requeridas por el PoR en relación con la aplicación del PoW y el PoA, para determinar la eficiencia en la velocidad de validación y mantener la reputación suficiente para que gran parte de los validadores sigan participando en la validación.

5.2 Datos de Prueba

En la siguiente Tabla se presentan los datos tomados como base para las simulaciones de comprobación del funcionamiento del modelo PoR. Esta tesis no tiene en cuenta el análisis del comportamiento de los validadores para la asignación de recompensas, por lo que se mantiene el proceso utilizado por *Ethereum.*

Tabla 6. *Datos utilizados para las simulaciones que permitieron comprobar el funcionamiento del modelo PoR*

***Blockchain*:**	*Ethereum* en blanco
Número de transacciones:	5000 utilizando por separado los tres métodos PoW, PoA y PoR
Identificador de cuentas de los participantes:	Número de cuentas: 5 Direcciones: 1. 0x00367aDeA9b42a3DA534Cc4288CED2bCe53EDffe 2. 0x007541B87000ad0C6C4cdD7ee2D6F3026B739f4c 3. 0xB49f0eeFe8182870993DcDFB1De6DFbdcf2Fb7d9 4. 0x00f2511CA6862a7a8aBc2a33dcd112767b5D276B 5. 0x00F33289Ba20f1A0e22062625f93Ebd0be8Ba489
Balance en *Ethers* en las billeteras de los participantes:	5 *Ethers*

5.3 Validación mediante Proof of Work (PoW)

El código utilizado para esta simulación fue el básico existente en la plataforma Ethereum que está programado en lenguaje solidity (ver ANEXO 6).

5.4 Validación mediante Proof of Authority (PoA)

Se utiliza en este caso, el protocolo PoA desarrollado por Gavin Wood, funcionando sobre la plataforma Ethereum. Mediante este sistema, los autorizados colaboran para crear la cadena más larga del *blockchain*, en lugar de utilizar el esquema de Proof of Work. Inicialmente, el primer rol lo adquiere el creador de la cadena, el cual define los primeros autorizados a participar y estos votando por otros participantes les permiten incorporarse como validadores.

Ver código en ANEXO 7.

5.5 Línea de base: Proof of Reputation V1 (PoRV1)

Se comienza validando una transacción con PoW (los validadores ingresan sin tener reputación, por tanto, es necesario realizar cierto esfuerzo para ganarla); si se lleva a cabo satisfactoriamente, se tendrá una reputación suficiente para seguir validando. Cada validación satisfactoria representa un punto de reputación.

Partiendo de la expresión que define el modelo del PoR articulada con la reputación (ver Expresión (5)), para la simulación base, se comienza asignando a la variable **x** el valor de 1 y cada vez que un validador falle alguna validación, esta variable se incrementará en 1 y la variable **R** se reseteará a 0. Esto quiere decir que la primera validación se realizará con PoW para comenzar a obtener reputación; luego, se continuará validando con PoA para seguir ganando reputación. Cada vez que falle un proceso de validación la variable **x** aumentará en una unidad indicando que se deben validar ahora 2 transacciones exitosas con PoW para poder continuar validando con PoA y así sucesivamente. Tener en cuenta igualmente, que para calcular la reputación se aplica la fórmula de la Ecuación (8) que utiliza a su vez la variable Tx_S que estará variando entre 0 y 1 dependiendo de si la validación salió exitosa o no.

El siguiente diagrama presenta el flujo de este proceso.

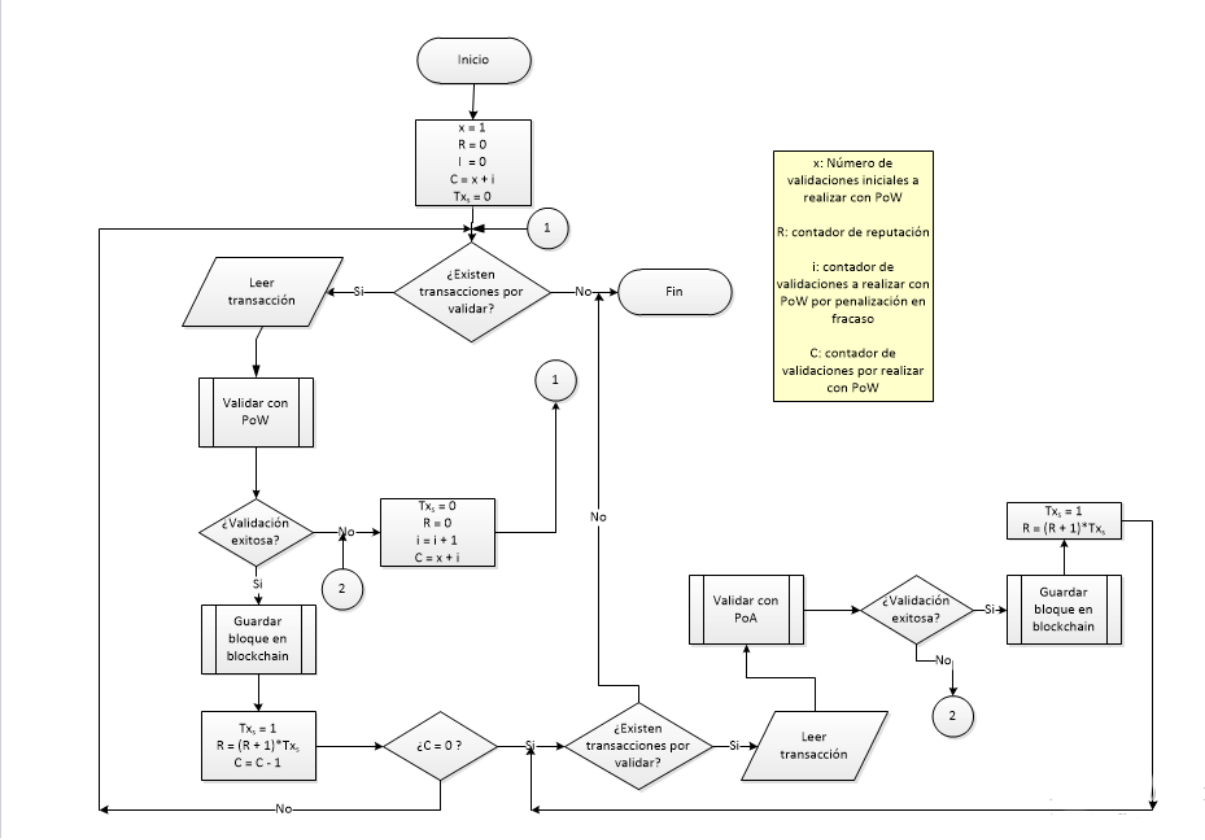


Figura 12. Proceso de validación con PoRV1 en la línea de base

5.6 Proof of Reputation V2 (PoRV2)

Para este caso, los validadores ingresan al proceso sin tener reputación, y por tanto la deben adquirir validando un mínimo de 10 transacciones utilizando PoW; si todas estas validaciones resultan exitosas se considera que existe una reputación suficiente, para seguir adquiriendo reputación y continuar validando. Recordar que validar con PoA le permite al validador seguir adquiriendo reputación para continuar en el sistema.

Utilizando el modelo del PoR propuesto en la Ecuación (5), la variable **x** comienza con un valor de 10 y se incrementa en 1 cada vez que un validador falle alguna validación, y la variable R se resetea a 0. Esto quiere decir que inicialmente se comienza validando 10 transacciones con PoW para ganar la reputación inicial (un punto de reputación por validación exitosa con PoW) y si estas validaciones resultan exitosas se continúa validando con PoA para seguir adquiriendo reputación. De manera similar al caso del PoRV1, cada vez que se falle en una validación con PoA, se deberá volver a comenzar a validar con PoW incrementando en 1 el contador de validaciones iniciales (es decir que si **x** comenzó en 10, ahora su valor sería 11) y la reputación se resetearía 0.

En la siguiente Figura se presenta el flujo de este proceso.

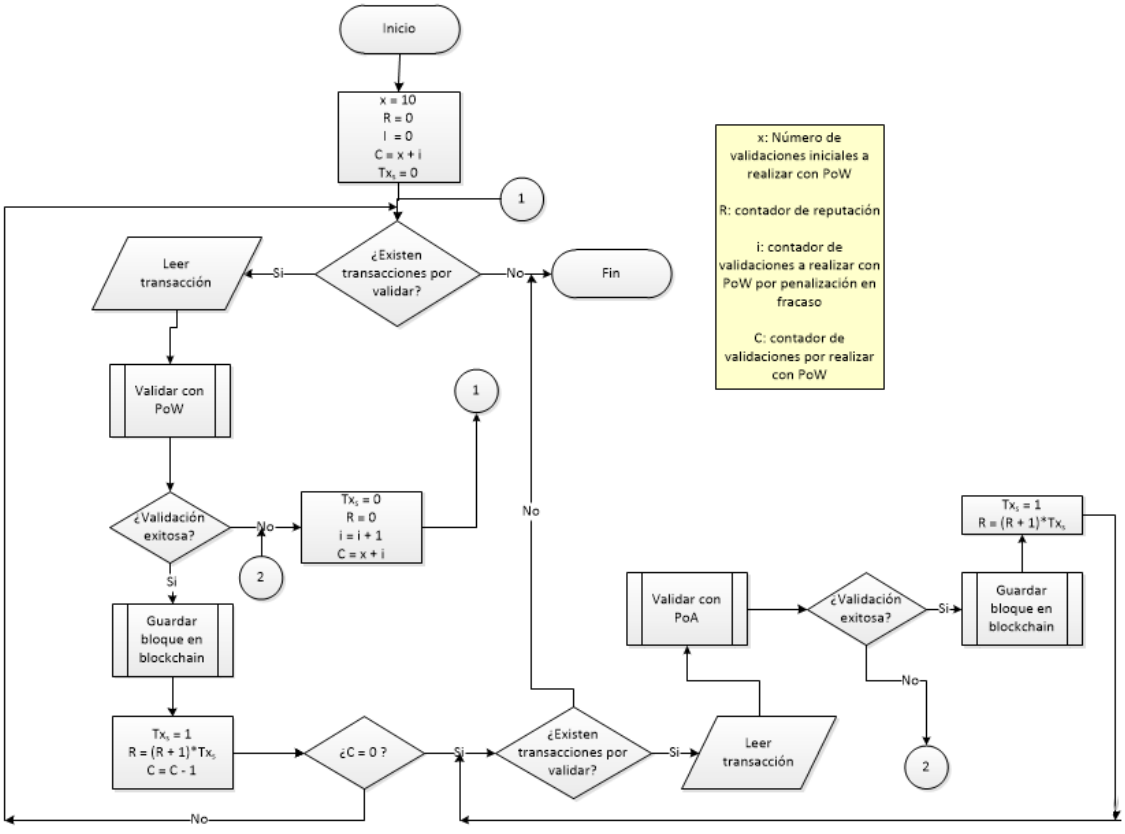


Figura 13. Representación del funcionamiento del PoRV2

5.7 Proof of Reputation V3 (PoRV3)

Este caso es una evolución del PoRV2 que consiste en comenzar a validar 10 transaciones con PoW y luego, después de haber validado exitosamente 100 transacciones con PoA, se valida nuevamente 1 con PoW con el fin de verificar que la reputación ha sido adquirida correctamente. Si se supera esta validación, se continúa realizando el proceso de manera cíclica entre validar 100 con PoA y 1 con PoW. En caso de fallo, el proceso se reinicia de acuerdo con lo propuesto en la Ecuación (5) teniendo en cuenta que a la variable que contabiliza el número de validaciones con PoW a ejecutar al inicio (o sea 10), se le incluye el valor del contador de penalizaciones efectuadas por fallo en alguna validación.

El modelo ajustado del PoR, teniendo en cuenta esta mejora, tendría la siguiente apariencia:

$$x' = A \vee B \vee C \quad (9)$$

Donde:
x’ = es el resultado de la validación de la transacción actual

A y B permanecen con la misma definición dada en las expresiones (6) y (7)

$$C = SHA3(nonce, challenge) \; \square \; e = \left(\left(R - (x + i)\right)/100\right) \in Z^{+} \quad (11)$$

La expresión (11) denotaría entonces la ejecución cíclica de 1 PoW después de haber validado exitosamente 100 transacciones con PoA. En caso de fallo el proceso se reinicia ejecutando A con el número de validaciones exigidas en PoW con base en las iniciales establecidas (10) más las penalizaciones por fallo.

El beneficio de esta mejora es propiciar que ataques que aprovechan un solo tipo de validación puedan ser mitigados con el uso de otro método complementario. Por ejemplo, cuando se valida sólo con PoA, atacantes que se hagan pasar por validadores autorizados pueden no superar las validaciones con PoW que se realizan cada cierto tiempo y por tanto, perderían su reputación, y hasta que no se realice la validación con PoW de la cantidad de transacciones requeridas para

adquirir reputación no se podrá seguir validando y en consecuencia no se tendrá acceso a la ganancia de las recompensas por dichas validaciones.

El diagrama de flujo de este proceso se puede observar en la siguiente Figura.

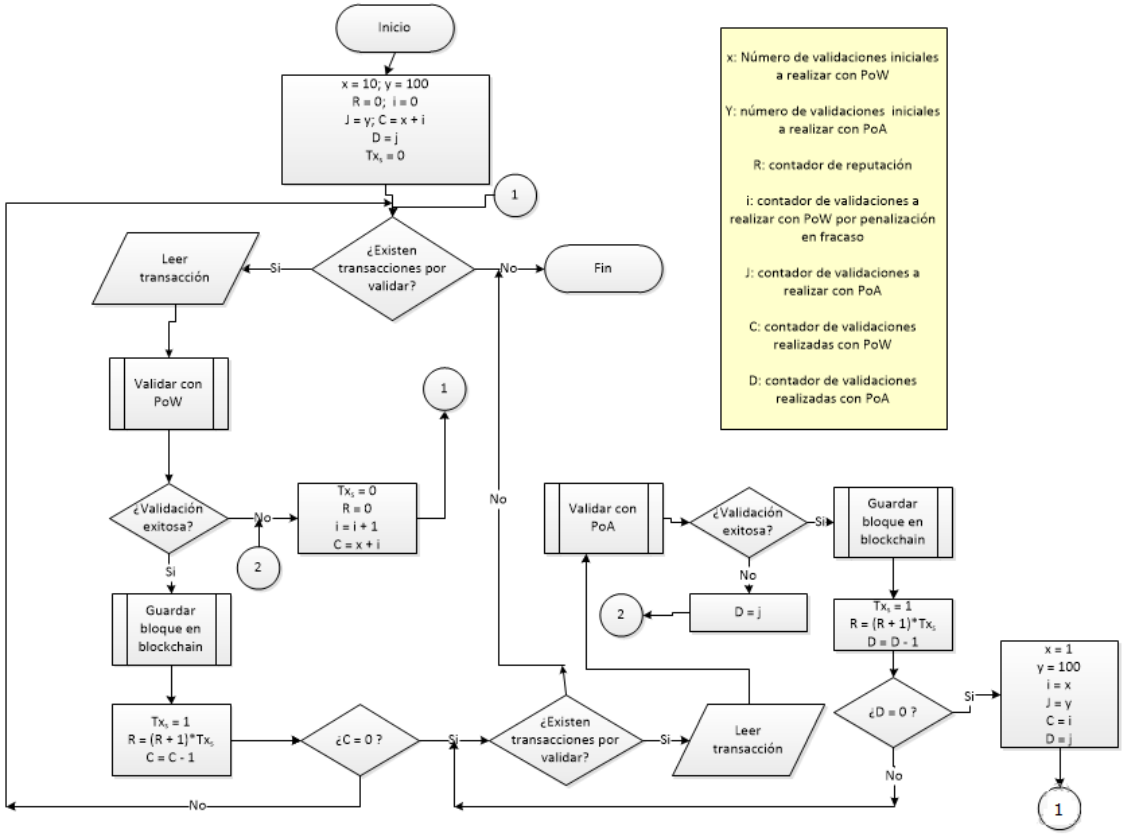


Figura 14. Representación del funcionamiento del PoRV3

6. ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

Con base en los datos recolectados en la sección 5 por medio de las simulaciones realizadas para las validaciones con PoW, PoA y PoR en sus versiones 1 a 3, se llevaron a cabo análisis, tomando como variable de comparación, el tiempo en segundos, que le lleva a las transacciones ser validadas, para poder concluir la combinación óptima del modelo PoR.

En la gráficas, el eje vertical representa el tiempo en segundos y el horizontal, el número de transacciones validadas. Las leyendas corresponden a los colores que identifican las cuentas de los validadores (*hash*) y en líneas punteadas la tendencia predominante del proceso que será lineal o exponencial según el caso.

6.1 Validación con PoW

La Figura 15 muestra el comportamiento de la validación con PoW en el uso del tiempo. Hay que recordar que este protocolo es el más utilizado por la mayoría de las criptomonedas. En la imagen se presenta el proceso de validación de 257 transacciones y se puede observar que el tiempo promedio utilizado es de **6** segundos (tendencia lineal).

6.2 Validación con PoA

En la Figura 16, se puede observar que el tiempo promedio de validación de 253 transacciones mediante PoA es de 2,5 segundos (tendencia lineal) y que después de que los nodos son aceptados por la comunidad de validadores, no se refleja un gran esfuerzo en la ejecución de este protocolo, en comparación con el PoW.

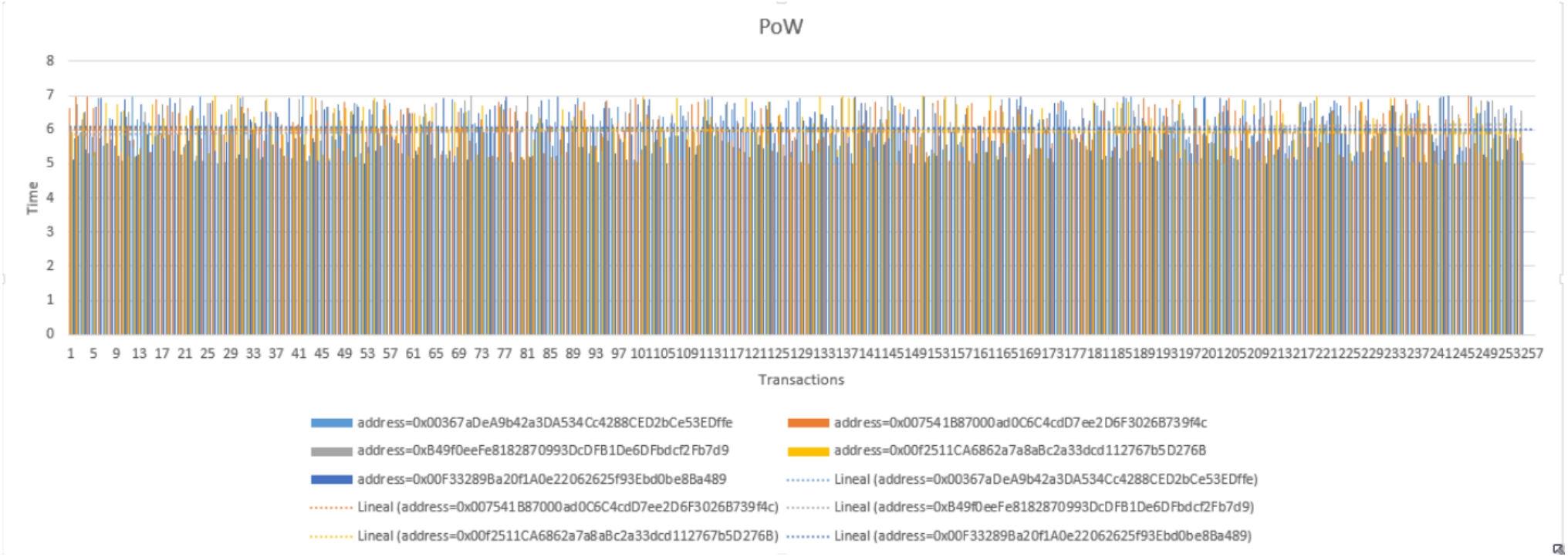


Figura 15. Representación del tiempo utilizado para validar transacciones mediante PoW

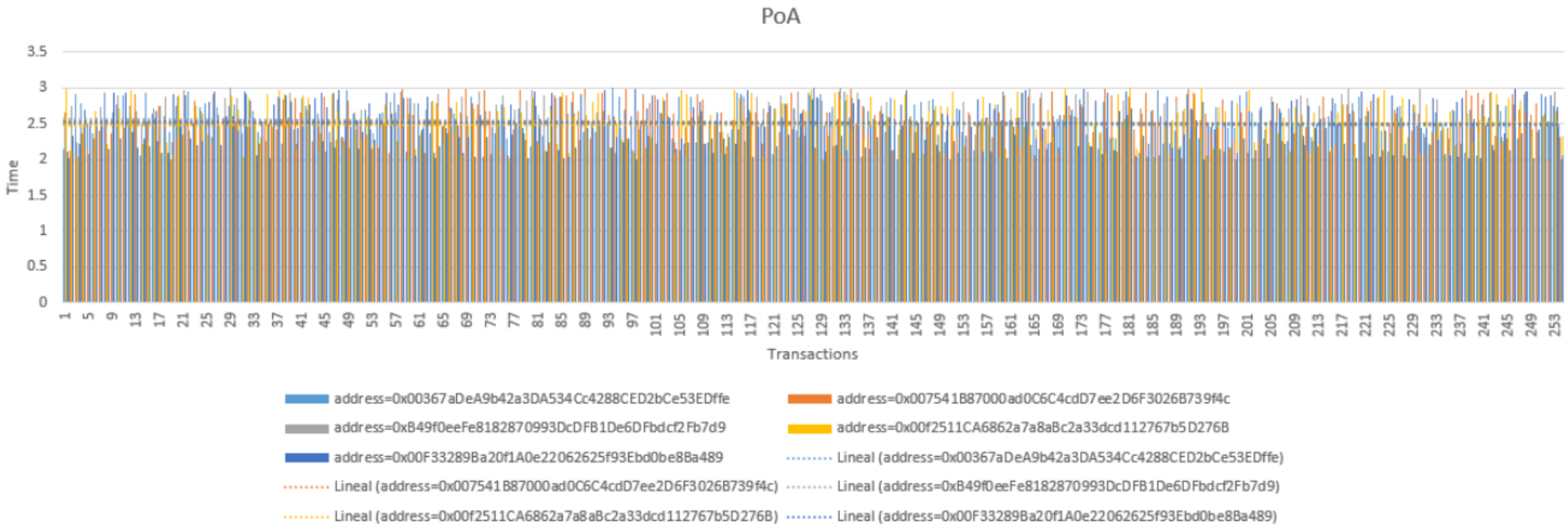


Figura 16. Representación del tiempo utilizado para validar transacciones mediante PoA

6.3 Validación con PoRV1

Tomando como línea de base el PoRV1, en la Figura 17 se puede observar la variabilidad del tiempo utilizado para iniciar validando una transacción con PoW que le permitirá al validador exitoso, adquirir confianza y continuar validando con PoA para fortalecer su reputación y tener la posibilidad de seguir en el sistema.

La imagen corresponde al proceso realizado hasta la transacción 505. En el área demarcada en rojo, se muestra que la primera validación realizada con PoW tarda casi 7 segundos, en cambio, al continuar validando con PoA para seguir adquiriendo reputación, existe más participación de los validadores y no se requiere tanto uso intensivo de la potencia de hardware de los mismos. El problema que puede surgir, es que la mayor parte del tiempo los validadores quedan expuestos a que los atacantes se hagan pasar por ellos y sigan firmando transacciones para obtener la recompensa
. El tiempo promedio de validación con PoA es de 2,5 segundos como se presentó en la sección 6.2.

Comparando las Figuras 15 y 16 con la 17, se observa que validando con PoRV1 el esfuerzo se reduce considerablemente (de 6 segundos en promedio en la primera transacción a 2,5 segundos en las siguientes transacciones) ya que la dificultad de validar con PoW no aumentará tan rápido como sí sucede si se validara solo con PoW (6 segundos en promedio siempre para todas las transacciones).

.

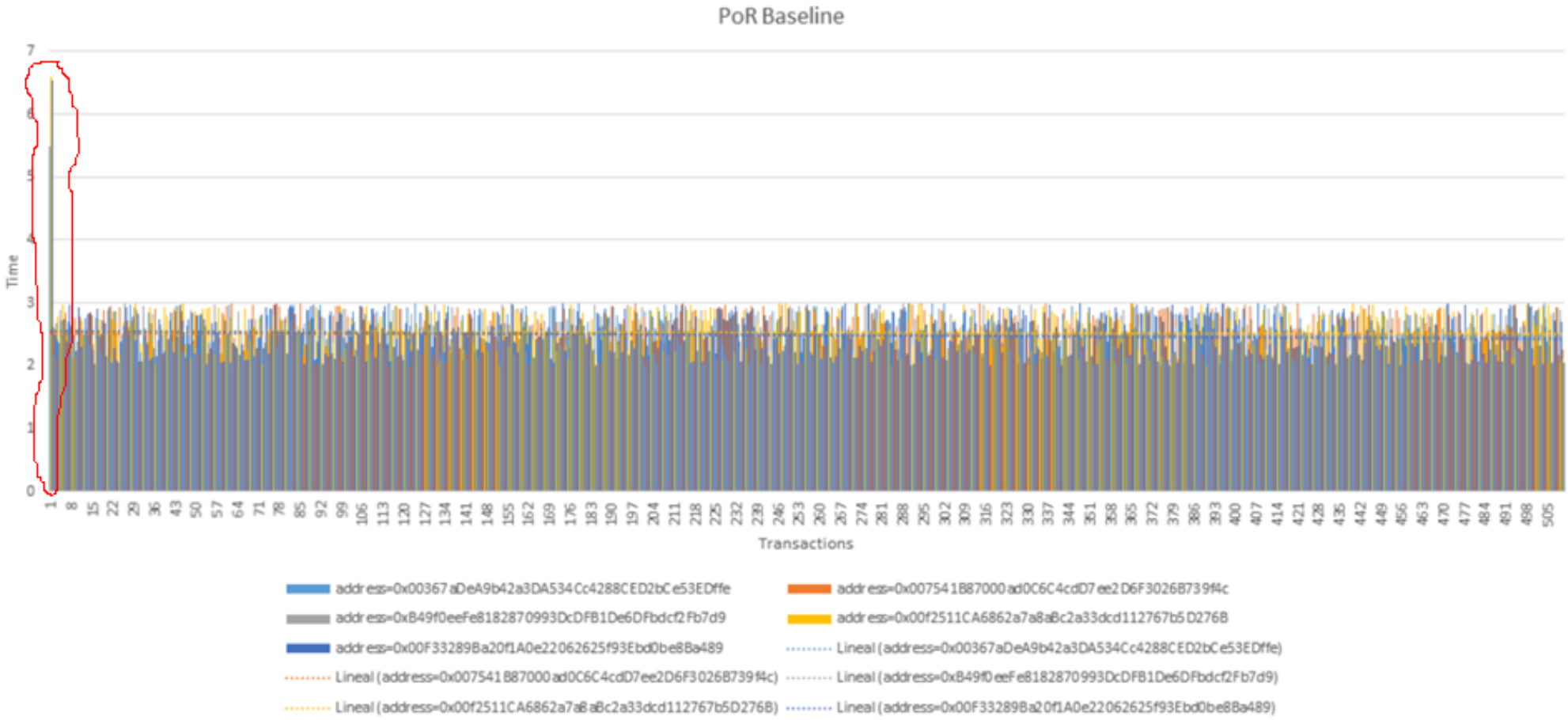


Figura 17. Representación del tiempo utilizado en la validación de transacciones mediante PoRV1

6.4 Validación con PoRV2

Este protocolo aumenta la dificultad del PoRV1, ya que la exigencia para ganar reputación y continuar validando con PoA es haber validado exitosamente con PoW, 10 transacciones. O sea, si se observa la Figura 18, las validaciones iniciales que implican la ejecución del PoW, tardan un máximo de 7 segundos y después, ese tiempo disminuye al comenzar a utilizar el PoA, por lo cual, se puede ver que esta propuesta de PoR tiende a disminuir el esfuerzo necesario a medida que se van validando transacciones, por tanto, el esfuerzo seria inversamente proporcional al incremento de transacciones validadas (tendencia exponencial), y se alcanzaría el objetivo de ser más confiable a medida que se aumenta la reputación del validador.

El único inconveniente que puede surgir como se mencionó también anteriormente, es que al continuar tanto tiempo validando con PoA, los validadores pueden aprovechar este método de validación para explotar otros ataques.

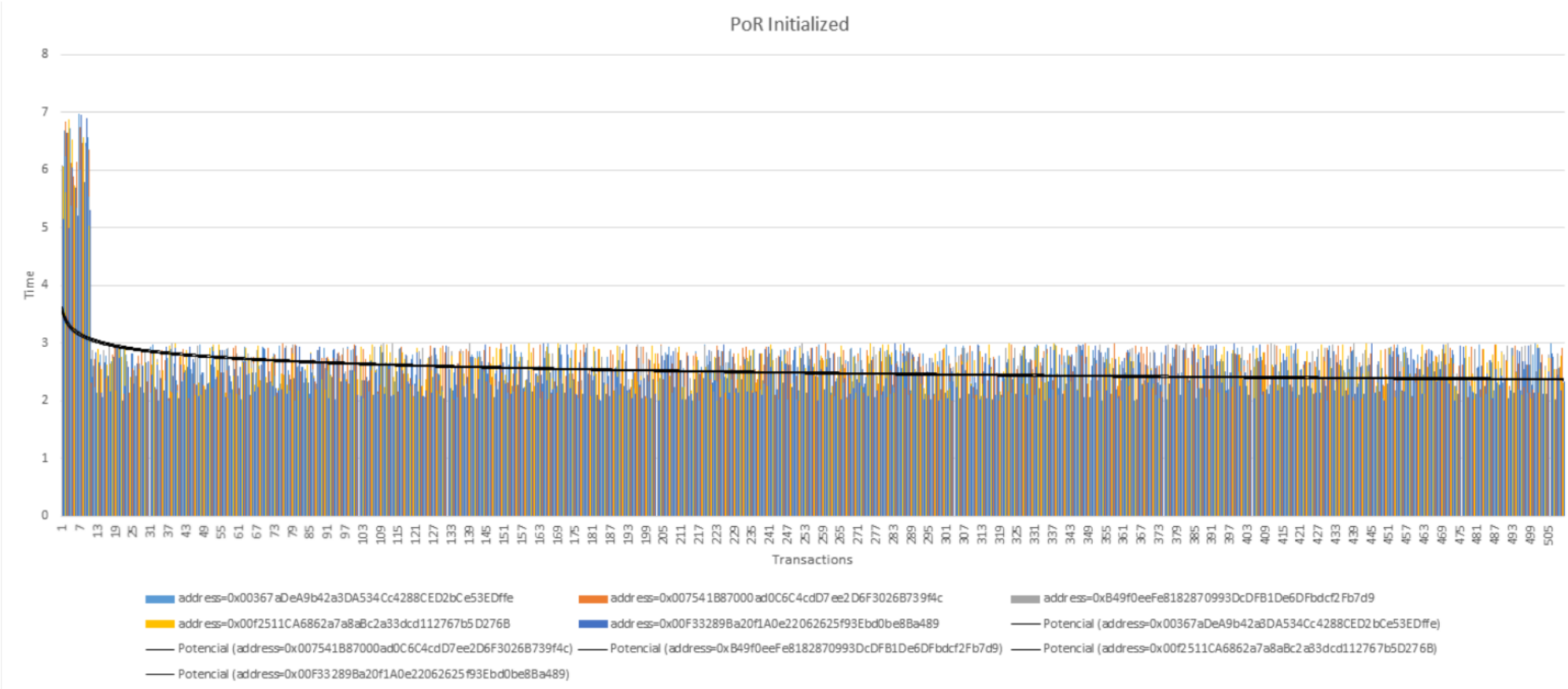


Figura 18. Representación del tiempo utilizado para validar transacciones mediante PoRV2

6.5 Validación con PoRV3

La Figura 19 muestra la ejecución de la simulación en la que se varía el uso del PoW y del PoA de manera iterativa, de tal manera que se comienza validando 10 transacciones con PoW para ganar reputación y poder continuar validando con PoA, pero ahora el proceso de validación con PoA también tiene un control y es que por cada 100 transacciones validadas y exitosas se debe volver a validar 1 transacción con PoW, esto con el fin de comprobar si la reputación se ha ido adquiriendo correctamente. Si esta validación con PoW no se supera, la reputación se resetea y se debe volver a validar con PoW según el valor del contador de penalizaciones por fallos.

Se puede concluir de acuerdo con el resultado de la gráfica, que a pesar de incorporar validaciones intermedias de PoW para mitigar los ataques al PoA por ser ejecutado por largo tiempo, que el tiempo promedio de validación sigue siendo 2,5 segundos. Por lo tanto, esta versión del protocolo PoR se considera que podría ser la implementación óptima que resuelve en parte el problema de investigación planteado en la presente tesis.

Para complementar esta decisión, se hace referencia al trabajo de Dos Santos y Swan (2018) que ayuda a entender también la importancia de evitar el aumento de la complejidad a las validaciones (ellos realizaron un análisis del PoW y el PoS). El híbrido PoW-PoA propuesto en esta tesis, disminuye el tiempo de validación de las transacciones en más de un 80% y ayuda a mitigar ataques directos contra el PoW o el PoA.

.

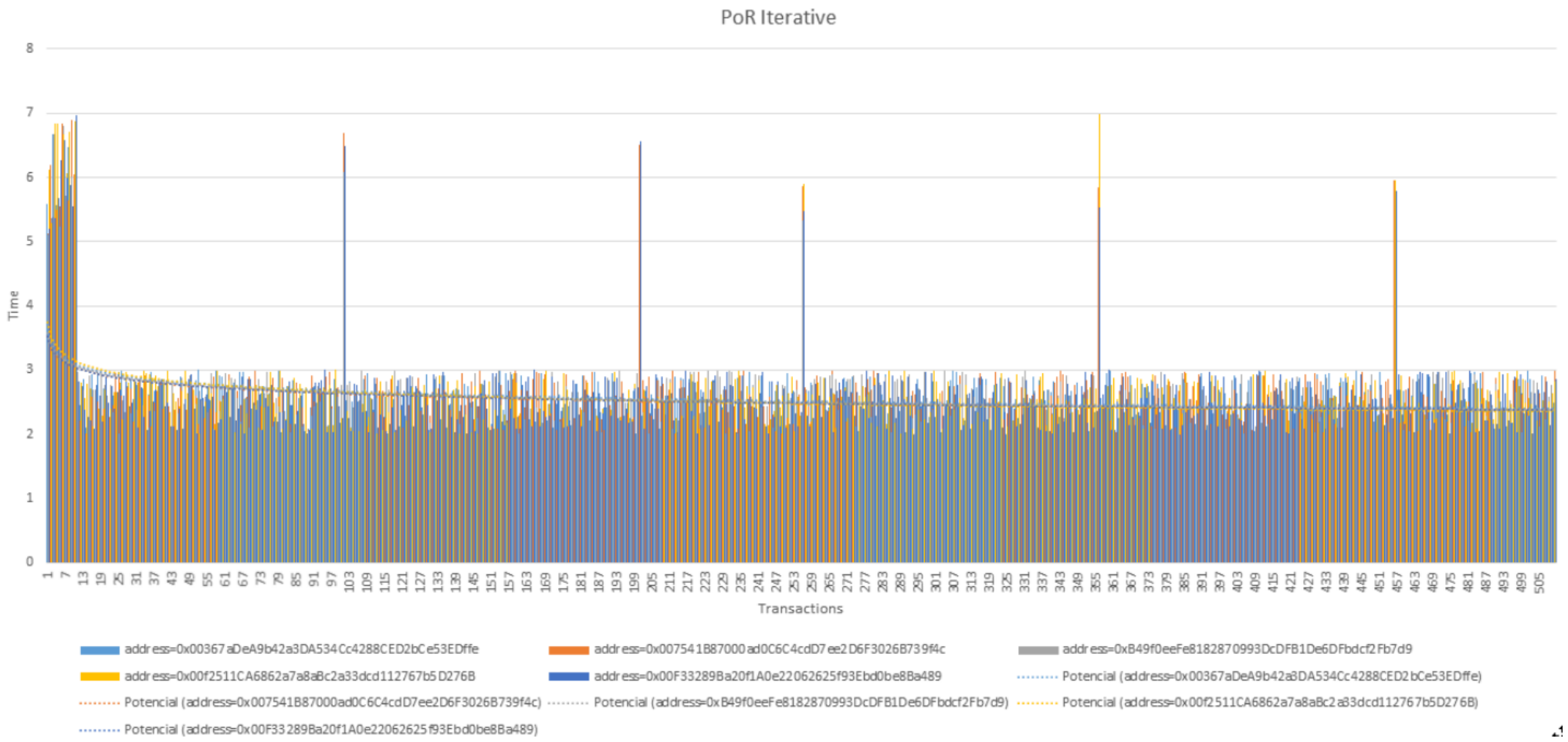


Figura 19. Representación de tiempos para validar transacciones mediante PoRV3

7. CONCLUSIONES

En los aspectos financieros existe integrada la triada Comercio-Mercado-Dinero, que relaciona diversas formas de intercambio, diversas formas de comercio y diversos tipos de moneda. Un sistema de mercado es un sistema cultural específico que con elementos económicos, políticos o ideológicos permite establecer formas de intercambio. Este intercambio requiere un sistema de convertibilidad de valor, que se ve universalmente representado por la moneda. Una moneda de curso legal, respaldada por un gobierno o banco central, era una herramienta que facilitaba el intercambio de productos y servicios y, estaba respaldada por un bien con valor intrínseco (oro, plata, seda, otros). El respaldo físico se eliminó, dando paso a una economía exponencial que permite la acumulación excesiva y, colateralmente, el colapso.

Las monedas sociales y complementarias fueron creadas para ayudar a los usuarios a pagar sus cuentas y permitir el desarrollo de gremios o regiones; y las criptomonedas se establecieron para mejorar la transaccionalidad y la seguridad. Esto ha hecho que cobren sentido, principalmente porque limitaciones como la velocidad de uso, la seguridad en las transacciones realizadas y la confianza las han convertido en una solución.

Bitcoin propuso una solución a la transaccionalidad basada en Blockchain. Esto permitió la descentralización para evitar que las funciones de asegurar el dinero existente y validar que las transacciones se realicen correctamente, se remitiera únicamente a los bancos. Superó, por tanto, el que una sola entidad le estableciera políticas y tasas exageradas como pago por su trabajo. Bitcoin privilegió el rol de la validación a aquellos que quisieran participar a cambio de una recompensa y entregó la protección del dinero a los propietarios de las carteras.

El proceso de validación que gran parte de las criptomonedas ha usado es la Prueba de Trabajo o *Proof of Work (PoW),* que está conformado por dos partes: encriptación y challenge. Para superar el *challenge* y obtener el *hash* que confirma la validación, es posible que se requieran millones de ejecuciones del algoritmo de encriptación y por tanto, cantidad elevada de recursos de hardware. Esta razón ha conllevado la propuesta de otras formas de validación, como lo demuestran protocolos como el *Proof of Stake (PoS), Proof of Stake on Velocity (PoSV), Proof of Authority (PoA), Tangle, Redundant Byzantine fault of tolerance (RBFT),* entre otros.

La validación de transacciones con criptomonedas utilizando un único protocolo de consenso, puede ser que mitigue algunos de los problemas o fricciones que surgen en estos procesos, pero a su vez puede aumentar la complejidad de su ejecución al cabo del tiempo ya que puede ralentizar su funcionamiento y/o abrir otras vulnerabilidades. Las siguientes consideraciones son importantes de tener en cuenta a la hora de decidir la mejor solución a esta problemática:

1. **Velocidad de la moneda**: un tipo de validación que contribuya a la disminución de algunas de las fricciones de las criptomonedas, pueda ser que bloquee algún tipo de ataque durante el proceso, pero en últimas los usuarios lo que esperan es que las transacciones se realicen y validen lo más rápido posible y sin errores, por tanto, si no se cumplen estas expectativas, lo más probable es que vuelvan a utilizar la moneda de curso legal como medio transaccional.

2. **Democratización en la validación y uso de la moneda**: el problema que tiene una validación que requiera mucha inversión en hardware y que genere mucho costo derivado, es que al final únicamente los que puedan adquirir las herramientas especializadas y mantener su funcionamiento, son los nuevos actores que reemplazarán a los bancos, pero la idea es que participantes más pequeños o no tan expertos (permitidos por la facilidad de uso) puedan usar monedas complementarias o validar criptomonedas sin reemplazar un poder por otro.

La validación a partir de la autoridad reconocida a un validador (principios del Proof of Authority) permite la aceleración del proceso al no tener que satisfacer el “challenge”. Sin embargo, es posible que el criterio de reconocimiento de la autoridad no sea tan respetable y aceptado.

Esta tesis pretende entre otros, simplificar y acelerar la validación en el largo plazo, y mitigar algunos tipos de ataques. Esto lo logra al reconocer autoridad a personas reconocidas, con lo cual se aceleraría el proceso, pero reconociéndolas mediante el éxito de un número determinado de validaciones según la relación encriptación - "challenges" para conformar su reputación. Este es el protocolo propuesto en esta investigación y se denomina Prueba de Reputación (*Proof of Reputation*) o PoR. Este protocolo exige un esfuerzo del validador para confirmar la ejecución con una máquina propia y que tiene una reputación adquirida honestamente para validar con menos esfuerzo (sin challenges) pero con la misma seguridad.

La reputación establecida en el método de validación PoR, además de ser un concepto integrador, permite darle una definición más significativa al trabajo que realizan los validadores. Un validador se considera altamente confiable si ha alcanzado una reputación alta durante los procesos de validación. Como trabajo futuro, sería interesante incluir la manera de implementar las recompensas por el esfuerzo realizado, dependiendo de la reputación alcanzada.

Un sistema de validación no es completamente perfecto y ninguno puede garantizar la totalidad de la validación de transacciones, ni su democratización. Por tanto, a partir de los análisis realizados en esta investigación, se considera que debe existir un equilibrio entre el esfuerzo requerido para validar una transacción y quien está autorizado a realizar dicha acción, para evitar que se convierta en un sistema lento e intrincado.

Un sistema como el PoR, permite la ejecución de la validación hasta que supere un número determinado de validaciones, según la relación encriptación - "challenges", para confirmar que es un validador real y no una réplica y, que la base del reconocimiento permita a cada validador incrementar la confianza en sí mismo (al incrementar su reputación por cada validación superada satisfactoriamente). Por otra parte, pensando en el usuario final, el PoR ayuda a superar barreras psicológicas como la desconfianza en el uso de una criptomoneda ya que es una validación lógica y no solo programática, e independiente de elementos externos o intrínsecos de su propio sistema.

La validación con hardware verifica que alguien real participa con su máquina y la validación con autorización previa brinda más velocidad de ejecución. El beneficio de haber realizado análisis mediante las relaciones encriptación - “challenges”.asegura que hay una entidad detrás de la validación, dando seguridad al proceso. En este aspecto, el PoW supera al PoR y éste al PoA. El beneficio de haber realizado un análisis teniendo como punto de comparación el tiempo en que se ejecutan las transacciones, muestra que el PoR supera el PoW pero no al PoA. Esto quiere decir que las debilidades ofrecidas por otros sistemas de validación se ven minimizadas por las fortalezas del PoR. Éste utiliza una proporción menor de la relación encriptaciòn-“challenges” pero homologa de forma periódica la Reputación. Esto hace que no tenga tantos requerimientos de hardware como el PoW y le da más seguridad al algoritmo en relación con el PoA.

El PoR permite adquirir reputación mediante hardware (relación encriptación “challenges” requerido inicialmente para ser aceptado como validador, y homologado periódicamente para verificar que el validador mantiene su correcto actuar) y validar mediante una firma de la transacción (haciendo el proceso muy rápido). Las Figuras 13 (del comportamiento del PoW), 14 (del comportamiento del PoA) y 15, 16 y 17 (versiones del comportamiento del PoR), confirman lo expresado anteriormente.

A pesar de que PoW y PoA sean los algoritmos más referenciados por ser estables y seguros, presentan problemas como requerir mucha potencia de hardware para validar (PoW) o que uno de los mayores ataques sea el de hacerse pasar por validador para obtener las recompensas (PoA). El PoR mejora estas condiciones.

Comparativamente entre las tres versiones del PoR, la versión 1 y la versión 2, se pueden parecer bastante al comportamiento real que se establece en la implementación del PoA de Ethereum, por tanto, podrían darse las mismas debilidades del PoA. El beneficio que tiene la versión 3 es que se enfoca más en la seguridad del consenso pues sacrifica tiempo en algunas validaciones que requieren hardware para prevenir ataques de suplantación. En conclusión, se considera una mejor implementación la versión 3 del PoR, porque no solo se tiene en cuenta la optimización en tiempo sino una mayor seguridad en el proceso de validación.

Entonces, con respecto a la pregunta de investigación,

¿Los nuevos diseños de monedas complementarias, teniendo en cuenta el desarrollo de las redes sociales y las tecnologías blockchain y de contratos inteligentes, cómo pueden mejorar los negocios?

Se puede establecer que los negocios actuales pueden mejorar la seguridad, velocidad en sus transacciones y motivación de su personal, mediante el uso de tecnologías cuasi instantáneas, transparentes y seguras, al reducir los costos de transacción, los tiempos de validación y el conocimiento del proceso, independientemente del lugar y tiempo en que se realice.

Adicionalmente a este desarrollo conceptual, por otra parte, el desarrollo de la Tesis le ha permitido al autor desarrollar destrezas en la presentación de informes, en la organización de la implementación, en el reconocimiento de trabajos previos, en la organización del trabajo y en las estrategias y dinámica de investigación.

7.1 Trabajo a futuro

Generar un nuevo sistema de validación puede conllevar una complejidad mucho mayor en comparación con los existentes, si este depende de factores que poco pueden ser controlados, ya sean externos (la existencia y propiedad de un documento) o internos (poseer una cantidad de moneda por cierto tiempo). Esto puede ser visto en problemas de uso del propio sistema, por ejemplo, en Proof of Stake disminuir la velocidad de uso de la moneda y en Proof of Existence, registrar como propios, documentos de conocimiento general como una constitución, la biblia, el Corán o el teorema de Pitágoras entre otros.

Los tipos de consenso estudiados en esta investigación, constituyen un gran mercado en el cual todos compiten por ser tomados en cuenta como el más ligero y seguro, futuras investigaciones deberían enfocarse en establecer arquitecturas de validación por capas, dependiendo del objetivo que requiera la criptomoneda. Así se podría dedicar por ejemplo, en una arquitectura de dos capas para mejorar el funcionamiento de la tecnología blokchain, una capa para el control de la seguridad que mitigue los ataques y otra que sea la especializada en la validación de las transacciones.

Se deben mejorar las consideraciones de seguridad con respecto a la sincronización del Blockchain y la comunicación entre los nodos de la red, ya que ataques comunes con otros tipos de redes, son aprovechados para que algunos nodos se beneficien de las recompensas asignadas por algunas validaciones o para acaparar todo el poder de validación de la red de nodos.

Para facilitar la utilización de este método en diferentes criptomonedas, se debería complementar el desarrollo con herramientas intuitivas, interfaces sencillas, casos de uso, y elementos que expliquen en qué consiste la seguridad en términos de verificación de la existencia del validador y de la reputación que se adquiere durante el proceso para continuar validando.

La investigación realizada sobre los modelos de validación también permitió tomar conciencia con respecto a la confianza del sistema, que no solamente está dada por el hecho de que funcione bien o mal, sino por la manera como ésta es vista por los usuarios. Si de la misma forma como algunas empresas muestran orgullosos sus datos sobre las ventas realizadas, se presenta en las interfaces de usuario de las criptomonedas información aproximada de las transacciones que han sido validadas hasta el momento y el número de validadores con reputación alta que están participando en los procesos y lo que esto significa, los usuarios rápidamente podrían sentirse cómodos viendo este movimiento y que realmente existen personas que se preocupan del buen funcionamiento del sistema y de su seguridad, lo cual motivaría el mantenerse en el sistema y atraería más usuarios.

La principal conclusión obtenida del análisis realizado en las variaciones del protocolo PoR para determinar cuál es la mejor configuración es que la versión 3 (en ella se validaban 10 transacciones con la relación encriptación-challenge y después de 100 validaciones se homologaba la reputación mediante 1 transacción con la relación encriptación-challenge) es la más apta ya que permite garantizar una mejor seguridad sin que esto implique aumentar exponencialmente la complejidad del algoritmo ni la dificultad para validar transacciones.

Esta investigación puede ser continuada en dirección a definir las acciones a aplicar a aquellos validadores que excedan un límite de fallos a fijar. La idea en la que se puede continuar es sobre qué acciones tomar si el validador falla muchas veces, ya que lo más probable es que, al final, éste no complete exitosamente ninguna (la dificultad relacionada a la primera parte del algoritmo habrá aumentado tanto que el mismo validador no podrá realizar su trabajo sin la ayuda de nadie más). Una opción viable es definir las condiciones para que éste sea bloqueado o agregado a una especie de "lista negra", y posteriormente eliminado de la lista de authorities.

Por otra parte, también se podría seguir variando la cantidad de validaciones iniciales necesarias de la relación encriptación-challenge según la reputación adquirida antes de fallar, por ejemplo, si falla con poca reputación podría aumentarse la cantidad de validaciones iniciales para volver a adquirir reputación, pero si tiene una reputación alta o muy alta, no se aumentaría o incluso se podría disminuir.

Se necesita seguir estudiando la cantidad de recompensa con la cual se le retribuye a los validadores y la comisión que se cobra a los usuarios por validar una transacción.

Se está trabajando en la escritura de 3 artículos producto de esta investigación desde diferentes perspectivas para las siguientes revistas:

1. *Open Access Journal "Future Internet"* (Factor de impacto 0,22 Q3 h-index 10), para el cual se recibió invitación con base en la difusión y aceptación que tuvo el artículo Nro. 6 de la lista de publicaciones realizadas por el autor de esta tesis. https://www.mdpi.com/journal/futureinternet

2. *Journal of Information Technologies* (Factor de impacto 4,535 Q1 h-index 66). https://link.springer.com/journal/41265

3. *International Journal on Webservice computing*: http://airccse.org/journal/jwsc/ijwsc.html

8. REFERENCIAS

Antonopoulos, A.M. (2014). *Mastering Bitcoin: unlooking digital cryptocurrencies*, Sebastopol, CA:O´Reilly.

Baeck, P. and Bria, F. (2014). *Digital social innovation: what it is and what we are doing*, London:Nesta.

Bentov, I., Lee, C., Mizrahi, A., and Rosenfeld, M. (2014). Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake. *ACM Digital Library,* Vol 42, Issue 3, 34-37.

Berger, J. (2013). *Contagious: Why Things Catch On*, New York:Simon & Schuster Inc.

BitcoinWiki. (2015). Proof of Work. Recuperado de https://en.Bitcoin.it/wiki/Proof_of_work

Blackcoin. (2015). Currency of the future. Recuperado de: http://blackcoin.co/

Blanc, J. (2011). Classifying CCs: community, complementary and local currencies' types and generations, *International Journal of Community Currency Research*, 15 (2011), 4-10.

Block Fortune. (2017). Characteristics of cryptocurrencies. *Bitcoin News.* Recuperado de http://www.blockfortune.com/2017/07/characteristics-of-cryptocurrencies/

Blondeau, O., Whiteford, N., Vercellone, C., Kyrou, A., Corsani, A., Rullani, E., Boutang, Y., and Lazzarato, M. (2004). *Capitalismo cognitivo – Propiedad intelectual y creación colectiva*, España:Traficantes de sueños.

Buntinx, J. (2015). Is Bitcoin A Digital Currency or a Virtual Currency?, News Bitcoin.com. Recuperado de https://news.Bitcoin.com/is-Bitcoin-a-digital-currency-or-a/

Buterin, V. (2015). A next generation smart contract & decentralized application platform, Ethereum white paper. Recuperado de https://www.weusecoins.com/assets/pdf/library/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf

Cachin, C. (2016). Architecture of the Hyperledger Blockchain Fabric. IBM Research – Zurich. Recuperado de https://www.zurich.ibm.com/dccl/papers/cachin_dccl.pdf

Carrillo, C., De la Rosa, J. Ll. & Canals, A. (2007). Towards a Knowledge Economy, *International Journal of Community Currency Research IJCCR, Vol* (11), 84-97. http://dx.doi.org/10.15133/j.ijccr.2007.005

Carrillo, P., Peña, C.I. and De la Rosa, J.Ll. (2016). Eurakos Next: A Cryptocurrency Based on Smart Contracts. In: Nebot, Angela; Binefa, Xavier; López Ramón (eds.), Artificial Intelligence Research and Development. *Frontiers in Artificial Intelligence and Applications*, 288, 221-226.

Carrillo, P.N., De la Rosa, J.Ll. y Peña de Carrillo, C.I. (2017). Validating EurakosNext cryptocurrency transactions using the Artificial Bees Colony Algorithm – an approach. *IV Conferencia Internacional Monedas Sociales y Complementarias, Dinero, Conciencia y Valores para el Cambio Social*, Universidad Oberta de Catalunya, Barcelona, España, 23p.

Cassano, J. (2014). What are Smart Contracts? Cryptocurrency's killer app, *APP Economy*. Recuperado de http://www.fastcolabs.com/3035723/app-economy/smart-contracts-could-be-cryptocurrencys-killer-app

Cialdini, R. (2006). *The psychology influence of persuasion*. New York:William Morrow & Company, Inc.

Criptonoticias. (s.f.). ¿Qué es Ethereum? Recuperado de https://www.criptonoticias.com/informacion/que-es-ethereum/

Curran, B. (2018). What is Proof of Authority Consensus? Staking Your Identity on The Blockchain. BLOCKONOMI. Recuperado de https://blockonomi.com/proof-of-authority/

De La Rosa, J.Ll., Batlle, J., Batlle, E., Szymanski, B. and Krishnamoorthy, M. (2009). A design of complementary community currencies for education. *International Conference on Computer Supported Education, CSEDU 2009*, Lisbon, Portugal.

Delmolino, K., Arnett, N., Kosba, A., Miller, A., and Shi, E. (2015). Step by step towards creating a safe smart contract: lessons and insights from a cryptocurrency, *Lab. International Association for Cryptologic Research IACR*.

Dwork, C. & Naor, M. (1993). Pricing via Processing, Or, Combatting Junk Mail. *Advances in Cryptology, CRYPTO'92: Lecture Notes in Computer Science*, No. 740. Springer: 139–147.

Dziembowski, S., Faust, S., Kolmogorov, V. y Pietrzak, K. (2015). Proofs of space. *CRYPTO*, 585–605.

Dos Santos, R., and Swan, M. (2018). PoW, PoS, & Hybrid protocols: A Matter of Complexity?. Cornell University Library, 14p. Recuperado de: https://export.arxiv.org/ftp/arxiv/papers/1805/1805.08674.pdf

EUMED. (2002). Red eumed.net. Recuperado de: http://www.eumed.net/cursecon/economistas/Gesell.htm

ERIS. (2016). Eris & Ledger form partnership to further blockchain technology. Bitcoin Technology. Recuperado de https://www.ccn.com/eris-ledger-blockchain/

Ether. (2018). The crypto-fuel for the Ethereum network. Recuperado de https://www.ethereum.org/ether

Gajendra, S. (2014). *A Hybrid Best-So-Far Artificial Bee Colony Algorithm*, Germany: Lambert Academic Publishing.

Gisbert, J. (2016a). *Vivir sin empleo: Trueque, bancos de tiempo, monedas sociales y otras alternativas*. Los libros del lince, Kindle Edition.

Gisbert, J. (2016b). Introducción a las monedas locales, sociales y complementarias. *I Ecuentro Internacional de Monedas Sociales, Complementarias y/o Virtuales*. Proyecto Vircoin2SME-Universidad Autónoma de Bucaramanga, Bucaramanga, Colombia. Recuperado de: http://gea.unab.edu.co:8080/exist/Congresos/JornadasMonedas/pdf/JulioGisbert.pdf

Golang. (s.f.) Proyecto Golang. Recuperado de https://golang.org/doc/

Graydon, C. (2014). What is cryptocurrency?. Cryptocoins News CCN.LA. Recuperado de: https://www.cryptocoinsnews.com/cryptocurrency/

Hayes, A. (2015). What Factors Give Cryptocurrencies Their Value: An Empirical Analysis. Working Papers, No 1406, New School for Social Research, Department of Economics.

Hirota, Y. (2012). Monedas complementarias como herramienta para fortalecer la economía social. Biblioteca Asocam. Recuperado de: http://www.asocam.org/biblioteca/items/show/1789

Hirota, Y. (2014). Panorama de monedas sociales en España. El País. Recuperado de: http://blogs.elpais.com/alterconsumismo/2014/09/panorama-de-monedas-sociales-en-espana.html

Hirota, Y. (2016). Monedas sociales y complementarias. *Oikonomics - Revista de Economía, Empresa y Sociedad.* Volumen (6), 35-42.

Holt, R.C., and Cordy, J.R. (1988). The Turing Programming Language. *Communications of the ACM*, 31(12), 1410-1423.

HRZONE. (2015). Interactive Community for Digital Media. Recuperado de: http://www.hrzone.com/hr-glossary/what-is-social-currency.

Khatwani, S. (2018). What is Proof-Of-Work & Proof-Of-Stake?. COINSUTRA. Recuperado de https://coinsutra.com/proof-of-work-vs-proof-of-stake-pow-vs-pos/

Kennedy M. y Kennedy D. (1998). *Dinero sin inflación ni tasas de interés*. Argentina:Editorial Nuevo Extremo.

King, S., and Nadal, S. (2012). PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake. White paper, recuperado de https://peercoin.net/assets/paper/peercoin-paper.pdf

Kirk, J. (2013). Could the Bitcoin network be used as an ultrasecure notary service?. IDG News, COMPUTERWORLD. Recuperado de https://www.computerworld.com/article/2498077/desktop-apps/could-the-Bitcoin-network-be-used-as-an-ultrasecure-notary-service-.html

Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, 4(3), 382–401. doi:10.1145/357172.357176

Lang, P. (2004). *Lets work: rebuilding the local economy*. Bristol:Grover

LANZAROTE. (2013). La moneda social: instrumento de desarrollo local y sostenible. Material de Trabajo, Reserva de la Biosfera de Lanzarote. Recuperado de: http://www.lanzarotebiosfera.org/wp-content/uploads/2014/02/Preguntas-y-respuestas-sobre-la-Moneda-Social.pdf

Lee, D. (2015). *Handbook of Digital Currency: Bitcoin, Innovation, Financial Instruments and Big Data*. London: Elsevier.

Li, W., Andreina, S., Bohli, JM., Karame, G. (2017). Securing Proof-of-Stake Blockchain Protocols. In: Garcia-Alfaro J., Navarro-Arribas G., Hartenstein H., Herrera-Joancomartí J. (eds) Data Privacy Management, Cryptocurrencies and Blockchain Technology. DPM 2017, CBT 2017. *Lecture Notes in Computer Science*, vol 10436, Springer, Cham.

Litecoin. (2018). Recuperado de: https://litecoin.info/index.php/Main_Page

Lietaer, B. (2001). *The future of money: Creating new wealth, work and a wiser world*. London: The Random House Group.

Lietaer, B. y Belgin, S. (2012). *New Money for a new world*. Boulder:Qiterra Press.

Lietaer, B., Arnsperger, C., Goerner, S., and Brunnhuber, S. (2012). *Money and Sustainability: The Missing Ring*. Devon:Triarchy Press.

Marvin, R. (2018). ConsenSys, Deloitte, and SAP Tip Enterprise Blockchain Moves. *PC Magazine.* Recuperado de https://www.pcmag.com/news/361209/consensys-deloitte-and-sap-tip-enterprise-blockchain-moves

Mattila, J. (2016). The blockchain phenomenon: The Disruptive Potential of Distributed Consensus Architectures, Working Paper, Berkeley Roundtable on the International Cconomy (BRIE), University of California, Berkeley. Recuperado de: https://brie.berkeley.edu/sites/default/files/juri-mattila-.pdf

Mochón, F. y Beker, V. (2008). *Economía, Principios y Aplicaciones*, Mexico: Mc Graw Hill.

Mougayar, W. (2016). *The business Blockchain: promise, practice and application of the next Internet technology*, New Jersey: John Wiley & Sons.

Nakamoto, S. (2005) Bitcoin: A Peer-to-Peer Electronic Cash System. Recuperado de https://Bitcoin.org/Bitcoin.pdf

Osterwalder, A., & Pigneur, Y. (2010). *Business Model Generation*, New Jersey: John Wiley & Sons.

PeerCoin. (2015). The secure and sustainable cryptocoin. Recuperado de https://peercoin.net/index.php?locale=es

Popov, S. (2018). The Tangle. Recuperado de https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf

Preukschat, A. (2017). Blockchain: la revolución industrial de Internet. Madrid:Planeta.

Reddcoin. (2015). Moneda social reddcoin. Recuperado de https://reddcoin.com/

Ren, L. (2014). Proof of Stake Velocity: Building the Social Currency of the Digital Age. *Crypto academic research papers*. Recuperado de https://www.Bitcoinpapers.com/crypto-research-package/reddcoin-proof-stake-velocity-whitepaper/

Remon, S. (2011). WIR o el modelo de antibanco suizo: otro sistema monetario puede ser posible. El Confidencial. Recuperado de: http://www.elconfidencial.com/economia/2011-06-19/wir-o-el-modelo-de-antibanco-suizo-otro-sistema-monetario-puede-ser-posible_396199/

Resico, M. (2010). *Introducción a la Economía Social de Mercado*. Berlin: Edición Latinoamericana Konrad AdenauerStiftung.

Rogers, J. (2011). On the money: Getting the message out. *International Journal of Community Currency Research IJCCR*, 15, Special Issue, 5-16.

Rosson, M. y Carroll, J. (2001). *Usability engineering: scenario-based development of human-computer interaction*. San Francisco: Morgan Kaufmann.

Saqib, M. y Saake, G. (2015). Merkle Hash Tree based Techniques for Data Integrity of Outsourced Data, 27th GI-Workshop on Foundations of Databases (Grundlagen von Datenbanken), Mayo 26 al 29 de 2015, Magdeburg, Germany.

Scott, J. and Green, K. (2013). Effects of Corporate Social Responsibility and Irresponsibility Policies, *Journal of Business Research*, 66(10), 1922-1927.

Shimpeno, P. and Ezeer, N. (2014). Improving the user interface through Gestalt design principles, The 26th Annual IEEE Software Technology Conference, Westin Long Beach, CA, 29 Mar – 03 Apr 2014, *IEEE computer society*, 1-63.

Seymour, D., Everhart, D., Yoshino, K. (2013). *Currency of higher education: credits and competencies*, American Council on Education & Blackboard.

Solidity. (s.f.). Lenguaje de programación Solidity para Contratos inteligentes. Recuperado de https://miethereum.com/smart-contracts/solidity/

Sousa, J., Bessani, A., and Vukolic, M. (2018). A Byzantine Fault-Tolerant Ordering Service for the Hyperledger Fabric Blockchain Platform, Cornell University Library. Recuperado de https://arxiv.org/abs/1709.06921

Swan, M. (2015). *Blockchain: Blueprint for a New Economy.* Sebastopol, California:O'Reilly

Trew, C., Brandon, G., and Dorier, N. (s.f.). Stratis:blockchain solutions whitepaper. Recuperado de https://stratisplatform.com/files/Stratis_Whitepaper.pdf

Stodder, J.P. (2009). Residual Barter Networks and Macro-Economic Stability: Switzerland's Wirtschaftsring, Journal of Economic Behavior & Organization, Elsevier, vol. 72(1), 79-95.

Vyas, C. and Lunagaria, M. (2014). Security Concerns and Issues for Bitcoin. *International Journal of Computer Applications IJCA*. 10-12.

Vivaldi Partners. (2012). *How brands and businesses can prosper in a digitally connected world*. New York:Vivaldi Inc.

Vivaldi Partners. (2013). *Social Currency in the B2B World: Building Strong Brands*. New York:Vivaldi Inc.

Wang, Y., and Mainwaring, S.D. (2008). Human-currency interaction: learning from virtual currency use in China, *Conference on Human Factors in Computing Systems*, Florence, Italy.

Weizsäcker, J. (2016). Proyecto de Informe sobre Monedas Virtuales, Comisión de Asuntos Económicos y Monetarios, Parlamento Europeo. Recuperado de http://www.europarl.europa.eu/meetdocs/2014_2019/plmrep/COMMITTEES/ECON/PR/2016/04-25/1083809ES.pdf

Wood, G. (s.f.). Gavin Wood - Ethereum founder and free-trust technologist. Recuperado de http://gavwood.com/

Wood, G. (2014). Ethereum: A secure descentralised generalised transaction ledger- EIP-150 revision. Recuperado de https://www.ethereum.org/.

Zwanenburg, J. (2018). Consensus Algorithms, Explained: What You Need To Know About Proof-Of-Work, Proof-Of-Stake, And Delegated Proof-Of-Stake. *INVEST IN Blockchain*. Recuperado de https://www.investinblockchain.com/consensus-algorithms-explained/

ANEXOS

ANEXO 1. CÓDIGO DEL CONTRATO *EURAKOS* EN EURAKOSNEXT

```
contract Eurakos {
    /* owner */
    address owner;
    /* This creates an array with all balances */
    mapping (address => uint256) public balanceOf;
    address[] participants;
    […….]

    /* Initializes contract with initial supply tokens to the creator of the contract */
    function MyToken(uint256 initialSupply) {
        […..]
     }

    /* Send coins */
    function transfer(address _from, address _to, uint256 _value) {
        […..]

     }

    /* sendInitialBalance */
    function sendInitialBalance (address to, uint value) {
        […..]
     }

    /* function askReward */
    function askReward(address _to, uint256 reward) {
            […..]
     }

    /* function  getBalanceOf */
    function getBalanceOf(address addressNew) constant returns (uint256) {
            […..]
    }
```

```
    function kill() {
        [.....]
    }
}
```

ANEXO 2. CÓDIGO DEL CONTRATO *PRODUCT* EN EURAKOSNEXT

```
contract Product {
	/* owner */
	address owner;
	/* unit price */
	int256 public priceByUnit;
	/* units */
	string public units;
	/* quantity */
	uint256 public quantity;
	/* Init */
	event Init (int256 priceNew, string unitsNew);

	/* AddQuantity */
	event AddQuantity (uint256 quantityNew);

	/* RemoveQuantity */
	event RemoveQuantity (uint256 quantityNew);

	/* modifier onlyOwner */
	modifier onlyOwner {
		if (msg.sender != owner)
		throw;
		_;
	}

	/* init */
	function init(int256 priceNew, string unitsNew) onlyOwner {
	   owner = msg.sender;
	   priceByUnit = priceNew;
	   units = unitsNew;
	   Init(priceNew, unitsNew);
	}
```

```
        /* setQuantity */
        function setQuantity (uint256 quantityNew) onlyOwner {
                [….]
        }

        /* Method addQuantity*/
        function addQuantity (uint256 quantityNew) {
                [….]
        }

        /* removeQuantity */
        function removeQuantity (uint256 quantityNew) onlyOwner {
                [….]
        }

        /* kill */
        function kill() onlyOwner {
                [….]
        }
}
```

ANEXO 3. CÓDIGO DEL CONTRATO BASKET EN *EURAKOSNEXT*

```
contract Basket {
    /* owner */
    address public owner;
    /* valueToSpend */
    uint256 public valueToSpend;
    /* sector */
    string public sector;
    /* offers */
    mapping(address=>uint) public offers;
    /* chosenProvider */
    address public chosenProvider;
    /* Add Offer */
    event AddOffer (address provider, uint offer);
     /* choosen Offer */
    event SetChosenOffer (address provider);
    /* Init */
    event Init(string sectorNew, uint256 valueToSpendNew);
    /* ValidateBasket */
    event ValidateBasket(address basketAddr, bool result);

    /* modifier onlyOwner */
    modifier onlyOwner {
        if (msg.sender != owner)
            throw;
        _;
    }

    function Basket() {
        owner = msg.sender;
    }
```

```
/* init */
function init(string sectorNew, uint256 valueToSpendNew) onlyOwner {
   sector = sectorNew;
   valueToSpend = valueToSpendNew;
}

/* Method Add Offer */
function addOffer (address provider, uint offer) {
        [….]
}

/* Method getOffer */
function getOffer(address provider) constant returns(uint) {
        [….]
}

/* Method setChoosenOffer */
function setChoosenOffer (address provider) {
        [….]
}

/* function stringsEqual*/
function stringsEqual(string storage _a,
   string memory _b) internal returns (bool) {
        [….]
}

/* function isValidBasket */
function isValidBasket() returns (bool) {
   if(owner != 0 && stringsEqual(sector, "") && valueToSpend > 0 &&
      chosenProvider != 0) {
      ValidateBasket (this, true);
      return true;
   } else {
       ValidateBasket (this, false);
      return false;
   }
}
```

```
    /* Method kill */
    function kill() {
            [….]
     }
}
```

ANEXO 4. CÓDIGO DEL CONTRATO AGREEMENT EN *EURAKOSNEXT*

```
contract Agreement {
    struct Clause {
        string name;
        address appliedTo;
        string action;
    }

    /* Clauses */
    Clause[] clauses;

    /* Eurakos token. Then could be other coin */
    address token;
    /* Owner Id. This is the client */
    address owner;
    /* Provider Id */
    address provider;
    /* Value ToSpend */
    uint256 valueToSpend;
    /* Current Balance */
    uint256 totalBalance;
    /* Agreement Status */
    uint status;
    /* Validation Status: Blocked, Waiting. Started. Finished. */
    string public validationStatus;
    /* Deadline */
    string deadline;
    /* ClientActions */
    mapping(address=>uint256) clientActions;

    /* Modifier onlyOwner */
    modifier onlyOwner {
        if (msg.sender != owner) throw;
        _;
    }
```

```
/* function init */
function init(address tokenNew, address providerNew,  uint256 valueToSpendNew,
        string deadline) onlyOwner {
        token = tokenNew;
        provider = providerNew;
        valueToSpend = valueToSpendNew;
        totalBalance = valueToSpendNew;

        clauses.push(Clause("cancel", providerNew, "return"));
        clauses.push(Clause("finish", providerNew, "pay"));
        clauses.push(Clause("bad_service", providerNew,"return"));
        clauses.push(Clause("service_denial", providerNew,"return"));

        Init(tokenNew, providerNew, owner, valueToSpendNew,
       deadline);
}

/* function doClientPay */
function doClientPay(uint256 value) returns (bool) {
   if(token != 0 && msg.sender == owner) {
        Eurakos eurakos = Eurakos(token);
        if(value <= totalBalance) {
            /* event to register logs*/
           DoClientPay(value, true);
           AddClientAction(msg.sender, value);
           return true;
        } else {
            /* event to register logs*/
          DoClientPay(value, false);
          return true;
       }
   } else {
        /* event to register logs*/
        DoClientPay(value, false);
        return false;
   }
}
```

```
/* function addClientAction */
function addClientAction (address clientAddr, uint256 value) {
        [….]
}

/* function cancelAgreement */
function cancelAgreement(string reason) returns (bool) {
   if(token != 0 && totalBalance != 0 && valueToSpend != 0 &&
        totalBalance <= valueToSpend) {
        Eurakos eurakos = Eurakos(token);
        eurakos.transfer(this, owner, totalBalance);

        /* clauses[0].name;*/
        CancelAgreement (reason, true);
        return true;
   } else {
        CancelAgreement (reason, false);
        return false;
   }
}

 /* function validateContent */
function isValidContent() returns (bool) {
   if(owner != 0 && provider != 0 && valueToSpend != 0 && totalBalance != 0 &&
        totalBalance <= valueToSpend) {

        ValidateAgreement(this, true);
        return true;
   } else {
        ValidateAgreement(this, false);
        return false;
   }
}
```

```
    /* function finishAgreement */
    function finishAgreement() returns (bool) {
            if(token != 0 && totalBalance > 0 && valueToSpend > 0 &&
            totalBalance <= valueToSpend) {
            Eurakos eurakos = Eurakos(token);
            eurakos.transfer (this, owner, totalBalance);
            /* clauses[0].name;*/
            FinishAgreement (true);
            return true;
        } else {
            FinishAgreement (false);
            return false;
        }
    }
}
```

ANEXO 5. CÓDIGO DEL CONTRATO *SWARMVALIDATORS* PARA POSB

```
contract SwarmValidators {
    /* Token Addr */
    address tokenAddr;
    /* Scoutt Counter */
    uint scoutCounter;
    /* Worker Counter */
    uint workerCounter;
    /* Onlooker Counter */
    uint onlookerCounter;

    struct Validator {
        /* validator  Address */
        address validAddr;
        uint validType;
        uint status;
    }

    Validator[] validatorList;
    /* elements To Validate. 1:basket,2:Agreement,3:clause */
    mapping (address=>uint) elementsToValidate;

    /* Method init */
    function init(address tokenAddrNew) {
        tokenAddr = tokenAddrNew;
        scoutCounter = 0;
        workerCounter = 0;
        onlookerCounter = 0;
    }

    /*Init*/
    event Init (address tokenAddrNew);
    /* function Addvalidator*/
    event Addvalidator (address validatorNew);
```

```
/* Method add validator */
function addvalidator (address validatorNew) {
   if(scoutCounter == 0) {
      /*New and free*/
      validatorList.push (Validator (validatorNew, 1, 1));
      scoutCounter++;
   } else if (workerCounter == 0) {
      validatorList.push(Validator(validatorNew, 2, 1));
      workerCounter++;
   } else if (onlookerCounter == 0) {
      /*New and free*/
      validatorList.push(Validator(validatorNew, 3, 1));
      onlookerCounter++;

   } else {
      if (workerCounter <= scoutCounter) {
         validatorList.push(Validator(validatorNew, 2, 1));
         workerCounter++;
      } else if (workerCounter <= onlookerCounter) {
         validatorList.push(Validator(validatorNew, 3, 1));
         onlookerCounter++;
      }
   }
}

/* function stringsEqual */
function stringsEqual (string storage _a, string memory _b) internal
   returns (bool) {
       [….]
}

   /* function addElementsToValidate */
function addElementsToValidate (address element, uint elementType) {
   [….]
}
```

```
/* function changeValidatorStatus. status: free, busy*/
function changeValidatorStatus(address validator, uint256 status) {
   for(uint i = 0; i < validatorList.length; i++) {
      if(validatorList[i].validAddr == validator) {
         validatorList[i].status = status;
      }
   }
}

/* function validateAgreement */
function validateAgreement (address agreeAddr) returns (bool) {
   for(uint i = 0; i < validatorList.length; i++) {
      if (validatorList[i].validAddr==msg.sender && validatorList[i].validType == 3) {
      Agreement agreement = Agreement (agreeAddr);
      agreement.isValidContent();
      return true;

      } else {
         return false;
      }
   }
}

/* actionSelection */
function actionSelection (address validatorNew, address elementToValidate,
   uint elementTypeNew) {
   /* change to busy*/
   for(uint i=0; i<validatorList.length;i++) {
      if(validatorList[i].validAddr == validatorNew) {
          validatorList[i].status = 2;
      }
      if(validatorList[i].validType == 1) {
         scoutAction (i, elementToValidate, elementTypeNew);
          validatorList[i].status = 1;                    /* change to free again*/
      } else if(validatorList[i].validType == 2) {
         workerAction (i, elementToValidate, elementTypeNew);
         validatorList[i].status = 1;                    /* change to free again*/
      } else {
```

```
            onlookerAction(i, elementToValidate, elementTypeNew);
            validatorList[i].status = 1;                    /* change to free again*/
        }
    }
}

/* function Scout Action. Add new element to validate. */
function scoutAction (uint validatorIndex, address elementToValidate,
    uint elementTypeNew) {
        [….]
 }

/* workerAction. Checks and assign */
function workerAction (uint validatorIndex, address elementToValidate,
    uint elementTypeNew) {
        [….]
    }

/* onlookerAction */
function onlookerAction (uint validatorIndex, address elementToValidate,
    uint elementTypeNew) {
        [….]
}

/* Returns a reward if belongs to swarm*/
function proofOfBelonging (uint validatorIndex) {
    Eurakos eurakos = Eurakos(tokenAddr);
    if (validatorList[validatorIndex].validType == 1) {
        eurakos.askReward (validatorList[validatorIndex].validAddr, 1);
    } else if (validatorList[validatorIndex].validType == 2) {
        eurakos.askReward (validatorList[validatorIndex].validAddr, 2);
    } else if (validatorList[validatorIndex].validType == 3) {
        eurakos.askReward (validatorList[validatorIndex].validAddr, 2);
    }
}
```

```
    /* get the validator role */
        function getValidatorRole(address validatorAddr) constant returns (uint) {
            [….]
}

    function isValidator(address validatorAddr) constant returns(bool) {
        for(uint i=0;i<validatorList.length;i++) {
            if(validatorList[i].validAddr == validatorAddr) {
                return true;
            }
        }
        return false;
    }

    /* function kill */
    function kill() {
        suicide(tokenAddr);
    }
}
```

ANEXO 6. PROTOCOLO DE CONSENSO PROOF OF WORK (POW)

Fuente: https://www.ethereum.org/token#proof-of-work

```
pragma solidity ^0.4.25;

contract PoW {
    bytes32 public currentChallenge;
    uint public timeOfLastProof;
    uint public difficulty = 10**32;
    mapping(address=>uint) balanceOf;

    function  proofOfWork(uint nonce) public {
        bytes8 n = bytes8(keccak256(nonce, currentChallenge));
        require(n > bytes8(difficulty));

        uint timeSinceLastProof = (now -timeOfLastProof);
        require (timeSinceLastProof >= 5 seconds);

        balanceOf[msg.sender] += timeSinceLastProof / 60 seconds;

        difficulty = difficulty* 10 minutes / timeSinceLastProof + 1;
        timeOfLastProof = now;
        currentChallenge = keccak256(nonce, currentChallenge, block.blockhash(block.number-1));
    }
}
```

ANEXO 7. PROTOCOLO DE CONSENSO PROOF OF AUTHORITY (POA)

```
pragma solidity ^0.4.25;

contract PoA {

    event InitiateChange(bytes32 indexed parentHash, address[] newSet);
    event ChangeFinalized(address[] newSet);
    event MoCInitializedProxyStorage(address proxyStorage);

    struct ValidatorState {
        // Is this a validator.
        bool isValidator;
        // Is a validator finalized.
        bool isValidatorFinalized;
        // Index in the currentValidators.
        uint256 index;
    }

    address public systemAddress = 0xffffFFFfFFffffffffffffffFfFFFfffFFFfFFfE;

    address[] public currentValidators;
    address[] public pendingList;
    mapping(address => ValidatorState) public validatorsState;

    address internal _moc;
    address internal _mocPending;
    address internal _owner;

    bool internal _isMoCRemoved = false;
    bool internal _isMoCRemovedPending = false;

    bool public finalized = false;
    bool public wasProxyStorageSet = false;
```

```
constructor(address _masterOfCeremony, address[] validators) public {
    // TODO: When you deploy this contract, make sure you hardcode items below
    // Make sure you have those addresses defined in spec.json
    require(_masterOfCeremony != address(0));
    _moc = _masterOfCeremony;
    currentValidators = [_masterOfCeremony];
    for (uint256 y = 0; y < validators.length; y++) {
        require(validators[y] != address(0));
        currentValidators.push(validators[y]);
    }
    for (uint256 i = 0; i < currentValidators.length; i++) {
        address validator = currentValidators[i];
        require(!isValidator(validator));
        validatorsState[validator] = ValidatorState({
            isValidator: true,
            isValidatorFinalized: true,
            index: i
        });
    }
    pendingList = currentValidators;
    _owner = msg.sender;
}

function isMasterOfCeremonyRemoved() public view returns(bool) {
    return _isMoCRemoved;
}

function isMasterOfCeremonyRemovedPending() public view returns(bool) {
    return _isMoCRemovedPending;
}

function masterOfCeremony() public view returns(address) {
    return _moc;
}

function masterOfCeremonyPending() public view returns(address) {
    return _mocPending;
}
```

```
/// Get current validator set (last enacted or initial if no changes ever made)
function getValidators() public view returns(address[]) {
    return currentValidators;
}

function getPendingList() public view returns(address[]) {
    return pendingList;
}

function addValidator(address _validator, bool _shouldFireEvent)
    public returns(bool) {
    if (_addValidatorAllowed(_validator)) {
        _addValidator(_validator, _shouldFireEvent);
        return true;
    }
    return false;
}

function removeValidator( address _validator, bool _shouldFireEvent)
          public returns(bool) {
    if (_removeValidatorAllowed(_validator)) {
        _removeValidator(_validator, _shouldFireEvent);
        return true;
    }
    return false;
}

function isValidator(address _someone) public view returns(bool) {
    return validatorsState[_someone].isValidator;
}

function isValidatorFinalized(address _someone) public view returns(bool) {
    bool _isValidator = validatorsState[_someone].isValidator;
    bool _isFinalized = validatorsState[_someone].isValidatorFinalized;
    return _isValidator && _isFinalized;
}
```

```
function getCurrentValidatorsLength() public view returns(uint256) {
    return currentValidators.length;
}

function getCurrentValidatorsLengthWithoutMoC() public view returns(uint256) {
    if (_isMoCRemoved) {
        return currentValidators.length;
    }
    if (currentValidators.length == 0) {
        return 0;
    }
    return currentValidators.length - 1; // exclude MoC
}

function _addValidatorAllowed(address _validator) private view returns(bool) {
    if (_validator == address(0)) return false;
    if (isValidator(_validator)) return false;
    return true;
}

function _addValidator(address _validator, bool _shouldFireEvent) private {
    validatorsState[_validator] = ValidatorState({
        isValidator: true,
        isValidatorFinalized: false,
        index: pendingList.length
    });
    pendingList.push(_validator);
    finalized = false;
    if (_shouldFireEvent) {
        emit InitiateChange(blockhash(block.number - 1), pendingList);
    }
}

function _removeValidatorAllowed(address _validator) private view returns(bool) {
    if (pendingList.length == 0) return false;
    if (!isValidator(_validator)) return false;
    return true;
}
```

```
    function _removeValidator(address _validator, bool _shouldFireEvent) private {
        uint256 removedIndex = validatorsState[_validator].index;
        // Can not remove the last validator.
        uint256 lastIndex = pendingList.length - 1;
        address lastValidator = pendingList[lastIndex];
        // Override the removed validator with the last one.
        pendingList[removedIndex] = lastValidator;
        // Update the index of the last validator.
        validatorsState[lastValidator].index = removedIndex;
        pendingList.length--;
        validatorsState[_validator].index = 0;
        validatorsState[_validator].isValidator = false;
        validatorsState[_validator].isValidatorFinalized = false;
        finalized = false;
        if (_shouldFireEvent) {
            if (_validator == _moc) {
                _isMoCRemovedPending = true;
            }
            emit InitiateChange(blockhash(block.number - 1), pendingList);
        }
    }

    function validate(bytes x) public view returns (bool) {
        if(isValidator(msg.sender)) {
            keccak256(x);
            return true;
        } else {
            return false;
        }
    }

}
```

ANEXO 8. PROTOCOLO DE CONSENSO PROOF OF REPUTATION V1 (PORV1)

```
pragma solidity ^0.4.25;

contract PoR {

    event InitiateChange(bytes32 indexed parentHash, address[] newSet);
    event ChangeFinalized(address[] newSet);

    bytes32 public currentChallenge;
    uint public timeOfLastProof;
    uint public difficulty = 10**32;
    mapping(address=>uint) balanceOf;


    struct ValidatorState {
        // Is this a validator.
        bool isValidator;
        // Is a validator finalized.
        bool isValidatorFinalized;
        // Index in the currentValidators.
        uint256 index;
    }

    address public systemAddress = 0xffffFFFfFFffffffffffffffFfFFFfffFFFfFFfE;

    address[] public currentValidators;
    address[] public pendingList;
    mapping(address => ValidatorState) public validatorsState;

    address internal _owner;

    bool public finalized = false;

    uint reputation;
```

```
constructor(address[] validators) public {
    // TODO: When you deploy this contract, make sure you hardcode items below
    // Make sure you have those addresses defined in spec.json

    for (uint256 y = 0; y < validators.length; y++) {
        require(validators[y] != address(0));
        currentValidators.push(validators[y]);
    }
    for (uint256 i = 0; i < currentValidators.length; i++) {
        address validator = currentValidators[i];
        require(!isValidator(validator));
        validatorsState[validator] = ValidatorState({
            isValidator: true,
            isValidatorFinalized: true,
            index: i
        });
    }
    pendingList = currentValidators;
    _owner = msg.sender;
    reputation = 0;
    poa_counter = 0;
}


function  validatePoW (bytes nonce) public {
    bytes8 n = bytes8(keccak256(nonce, currentChallenge));
    require(n > bytes8(difficulty));

    uint timeSinceLastProof = (now -timeOfLastProof);
    require (timeSinceLastProof >= 5 seconds);

    balanceOf[msg.sender] += timeSinceLastProof / 60 seconds;

    difficulty = difficulty* 10 minutes / timeSinceLastProof + 1;
    timeOfLastProof = now;
    currentChallenge = keccak256(nonce, currentChallenge, block.blockhash(block.number-1));
}
```

```
function validatePoA (bytes x) public view returns (bool) {
    if(isValidator(msg.sender)) {
        keccak256(x);
        return true;
    } else {
        return false;
    }
}

function validatePoR (bytes x) public returns (bool){
    if(isValidator(msg.sender)) {
        if(reputation <= 1) {
            validatePoW (x);
            reputation += 1;
        } else {
            validatePoA (x);
            reputation += 1;
        }
        return true;
    } else {
        return false;
    }
}

/// Get current validator set (last enacted or initial if no changes ever made)
function getValidators() public view returns(address[]) {
    return currentValidators;
}

function getPendingList() public view returns(address[]) {
    return pendingList;
}
```

```
function addValidator (address _validator, bool _shouldFireEvent)
    public returns(bool) {
    if (_addValidatorAllowed(_validator)) {
        _addValidator(_validator, _shouldFireEvent);
        return true;
    }
    return false;
}

function removeValidator (address _validator, bool _shouldFireEvent)
          public returns(bool) {
    if (_removeValidatorAllowed(_validator)) {
        _removeValidator(_validator, _shouldFireEvent);
        return true;
    }
    return false;
}

function isValidator (address _someone) public view returns(bool) {
    return validatorsState[_someone].isValidator;
}

function isValidatorFinalized (address _someone) public view returns(bool) {
    bool _isValidator = validatorsState[_someone].isValidator;
    bool _isFinalized = validatorsState[_someone].isValidatorFinalized;
    return _isValidator && _isFinalized;
}

function getCurrentValidatorsLength() public view returns(uint256) {
    return currentValidators.length;
}

function _addValidatorAllowed(address _validator) private view returns(bool) {
    if (_validator == address(0)) return false;
    if (isValidator(_validator)) return false;
    return true;
}
```

```
    function _addValidator(address _validator, bool _shouldFireEvent) private {
        validatorsState[_validator] = ValidatorState({
            isValidator: true,
            isValidatorFinalized: false,
            index: pendingList.length
        });
        pendingList.push(_validator);
        finalized = false;
        if (_shouldFireEvent) {
            emit InitiateChange(blockhash(block.number - 1), pendingList);
        }
    }

    function _removeValidatorAllowed(address _validator) private view returns(bool) {
        if (pendingList.length == 0) return false;
        if (!isValidator(_validator)) return false;
        return true;
    }

    function _removeValidator(address _validator, bool _shouldFireEvent) private {
        uint256 removedIndex = validatorsState[_validator].index;
        // Can not remove the last validator.
        uint256 lastIndex = pendingList.length - 1;
        address lastValidator = pendingList[lastIndex];
        // Override the removed validator with the last one.
        pendingList[removedIndex] = lastValidator;
        // Update the index of the last validator.
        validatorsState[lastValidator].index = removedIndex;
        pendingList.length--;
        validatorsState[_validator].index = 0;
        validatorsState[_validator].isValidator = false;
        validatorsState[_validator].isValidatorFinalized = false;
        finalized = false;
        if (_shouldFireEvent) {
            emit InitiateChange(blockhash(block.number - 1), pendingList);
        }
    }

}
```

ANEXO 9. PROTOCOLO DE CONSENSO PROOF OF REPUTATION V2 (PORV2)

```
pragma solidity ^0.4.25;

contract PoR {

    event InitiateChange(bytes32 indexed parentHash, address[] newSet);
    event ChangeFinalized(address[] newSet);

    bytes32 public currentChallenge;
    uint public timeOfLastProof;
    uint public difficulty = 10**32;
    mapping(address=>uint) balanceOf;


    struct ValidatorState {
        // Is this a validator.
        bool isValidator;
        // Is a validator finalized.
        bool isValidatorFinalized;
        // Index in the currentValidators.
        uint256 index;
    }

    address public systemAddress = 0xffffFFFfFFffffffffffffffFfFFFfffFFFfFFfE;

    address[] public currentValidators;
    address[] public pendingList;
    mapping(address => ValidatorState) public validatorsState;

    address internal _owner;

    bool public finalized = false;
```

```
uint reputation;
uint pow_init = 10;
uint poa_counter;
uint pow_counter;

constructor(address[] validators) public {
    // TODO: When you deploy this contract, make sure you hardcode items below
    // Make sure you have those addresses defined in spec.json

    for (uint256 y = 0; y < validators.length; y++) {
        require(validators[y] != address(0));
        currentValidators.push(validators[y]);
    }
    for (uint256 i = 0; i < currentValidators.length; i++) {
        address validator = currentValidators[i];
        require(!isValidator(validator));
        validatorsState[validator] = ValidatorState({
            isValidator: true,
            isValidatorFinalized: true,
            index: i
        });
    }
    pendingList = currentValidators;
    _owner = msg.sender;
    reputation = 0;
    poa_counter = 0;
}
```

```
function  validatePoW (bytes nonce) public {
    bytes8 n = bytes8(keccak256(nonce, currentChallenge));
    require(n > bytes8(difficulty));

    uint timeSinceLastProof = (now -timeOfLastProof);
    require (timeSinceLastProof >= 5 seconds);

    balanceOf[msg.sender] += timeSinceLastProof / 60 seconds;

    difficulty = difficulty* 10 minutes / timeSinceLastProof + 1;
    timeOfLastProof = now;
    currentChallenge = keccak256(nonce, currentChallenge, block.blockhash(block.number-1));
}

function validatePoA (bytes x) public view returns (bool) {
    if(isValidator(msg.sender)) {
        keccak256(x);
        return true;
    } else {
        return false;
    }
}

function validatePoR (bytes x) public returns (bool){
    if(isValidator(msg.sender)) {
        if(reputation <= pow_init) {
            validatePoW (x);
            pow_counter += 1;
            reputation += 1;
        } else {
            pow_counter = 0;

            validatePoA (x);
            reputation += 1;
        }
```

```
        return true;
    } else {
        return false;
    }
}

/// Get current validator set (last enacted or initial if no changes ever made)
function getValidators() public view returns(address[]) {
    return currentValidators;
}

function getPendingList() public view returns(address[]) {
    return pendingList;
}

function addValidator (address _validator, bool _shouldFireEvent)
    public returns(bool) {
    if (_addValidatorAllowed(_validator)) {
        _addValidator(_validator, _shouldFireEvent);
        return true;
    }
    return false;
}

function removeValidator (address _validator, bool _shouldFireEvent)
          public returns(bool) {
    if (_removeValidatorAllowed(_validator)) {
        _removeValidator(_validator, _shouldFireEvent);
        return true;
    }
    return false;
}

function isValidator (address _someone) public view returns(bool) {
    return validatorsState[_someone].isValidator;
}
```

```
function isValidatorFinalized (address _someone) public view returns(bool) {
    bool _isValidator = validatorsState[_someone].isValidator;
    bool _isFinalized = validatorsState[_someone].isValidatorFinalized;
    return _isValidator && _isFinalized;
}

function getCurrentValidatorsLength() public view returns(uint256) {
    return currentValidators.length;
}

function _addValidatorAllowed(address _validator) private view returns(bool) {
    if (_validator == address(0)) return false;
    if (isValidator(_validator)) return false;
    return true;
}

function _addValidator(address _validator, bool _shouldFireEvent) private {
    validatorsState[_validator] = ValidatorState({
        isValidator: true,
        isValidatorFinalized: false,
        index: pendingList.length
    });
    pendingList.push(_validator);
    finalized = false;
    if (_shouldFireEvent) {
        emit InitiateChange(blockhash(block.number - 1), pendingList);
    }
}

function _removeValidatorAllowed(address _validator) private view returns(bool) {
    if (pendingList.length == 0) return false;
    if (!isValidator(_validator)) return false;
    return true;
}
```

```
    function _removeValidator(address _validator, bool _shouldFireEvent) private {
        uint256 removedIndex = validatorsState[_validator].index;
        // Can not remove the last validator.
        uint256 lastIndex = pendingList.length - 1;
        address lastValidator = pendingList[lastIndex];
        // Override the removed validator with the last one.
        pendingList[removedIndex] = lastValidator;
        // Update the index of the last validator.
        validatorsState[lastValidator].index = removedIndex;
        pendingList.length--;
        validatorsState[_validator].index = 0;
        validatorsState[_validator].isValidator = false;
        validatorsState[_validator].isValidatorFinalized = false;
        finalized = false;
        if (_shouldFireEvent) {
            emit InitiateChange(blockhash(block.number - 1), pendingList);
        }
    }

}
```

ANEXO 10. PROTOCOLO DE CONSENSO PROOF OF REPUTATION V3 (PORV3)

```
pragma solidity ^0.4.25;

contract PoR {

    event InitiateChange(bytes32 indexed parentHash, address[] newSet);
    event ChangeFinalized(address[] newSet);

    bytes32 public currentChallenge;
    uint public timeOfLastProof;
    uint public difficulty = 10**32;
    mapping(address=>uint) balanceOf;


    struct ValidatorState {
        // Is this a validator.
        bool isValidator;
        // Is a validator finalized.
        bool isValidatorFinalized;
        // Index in the currentValidators.
        uint256 index;
    }

    address public systemAddress = 0xffffFFFfFFffffffffffffffFfFFFfffFFFfFFfE;

    address[] public currentValidators;
    address[] public pendingList;
    mapping(address => ValidatorState) public validatorsState;

    address internal _owner;

    bool public finalized = false;
```

```
uint reputation;
uint pow_init = 10;
uint pow_mod = 100;
uint poa_counter;
uint pow_counter;

constructor(address[] validators) public {
    // TODO: When you deploy this contract, make sure you hardcode items below
    // Make sure you have those addresses defined in spec.json

    for (uint256 y = 0; y < validators.length; y++) {
        require(validators[y] != address(0));
        currentValidators.push(validators[y]);
    }
    for (uint256 i = 0; i < currentValidators.length; i++) {
        address validator = currentValidators[i];
        require(!isValidator(validator));
        validatorsState[validator] = ValidatorState({
            isValidator: true,
            isValidatorFinalized: true,
            index: i
        });
    }
    pendingList = currentValidators;
    _owner = msg.sender;
    reputation = 0;
    poa_counter = 0;
    pow_counter = 0;
}
```

```
function  validatePoW (bytes nonce) public {
    bytes8 n = bytes8(keccak256(nonce, currentChallenge));
    require(n > bytes8(difficulty));

    uint timeSinceLastProof = (now -timeOfLastProof);
    require (timeSinceLastProof >= 5 seconds);

    balanceOf[msg.sender] += timeSinceLastProof / 60 seconds;

    difficulty = difficulty* 10 minutes / timeSinceLastProof + 1;
    timeOfLastProof = now;
    currentChallenge = keccak256(nonce, currentChallenge, block.blockhash(block.number-1));
}

function validatePoA (bytes x) public view returns (bool) {
    if(isValidator(msg.sender)) {
        keccak256(x);
        return true;
    } else {
        return false;
    }
}

function validatePoR (bytes x) public returns (bool){
    if(isValidator(msg.sender)) {
        if(reputation <= pow_init) {
            validatePoW (x);
            pow_counter += 1;
            reputation += 1;
        } else {
            pow_counter = 0;

            if (poa_counter != 100) {
                poa_counter += 1;
```

```
                validatePoA (x);
                reputation += 1;
            } else {
                poa_counter = 0;
                reputation += 1;
                validatePoW (x);
            }
        }
        return true;
    } else {
        return false;
    }
}

/// Get current validator set (last enacted or initial if no changes ever made)
function getValidators() public view returns(address[]) {
    return currentValidators;
}

function getPendingList() public view returns(address[]) {
    return pendingList;
}

function addValidator (address _validator, bool _shouldFireEvent)
    public returns(bool) {
    if (_addValidatorAllowed(_validator)) {
        _addValidator(_validator, _shouldFireEvent);
        return true;
    }
    return false;
}
```

```
function removeValidator (address _validator, bool _shouldFireEvent)
        public returns(bool) {
    if (_removeValidatorAllowed(_validator)) {
        _removeValidator(_validator, _shouldFireEvent);
        return true;
    }
    return false;
}

function isValidator (address _someone) public view returns(bool) {
    return validatorsState[_someone].isValidator;
}

function isValidatorFinalized (address _someone) public view returns(bool) {
    bool _isValidator = validatorsState[_someone].isValidator;
    bool _isFinalized = validatorsState[_someone].isValidatorFinalized;
    return _isValidator && _isFinalized;
}

function getCurrentValidatorsLength() public view returns(uint256) {
    return currentValidators.length;
}

function _addValidatorAllowed(address _validator) private view returns(bool) {
    if (_validator == address(0)) return false;
    if (isValidator(_validator)) return false;
    return true;
}
```

```
    function _addValidator(address _validator, bool _shouldFireEvent) private {
        validatorsState[_validator] = ValidatorState({
            isValidator: true,
            isValidatorFinalized: false,
            index: pendingList.length
        });
        pendingList.push(_validator);
        finalized = false;
        if (_shouldFireEvent) {
            emit InitiateChange(blockhash(block.number - 1), pendingList);
        }
    }

    function _removeValidatorAllowed(address _validator) private view returns(bool) {
        if (pendingList.length == 0) return false;
        if (!isValidator(_validator)) return false;
        return true;
    }
    function _removeValidator(address _validator, bool _shouldFireEvent) private {
        uint256 removedIndex = validatorsState[_validator].index;
        // Can not remove the last validator.
        uint256 lastIndex = pendingList.length - 1;
        address lastValidator = pendingList[lastIndex];
        // Override the removed validator with the last one.
        pendingList[removedIndex] = lastValidator;
        // Update the index of the last validator.
        validatorsState[lastValidator].index = removedIndex;
        pendingList.length--;
        validatorsState[_validator].index = 0;
        validatorsState[_validator].isValidator = false;
        validatorsState[_validator].isValidatorFinalized = false;
        finalized = false;
        if (_shouldFireEvent) {
            emit InitiateChange(blockhash(block.number - 1), pendingList);
        }
    }

}
```