ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

MÁSTER EN INGENIERÍA DE TELECOMUNICACIÓN

TECNOLOGÍA BLOCKCHAIN. ANÁLISIS, IMPLEMENTACIONES Y APLICACIÓN A OTROS PARADIGMAS

Autor: Javier Bustos Sánchez

Director: Atilano Fernández-Pacheco Sánchez-Migallón

Madrid

Junio 2018

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Tecnología Blockchain. Análisis, Implementaciones y Aplicación a otros Paradigmas

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2017/18 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.

Fdo.: **Javier Bustos Sánchez** Fecha: ……/ ……/ ……

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: **Atilano Fernández-Pacheco Sánchez-Migallón** Fecha: ……/ ……/ ……

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. **Javier Bustos Sánchez**

DECLARA ser el titular de los derechos de propiedad intelectual de la obra: **Tecnología Blockchain. Análisis, Implementaciones y Aplicación a otros Paradigmas**, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar "marcas de agua" o cualquier otro sistema de seguridad o de protección.
b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
e) Asignar por defecto a estos trabajos una licencia Creative Commons.
f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente).*

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

a) Que la Universidad identifique claramente su nombre como autor de la misma
b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
c) Solicitar la retirada de la obra del repositorio por causa justificada.
d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.
d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción

de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusive del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a …….. de ……... de …….

ACEPTA

Fdo……

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

MÁSTER EN INGENIERÍA DE TELECOMUNICACIÓN

TECNOLOGÍA BLOCKCHAIN. ANÁLISIS, IMPLEMENTACIONES Y APLICACIÓN A OTROS PARADIGMAS

Autor: Javier Bustos Sánchez

Director: Atilano Fernández-Pacheco Sánchez-Migallón

Madrid

Junio 2018

Agradecimientos

Principalmente, me gustaría expresar un fuerte agradecimiento a mis padres, por su enorme apoyo y confianza durante los seis años de mi paso por la escuela. Fueron los únicos en apostar por mí en momentos en los que ni yo mismo lo hacía. Gracias.

Además, dentro del contexto de la elaboración de este Trabajo de Fin de Máster, me gustaría agradecer la asistencia de dos personas fundamentalmente.

En primer lugar, agradecer a Atilano Fernández – Pacheco, director del proyecto, por sus ayudas y aportaciones durante el desarrollo a distancia del TFM.

Agradecer también a Francisco Javier Herraiz, coordinador del proyecto dentro de ICAI, por su seguimiento y su asistencia ante los imprevistos que han podido surgir durante el trascurso de la tesis.

Finally, I would like to thank Dr. Yuan An (Drexel University), for helping me better understand two complex technology paradigms (Blockchain and Semantic Web) and providing great ideas to improve my work. Thank you, professor.

TECNOLOGÍA BLOCKCHAIN. ANÁLISIS, IMPLEMENTACIONES Y APLICACIÓN A OTROS PARADIGMAS

Autor: Bustos Sánchez, Javier
Director: Fernández-Pacheco Sánchez-Migallón, Atilano
Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

La Blockchain promete ser uno de los paradigmas impulsores de la actual revolución tecnológica. En este proyecto se ha realizado una labor de investigación exhaustiva sobre la cadena de bloques, sus aspectos fundamentales y sus oportunidades de aplicabilidad. Además, se ha analizado en profundidad la red Ethereum y el proyecto Hyperledger, realizándose implementaciones de *Smart Contracts* y e introduciéndose algunas ideas iniciales sobre la aplicación de la cadena de bloques en sectores como la propiedad intelectual o el IoT. Por último, se ha analizado la colaboración de la Blockchain con la *Semantic Web*, integración que aportaría beneficios a ambos paradigmas.

Palabras clave: Blockchain, descentralización, interoperabilidad, Smart Contract, Ethereum, Hyperledger, Semantic Web, ontología, Linked Data.

1. Introducción

La situación mundial actual se caracteriza por una realidad totalmente digitalizada e interconectada, en la que la distribución y transferencia de datos y activos digitales no para de crecer. Es por ello por lo que surge la necesidad imperiosa de desarrollar *software* e infraestructuras que satisfagan estos requerimientos.

En los últimos años, numerosos productos y soluciones han irrumpido en el campo del almacenamiento y el procesado de datos: MongoDB, Hadoop, Cassandra, etc. son algunos de ellos. No obstante, a pesar de contar con evidentes beneficios y ser imprescindibles a día de hoy en multitud de entornos, este tipo de soluciones sigue presentando riesgos en términos de tolerancia a fallos y centralización, entre otros aspectos.

En este panorama se sitúa el nacimiento de la Blockchain, tecnología que proporciona un registro distribuido, compartido y replicado en absolutamente todos los nodos participantes de cada red. Uno de los aspectos más potentes de este paradigma es el control de modificaciones y la verificación de los datos alojados en la cadena: lejos de contar con un control centralizado en el que el propietario de la Blockchain puede con toda libertad manipular los datos, los sistemas basados en esta tecnología cuentan con mecanismos de consenso que impiden la realización de modificaciones sobre datos y/o transacciones sin contar con el acuerdo del resto de participantes de la cadena.

Todo esto, junto con otros beneficios, hacen de la cadena de bloques la solución ideal para la instauración de lo que comúnmente se conoce como *Trust Economy*, un ecosistema en el que multitud de participantes comparten e intercambian datos y activos digitales sin que exista una confianza previa entre los mismos.

2. Definición de los objetivos del proyecto

En primer lugar, el presente trabajo busca realizar una labor de investigación general sobre el paradigma de la Blockchain, abarcando ésta sus fundamentos y los componentes principales de toda solución basada en el paradigma. Además, se busca presentar una descripción introductoria de los principales protocolos Blockchain.

El segundo objetivo está centrado en el análisis de la aplicabilidad de los distintos aspectos investigados, mediante el estudio sobre algunas de las tecnologías Blockchain más empleadas actualmente y la posibilidad de integrar éstas en las infraestructuras de una serie de sectores empresariales.

Por último, se analizará la viabilidad de integración de la Blockchain con el paradigma de la Web Semántica mediante el empleo de los conocimientos adquiridos tras alcanzar los dos objetivos anteriores.

3. Descripción del trabajo realizado

Para la realización de todo el trabajo que constituye esta tesis, se ha decidido seguir aproximadamente el orden establecido por la definición de objetivos. Por tanto, en primer lugar se ha recorrido el paradigma de la cadena de bloques. *Grosso modo*, la Blockchain constituye una tecnología basada en un registro distribuido y compartido por todos los nodos de la red, contenedor de los distintos bloques y garante de principios fundamentales como la descentralización, la transparencia y la seguridad del dato. Estos bloques suelen tener una estructura común, contienen información sobre transacciones o lógica del sistema (*Smart Contracts*) y van siendo introducidos en la cadena en función del protocolo de consenso en cuestión.

El algoritmo de consenso permite alcanzar un acuerdo sobre la validez del dato y el nodo de la red encargado en introducir el próximo bloque en la Blockchain. Las bases de estos protocolos varían desde la resolución de un *puzzle* computacional complejo (PoS, minado) al establecimiento de tiempos aleatorios de espera (PoET). Cada implementación de Blockchain puede definir su propio protocolo de consenso, además de otros aspectos como la existencia o no de permisos.

Actualmente, existen numerosas redes o protocolos de la Blockchain, tales como la archiconocida Bitcoin, Ripple (transacciones globales de un modo económico) o Corda (sector financiero) [1]. En esta ocasión se ha optado por analizar la red Ethereum y el proyecto Hyperledger, con un alto grado de adopción en el panorama empresarial actual.

Por un lado, la red Ethereum, mayoritariamente pública, ofrece un gran nivel de versatilidad gracias a que ofrece la posibilidad de ejecución de lógica dentro de lo que se conoce como Dapps (aplicaciones descentralizadas). Esto es posible gracias a los *Smart Contract*, piezas de *software* que señalan las condiciones lógicas existentes en los procesos de ejecución o verificación de transacción de activos, siendo estos expresados en Ether (criptomoneda nativa) o un nuevo *token* que se adapte a las necesidades del usuario.

Por otro lado, Hyperledger es un proyecto *open-source* iniciado por la Fundación Linux, que provee al usuario de *frameworks* y herramientas para simplificar la implantación de soluciones Blockchain mediante el empleo de estándares y piezas modulares y reutilizables. En concreto, se ha analizado uno de sus proyectos más ambiciosos: Hyperledger Sawtooth.

El análisis conceptual y práctico realizado sobre el paradigma ha permitido extraer un "procedimiento estándar" que define las distintas tomas de decisiones a la hora de implantar una solución basada en Blockchain. Este flujo de decisiones aparece reflejado en la siguiente figura:

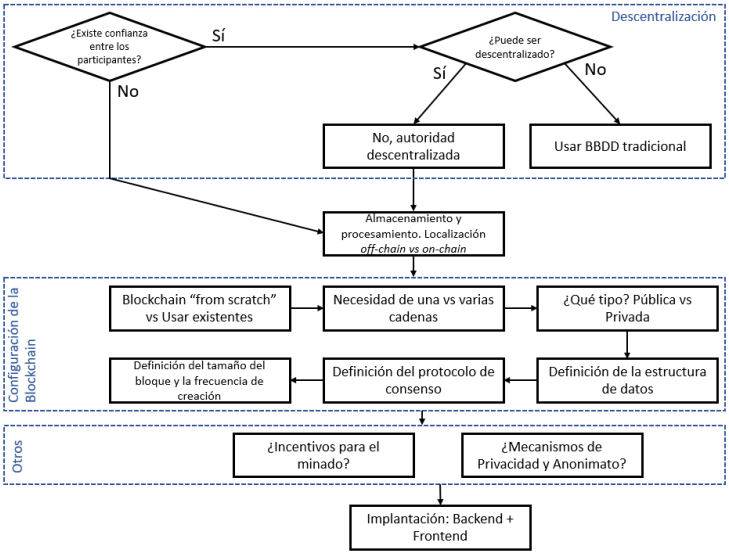


Ilustración 1. Implantación de soluciones en Blockchain. Alternativas disponibles

Finalmente, se ha analizado la posibilidad de integración de la Blockchain en diversos campos industriales y paradigmas, profundizándose especialmente en el establecimiento de una relación simbiótica entre ésta y la Web Semántica. La *Semantic Web* y el *Linked Data* defienden la interconexión de toda la WWW en términos de significados. Es decir, se pretende interconectar todos los datos presentes en la web mediante el uso de metadatos y estándares basados en RDF [2], un *framework* que refleja las relaciones y propiedades de recursos webs mediante el uso de grafos. Fundamentalmente, la relación simbiótica antes mencionada puede realizarse de tres maneras:

a) Mapeando el contenido interno de la Blockchain empleando estándares semánticos, lo que trae beneficios tales como la reutilización y la simplificación del aprendizaje.
b) Almacenando contenido RDF directamente en la Blockchain, lo que aseguraría una descentralización de la gestión de estos recursos y una protección de los datos ante manipulaciones indeseadas.
c) Creando Blockchains "*from scratch"* basadas en estándares semánticos.

4. Aplicaciones propuestas

Paralelamente a la investigación y el análisis descritos, se han empleado los conocimientos adquiridos para tener una primera toma de contacto con la Blockchain en términos prácticos.

En primer lugar, dentro del trabajo realizado en relación con la implementación y ejecución de *Smart Contracts* sobre la red de prueba de Ethereum, cabe destacar la implementación propia de una criptomoneda o *token*: Icaicoin (ICC) y su proceso de *testing* empleando la aplicación Metamask.

Además de lo anterior, se han realizado una serie de modelos de soluciones basadas en Blockchain. Cabe destacar la presentación de la arquitectura completa de una plataforma

de control de la propiedad intelectual mediante el uso de Hyperledger Sawtooth [3], o de un modelo inicial de la integración Blockchain – Semantic Web ya mencionada. El esquema de esta última propuesta se puede observar en la siguiente ilustración:

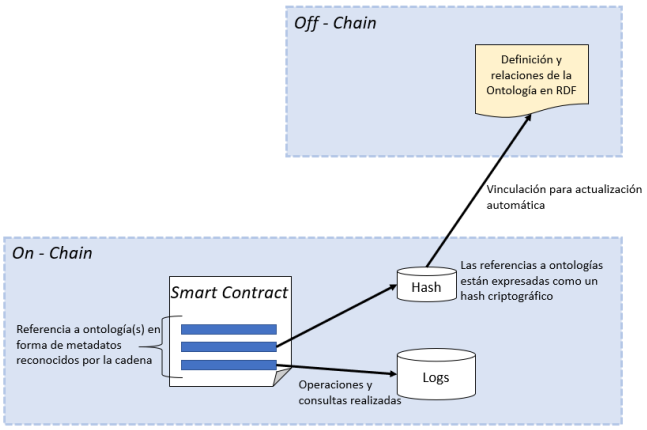


Ilustración 2. Integración de Ontologías y Blockchain. Modelo Inicial

5. Conclusiones

En definitiva, la Blockchain se consolida como un paradigma crucial en el panorama tecnológico actual, caracterizado por la interoperabilidad global y la necesidad de mecanismos garantes de confianza que aporten transparencia y no comprometan la integridad de los datos de los partidos involucrados. No obstante, es necesario gestionar los desafíos a los que se enfrenta el paradigma (ausencia de estandarización, controversias regulatorias) con el fin de agilizar su adopción en el sector empresarial.

6. Referencias

[1] Stanford Computer Science. (n.d.). Digital Currencies. Retrieved from https://cs.stanford.edu/people/eroberts/cs181/projects/2010-11/DigitalCurrencies/disadvantages/index.html

[2] English, M., Auer, S., & Domingue, J. (n.d.). *Blockchain Technologies & The Semantic Web: A Framework for Symbiotic Development* (Unpublished master's thesis). University of Bonn. Retrieved from http://cscubs.cs.uni-bonn.de/2016/proceedings/paper-10.pdf

[3] Hyperledger Technical Steering and Marketing Committees (Ed.). (n.d.). *Hyperledger. The Whitepaper*. Linux Foundation.

THE BLOCKCHAIN. ANALYSIS, TECHNOLOGIES AND INTEGRATION WITH OTHER PARADIGMS

Author: Bustos Sánchez, Javier
Supervisor: Fernández-Pacheco Sánchez-Migallón, Atilano
Collaborating Entity: ICAI – Comillas Pontifical University

ABSTRACT

Blockchain will probably become one of the main enablers of the current technology revolution. In this thesis, an initial research has been undertaken, covering the Blockchain's core principles and specifying several deployment opportunities. Also, the Ethereum network and the Hyperledger project have been analyzed by programming *Smart Contracts* and introducing the use of Blockchain to solve different scenarios within sectors such as the IoT and the intellectual property. Finally, several ideas have been pointed out regarding a symbiotic collaboration between Blockchain and the *Semantic Web.*

Keywords: Blockchain, decentralization, interoperability, Smart Contract, Ethereum, Hyperledger, Semantic Web, ontology, Linked Data.

1. Introduction

Nowadays, global situation can be defined as a digital and interconnected reality, in which data or digital assets transfer and distribution is a non-stop process. That's why there is a huge need for high quality *software* and infrastructure that meet these requirements.

Lately, lots of different products and solutions have disrupted the data storage and processing fields: MongoDB, Hadoop and Cassandra are some of these. However, despite having many benefits, this sort of solutions presents challenges and risks in terms of fault-tolerance and centralization.

Blockchain aims to solve all these problems. It consists in a distributed and shared ledger, replicated among all the network nodes. One of the most powerful features in Blockchain is the consensus protocol: instead of easing data modification and/or manipulation, this mechanism forces a collective agreement and verification before any changes approval and incorporation in the network.

These are some of the benefits that turn Blockchain into the optimal alternative to establish the *Trust Economy*, an operations framework in which lots of parties can share and exchange data and/or digital assets without necessarily trusting each other.

2. Thesis Objectives

First, this project aims to undertake a general research about the Blockchain, covering aspects such as its main design principles and the core elements that can be found in a Blockchain-based system. Also, an introduction about the different Blockchain protocols will be given.

Second, the latter research will be used to analyze how suitable Blockchain is for different applications. To achieve this, some of the most used technologies and projects will be studied by undertaking development and modelling tasks.

Third and last, knowledge extracted from objective #1 and #2 will be used to describe a framework of symbiotic collaboration between Blockchain and Semantic Web.

3. Undergone work

The preparation of the thesis has followed the order that was previously established in the objectives section. First, the Blockchain has been researched in general terms. Overall, this paradigm is based on a distributed ledger that is shared among the network's participants, consists of different blocks and guarantees benefits such as decentralization, transparency and data security. This blocks usually share a common structure and include information regarding transactions and system logic (via Smart Contracts).

The blocks introduction in the Blockchain is orchestrated via the consensus protocol, which allows the system to reach a collective agreement on the data validation and the network node in charge of introducing the next block. This protocol's basis is different depending on the Blockchain: the resolution of a complex algorithmic puzzle (mining) in PoS, the definition of random waiting times in PoET, etc. Each Blockchain network has the capability of defining several functional aspects such as the consensus protocol or the permissions policy.

Nowadays, there are lots of different implementations of Blockchains. Some of them are the Bitcoin, Ripple (international wires) or Corda (financial sector) [1]. In particular, this thesis has analyzed two of the most used Blockchain solutions: the Ethereum network and the Hyperledger project.

On the one hand, Ethereum is a mostly public Blockchain, which provides high levels of flexibility due to the capability to run customized logic within Decentralized Applications, also known as Dapps. Smart Contracts are the key enabler of this internal logic and consist in software pieces that define the different logic and conditions that rule the digital asset's transactions. These assets can be expressed in terms of Ether (Ethereum's native cryptocurrency) or a new token that fits the user's needs.

On the other hand, Hyperledger is an open-source project initiated by the Linux Foundation. It provides a large set of tools and frameworks that help the user easily implement his/her Blockchain solutions by following design principles such as standardization, modular components and reutilization. In particular, an analysis has been undertaken regarding one of Hyperledger's most ambitious projects: Sawtooth.

After finishing the previously defined research and analysis tasks, a set of decision-making processes has been described regarding the deployment of a Blockchain-based solution. This decision's flow is presented below:

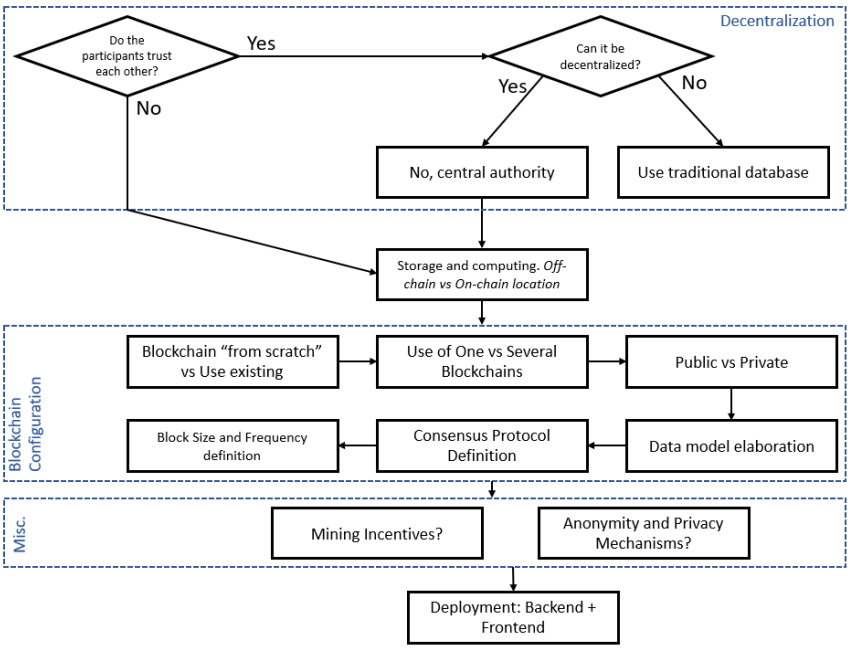


Figure 1. Blockchain deployments. Possible alternatives

Finally, the use of Blockchain for a diverse range of purposes has been analyzed. In particular, a symbiotic and collaborative relationship has been defined between Blockchain and Semantic Web. This latter paradigm, along with Linked Data, support the connection about every data within the WWW, mostly regarding meaning. That is, it aims to link every data in the web by using metadata and RDF-based standards [2]. RDF is a framework that describes the different relations and properties of a web resource, shaping these by using graphs.

This symbiotic relationship can be achieved by following three different strategies:

a) Map the Blockchain's internal content by using semantic standards, which would bring benefits in terms of reutilization and training simplification.

b) Store RDF-based content directly in the Blockchain, guaranteeing a decentralization and a data protection of all these resources.

c) Create Blockchains from scratch, basing these on semantic standards.

4. Proposed Models

Along with the undertaken analysis and research task, the acquired knowledge has been used to lightly familiarize with some Blockchain's practical aspects.

First, regarding the work done in terms of Smart Contracts implementation on the Ethereum test net (Ropsten), it is worth to high light the creation of a cryptocurrency or token: Icaicoin (ICC) and its testing process by using the Metamask application.

Also, a series of models describing Blockchain-based solutions has been presented. For example, a platform focused on Intellectual Property management has been devised by following the Hyperledger Sawtooth [3] structure and components.

Finally, a proposal for Blockchain – Semantic Web integration has been explained. Basically, it consists in linking the existing ontologies with a Blockchain, storing all the ontologies' structure *off-chain* and connecting to this by a hash repository located *on-chain*. This solution's high-level model is shown below:

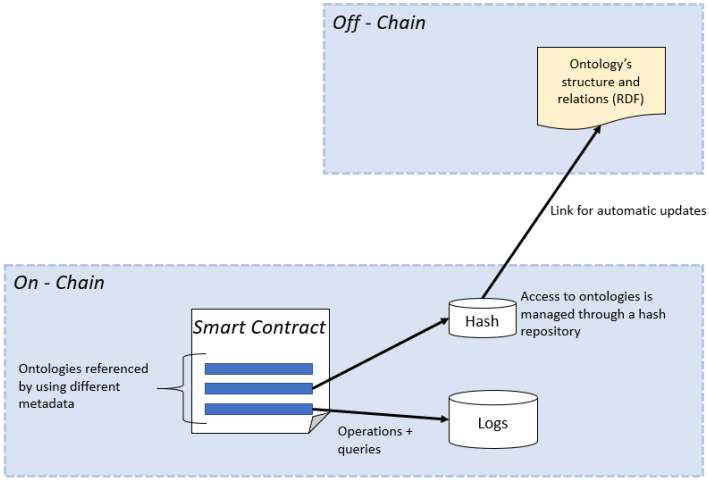


Figure 2. Blockchain and Semantic Web. Initial Model

5. Conclusions

Blockchain seems to be a crucial paradigm within the current technological situation, characterized by a global interoperability and an urgent need for trust and transparency mechanisms. However, some challenges (lack of standardization, regulatory controversies) still need to be addressed in order to speed up the technology's adoption within most of the industry fields.

6. References

[1] Stanford Computer Science. (n.d.). Digital Currencies. Retrieved from https://cs.stanford.edu/people/eroberts/cs181/projects/2010-11/DigitalCurrencies/disadvantages/index.html

[2] English, M., Auer, S., & Domingue, J. (n.d.). *Blockchain Technologies & The Semantic Web: A Framework for Symbiotic Development* (Unpublished master's thesis). University of Bonn. Retrieved from http://cscubs.cs.uni-bonn.de/2016/proceedings/paper-10.pdf

[3] Hyperledger Technical Steering and Marketing Committees (Ed.). (n.d.). *Hyperledger. The Whitepaper*. Linux Foundation.

Índice de la memoria

Índice de figuras

Índice de tablas

Capítulo 1. INTRODUCCIÓN

El presente Trabajo de Fin de Máster está centrado en el análisis y la aplicación de la tecnología Blockchain, paradigma que ha despertado el interés de multitud de compañías y profesionales. A pesar de que su representación más característica hasta la fecha es el Bitcoin y demás criptomonedas, Blockchain podría revolucionar las operaciones en numerosos campos, tales como la gestión de cadenas de producción o los sistemas electorales.

En primer lugar, se procederá a describir el estado del arte de esta tendencia tecnológica y se aportarán diversos aspectos formales, tales como la metodología de trabajo a emplear o la estimación temporal y económica del proyecto.

Tras esto, se procederá al estudio exhaustivo de la tecnología Blockchain, sus principios, tecnología e implementaciones llevadas a cabo hasta la fecha.

Finalmente, los últimos capítulos de la memoria abordaran la dimensión más práctica del proyecto, tratando de analizar la viabilidad de la implementación de Blockchain para dar respuesta a ciertos casos de uso, así como la posible combinación de esta tecnología con otros paradigmas tecnológicos como el Internet de las Cosas.

1.1 MOTIVACIÓN DEL PROYECTO

Blockchain promete constituir la vía hacia una revolución en la forma de hacer negocios y gestionar economías, según multitud de expertos. No obstante, para que esta tecnología sea asimilada y adoptada de un modo generalizado, se deben llevar a cabo numerosas modificaciones y/o cambios de perspectiva en términos de gobernabilidad, organización, perspectiva social, etc. Es precisamente esta situación de incertidumbre, esta necesidad de un cambio de paradigma, lo que hace de Blockchain una temática atractiva para el estudio.

> *"Anything that you can conceive of as supply chain, blockchain can vastly improve its efficiency – it doesn't matter if it's people, numbers, data, money"* - Ginny Rometty, CEO de IBM
>
> *"Blockchain is a technological tour de force"* – Bill Gates, fundador de Microsoft
>
> *"Blockchain is thus also turning out to be more complicated than most of us thought"* – Centre of the Digital Economy, University of Surrey (Reino Unido)

Las citas anteriores son una clara muestra de la diversidad de opiniones que hoy genera el término "Blockchain". Esta volatilidad es uno de los principales factores por los que se plantea la realización del presente Trabajo de Fin de Máster. Tras investigar esta tecnología y alguna de sus principales aplicaciones, se pretende dar respuesta a la incertidumbre general existente en torno a esta tendencia y aportar un análisis claro y completo sobre esta nueva realidad tecnológica.

Por otro lado, el carácter emergente de la tecnología Blockchain permite el estudio de la viabilidad de su uso en multitud de campos. Actualmente se habla fundamentalmente de su empleo en el sector de las criptomonedas, en logística, cadenas de producción…no obstante, los principios y el funcionamiento de Blockchain son probablemente compatibles con otros muchos propósitos. Por tanto, la posibilidad de indagar en estas nuevas oportunidades supone el segundo fuerte incentivo para la realización de esta tesis.

1.2 INTRODUCCIÓN A BLOCKCHAIN: PRINCIPIOS Y BASE TÉCNICA

La tecnología Blockchain es, fundamentalmente, una base de datos distribuida constituida por una serie de bloques que emplea un sellado o *hash* de tiempo para evitar modificaciones indeseadas tras crear un dato. Esta base de datos descentralizada está conectada a multitud de nodos, donde cada dato (transacción) se registra y encripta, asegurándose la no alteración del mismo.

Cualquier aplicación de esta tecnología debe contar con cuatro elementos:

- Un registro compartido

Contabiliza toda la información que atraviesa la Blockchain y es accesible por todos los equipos que constituyen la red de nodos en función de una serie de permisos predefinidos.

- Un Contrato Inteligente (o *Smart Contract*)
  Básicamente, consiste en un elemento *software* (un programa) que define las reglas de negocio en las que se deben basar las transacciones y va embebido en la propia Blockchain.
- Privacidad
  Este elemento está garantizado en cualquier Blockchain debido a factores como el anonimato de las transacciones y la encriptación de la información.
- Confianza
  Cada vez que se produce una transacción dentro de la Blockchain, está se incluye en el registro compartido, facilitando la transparencia de la información y permitiendo futuras auditorias. Este concepto se conoce comúnmente como "consenso distribuido".

Estos elementos, junto con otras propiedades que se describirán más adelante, hacen de Blockchain una tecnología prometedora con una amplia variedad de campos de aplicación. Entre sus principales beneficios están el ahorro de tiempo y dinero (ausencia de intermediarios para las transacciones entre dos nodos), una reducción potencial del volumen de fraudes y un aumento de la transparencia y la confianza.

Por otro lado, detrás del concepto "Blockchain" existe una compleja realidad técnica. Como su nombre indica, la base de este paradigma es una "cadena de bloques". Cada uno de estos bloques hace referencia a un trozo de información que almacena un listado de transacciones entre distintos puntos de la red distribuida. Estas piezas de información no son locales, sino que pueden ser editadas en cualquier momento y de un modo anónimo por cualquier equipo de la red. Dicho esto, el término "cadena" toma importancia a la hora de asegurar que no haya modificaciones ilícitas sobre todas estas transacciones.

Cada bloque se firma digitalmente de manera única, por lo que si alguien intenta modificarlo la firma dejará de ser válida y se alertará de este intento de modificación. Este firmado se ve reforzado en términos de seguridad por la cadena: cada bloque almacena su firma y la de su

antecesor, por lo que un cambio en un bloque afecta la firma de todos los que le sucedan. De este hecho se puede inferir que la seguridad de un bloque es proporcional a su antigüedad. Esta firma es generada en un proceso denominado minado, consistente en el empleo de un algoritmo de gran complejidad para la creación de una clave. Este proceso es llevado a cabo de manera colaborativa por miles de ordenadores de todo el mundo [1]. La siguiente ilustración sintetiza gráficamente los conceptos recién mencionados:

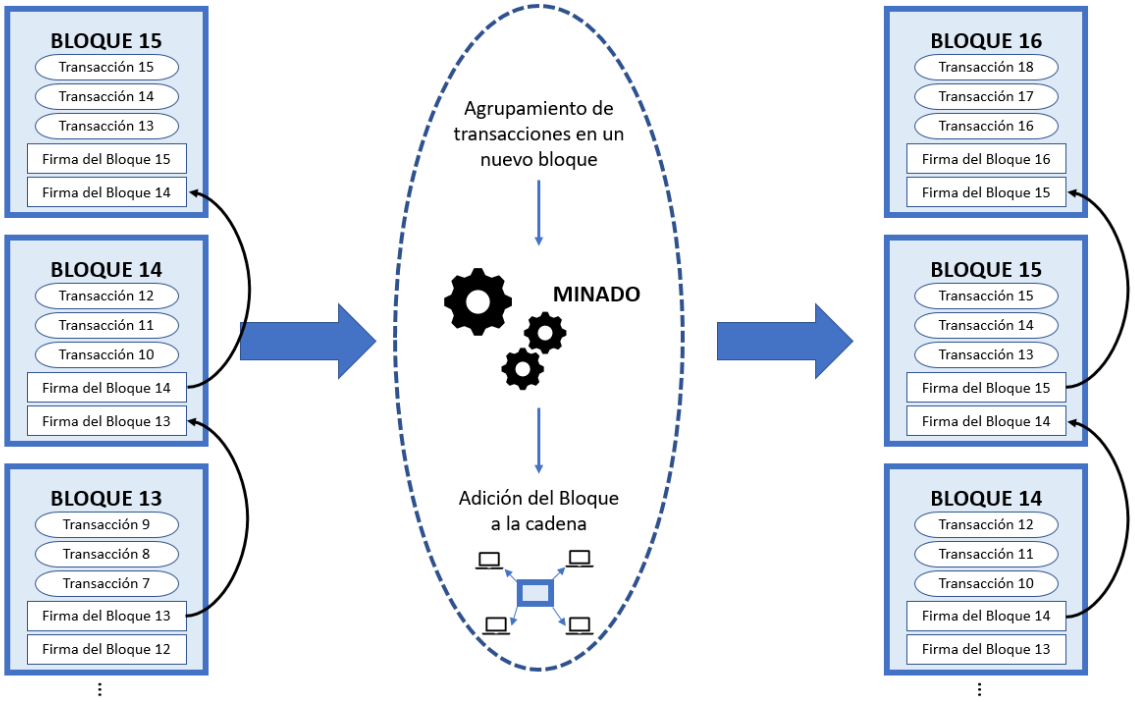


Figura 1. Blockchain: Base Técnica

Los elementos y procesos descritos en la anterior ilustración serán estudiados, analizados e implementados en mayor profundidad en futuras secciones del proyecto.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

A pesar de que la mayor parte de este Trabajo de Fin de Máster ha consistido en la realización de una labor de investigación, ciertas labores de análisis han ido acompañadas del empleo de la tecnología con el fin de alcanzar un mayor grado de comprensión del paradigma.

Aunque a lo largo del proyecto se mencionarán numerosas tecnologías (protocolos, *software*, *hardware*, etc.), tales como Node.js, Web3, Hyperledger…los procesos en los que se ha implementado a algún tipo de solución han recurrido fundamentalmente a cuatro tecnologías, descritas en los siguientes apartados del capítulo.

2.1 REMIX – SOLIDITY IDE

remix

Remix es una herramienta *open – source* que permite al usuario el desarrollo de *Smart Contracts* (piezas de software que aportan cierta "inteligencia" a las transacciones de la cadena) en Solidity **directamente desde el navegador**. Está escrito en JavaScript, y soporta labores de codificación, *testing*, *debugging* e implantación de contratos en la red Ethereum.

La gran ventaja de esta herramienta es el hecho de que el usuario no necesita instalar ningún componente localmente, ahorrándose una gran cantidad de recursos de memoria.

Fundamentalmente, esta herramienta *online* cuenta con las siguientes pestañas:

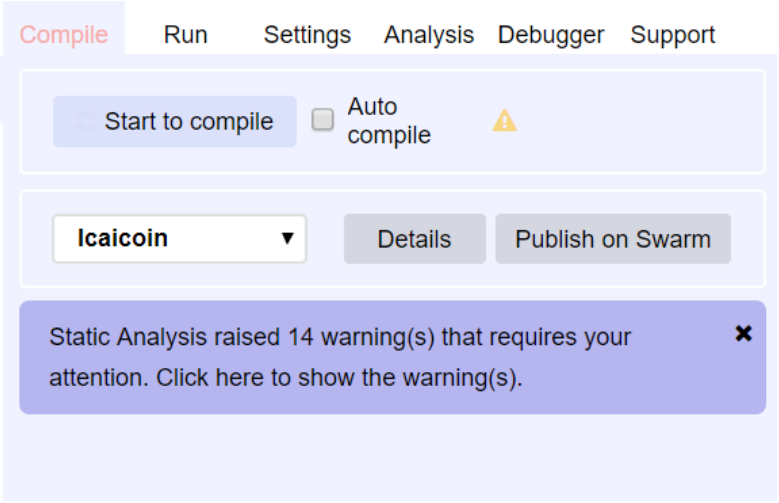


Figura 2. Remix IDE. Funcionalidad

La **pestaña *compile*** incluye toda la funcionalidad relativa a la compilación del código en cuestión. Permite señalar el archivo a compilar y visualizar el resultado de este proceso (*warnings* y errores, o ausencia de estos).

La **pestaña *run*** permite introducir todos los datos de la cuenta Ethereum sobre la que van a recaer los gastos de implantación del contrato. El usuario tiene la opción de introducir el ecosistema de ejecución (red real, red de pruebas, Web3, etc.), el ID de su cuenta y el número de Gas a invertir para "motivar" la ejecución del programa.

La **pestaña *Settings*** permite acceder a la documentación, escoger la versión del compilador o establecer una serie de ajustes personalizados, entre otras opciones.

La **pestaña *Analysis*** permite configurar la comprobación de diversos aspectos del código en términos de seguridad, tarificación, y otros.

La **pestaña *Debugger*** permite personalizar el proceso de depuración del código, muy útil a la hora de identificar y resolver errores.

La **pestaña *Support*** aporta información de contacto para reporte de incidencias o mejoras sobre la herramienta, además de un *chat* para comunicación y resolución de dudas con la comunidad Ethereum.

2.2 METAMASK

Metamask es una de las muchas herramientas que están surgiendo con el principal objetivo de hacer de Ethereum una Blockchain de uso sencillo y acceso disponible a todo el mundo.

Fundamentalmente, esta solución constituye un puente entre un navegador (Chrome, Firefox, Opera o Brave) y una Blockchain, permitiendo la ejecución de aplicaciones descentralizadas sin necesidad de descargar e instalar un nodo Ethereum, acción que supondría un consumo de muchos Gb de memoria.

Con esta herramienta, cualquier usuario puede obtener datos de la Blockchain y gestionar de un modo seguro su identidad y sus transacciones desde su navegador. Además, este producto ofrece otras funcionalidades, tales como:

- Creación o agregado de *tokens* distintas de la moneda nativa de Ethereum (Ether).
- Conexión de Dapps con la Blockchain mediante el uso de la API Web 3.
- Envió y recepción de transacciones.

La siguiente captura presenta el aspecto de la página de inicio de una cuenta Metamask:

Figura 3. Metamask. Página de Inicio

2.3 *ETHERSCAN*

Etherscan es un explorador a nivel de bloque y una plataforma de análisis para la red Ethereum, así como una plataforma descentralizada para la implantación de *Smart Contracts*.

Permite una conexión directa con la herramienta Metamask, por lo que resulta un producto muy interesante tanto en términos de aprendizaje como de análisis del comportamiento de transacciones, contratos o *tokens*. Ofrece funcionalidades como:

- Visualización de la inserción de bloques en la red Ethereum en tiempo real:

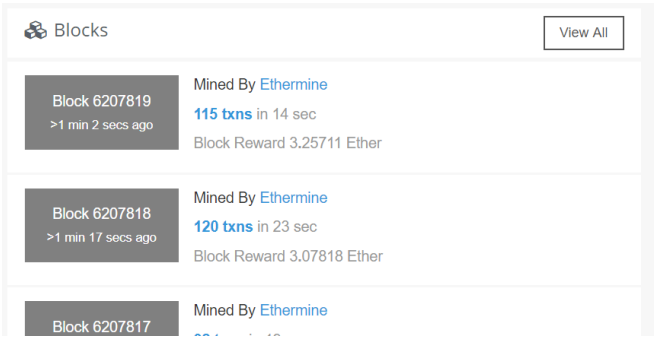


Figura 4. Red Ethereum. Actualización en tiempo real

- Visualización de estado y detalle de transacciones (ver ANEXO C).
- Visualización de diversos detalles relativos a *tokens* y *Smart Contracts* (código, transacciones, participantes, etc.) (ver ANEXO C).
- Realización labores de análisis y exploración tanto en la red real de Ethereum como en sus redes de prueba (Ropsten).
- Etc.

2.4 SOLIDITY. DESCRIPCIÓN DE SMART CONTRACTS

Solidity es el lenguaje de programación de *Smart Contracts* de Ethereum por excelencia, aunque actualmente es soportado por muchas otras Blockchain. Este lenguaje comparte muchas similitudes con otros como Java, JavaScript o C.

Como si de una clase en Programación Orientada a Objetos se tratará, cada contrato en un código Solidity contiene componentes como variables de estado, funciones y tipos de datos. Además, existen propiedades más especificas de los *Contracts*, como las cláusulas, los mapeos o los eventos.

Una referencia más detallada sobre la programación en Solidity será proporcionada en los apartados que contienen las distintas implementaciones realizadas en el proyecto, todas ellas descritas empleando este lenguaje. No obstante, cabe concluir señalando la **importancia del proceso de *testing*** en la programación en Solidity debido al alto riesgo de error**,** y al elevadísimo coste que puede suponer éste.

Capítulo 3. ESTADO DE LA CUESTIÓN

Blockchain es considerada por multitud de profesionales y compañías como la tendencia tecnológica que revolucionará los Sistemas de Información en los próximos años. No obstante, también existe un fuerte escepticismo por parte de expertos de esta misma área. Esta sección se centra en la descripción de la realidad tecnológica actual en torno al ámbito de la Blockchain, sus potenciales aplicaciones y los desafíos a los que se enfrenta.

3.1 BITCOIN Y CRIPTOMONEDAS

Actualmente, no cabe duda de que la aplicación más extendida (aunque no la única) del concepto de Blockchain son las criptomonedas. Como su propio nombre indica, la implementación de Bitcoin y demás monedas (Litecoin, Ethereum, etc.) incluye un elevado y complejo uso de criptografía. La característica fundamental de estos sistemas es la descentralización: la moneda funciona sin necesidad de un administrador central. Existen opiniones de todo tipo sobre la cuestión de si las criptomonedas deberían ser consideradas como una divisa desde el punto de vista económico.

Según El Economista: "llamamos dinero a todo activo aceptado como medio de pago o medición del valor por los agentes económicos para sus intercambios y además cumple con la función de ser unidad de cuenta y depósito de valor" [2]. Si se trata de relacionar las criptomonedas con esta definición, se pueden hacer las siguientes observaciones:

- Numerosas compañías de renombre aceptan Bitcoin como medio de pago
- El valor actual del Bitcoin es de unos 10400 €

A simple vista, las observaciones anteriores permiten inferir que, efectivamente, cualquier criptomoneda puede ser considerada como una forma de dinero o medio de pago. A pesar de esto, en la actualidad existen posturas reacias a esta tendencia.

Firmas como JPMorgan Chase & Co. o Credit Suisse Group AG han calificado la criptomoneda de "burbuja al borde de la explosión" o "opción de inversión absurda", entre otros. El hecho de que compañías de esta envergadura no apoyen el desarrollo de Bitcoin y demás *cryptocurrencies* puede verse justificado por la existencia de diversos inconvenientes o desafíos en estas "tecnologías".

En primer lugar, tanto Bitcoin como la tecnología Blockchain en general presentan desafíos significativos en términos de escalabilidad y efectividad que deben ser resueltos antes de que las criptomonedas tengan la mínima oportunidad de sustituir al "dinero convencional". Actualmente, la red mundial de Bitcoin puede procesar un máximo de siete transacciones por segundo, y es recomendable una espera de unos 50 minutos entre cada transacción, factores que impiden el uso de estas tecnologías para compras habituales del día a día [3].

Otro desafío/inconveniente de esta criptomoneda viene de la mano de los "monederos Bitcoin" (o *Bitcoin Wallet*). Esencialmente, un monedero es una aplicación *software* donde un usuario puede guardar sus bitcoins o, más bien, una clave privada que le da acceso a sus criptomonedas, facilitando las transacciones de esta divisa. Por tanto, un robo de credenciales, un ataque informático o la destrucción del dispositivo (en monederos basados en *hardware*) pondría en un grave peligro todo el capital del usuario [4].

Además, algunas de las "ventajas" que ofrecen las criptomonedas pueden, en realidad, suponer inconvenientes en algunos aspectos o escenarios. Por ejemplo, la descentralización y ausencia de gobierno incluye el riesgo de que una agrupación poderosa de usuarios se deshaga de sus monedas y abandone el sistema, devaluando gravemente la red Bitcoin y dañando a gran cantidad de inversores. Por otro lado, la total transparencia de la Blockchain puede permitir la fácil visualización de todo el historial financiero de un usuario por parte del receptor de una transacción.

En definitiva, Bitcoin supone una de las principales novedades financieras y tecnológicas de los últimos tiempos, por lo que el análisis de las ventajas e inconvenientes de su implementación resulta de gran interés.

3.2 LA REVOLUCIÓN BLOCKCHAIN

En la actualidad, gran cantidad de individuales y organizaciones están trabajando en la investigación y la explotación de las facilidades que ofrece la tecnología Blockchain para innovar en una gran variedad de campos. Un claro ejemplo de esto es R3, una firma tecnológica resultado de la unión de más de doscientas instituciones financieras de todo el mundo. Esta organización centra sus esfuerzos en el desarrollo de una plataforma basada en Blockchain, Corda, orientada tanto al sector financiero como a cualquier otro tipo de industria.

La constante publicación de noticias sobre Bitcoin y Ethereum parece indicar que la implementación de Blockchain para multitud de propósitos es ya una realidad. Sin embargo, la mayor parte de iniciativas se encuentran aún en proceso de implantación o investigación. El último Ciclo de Gartner sobre Tecnologías Emergentes [5] posiciona a Blockchain dentro del grupo de "expectativas sobredimensionadas":

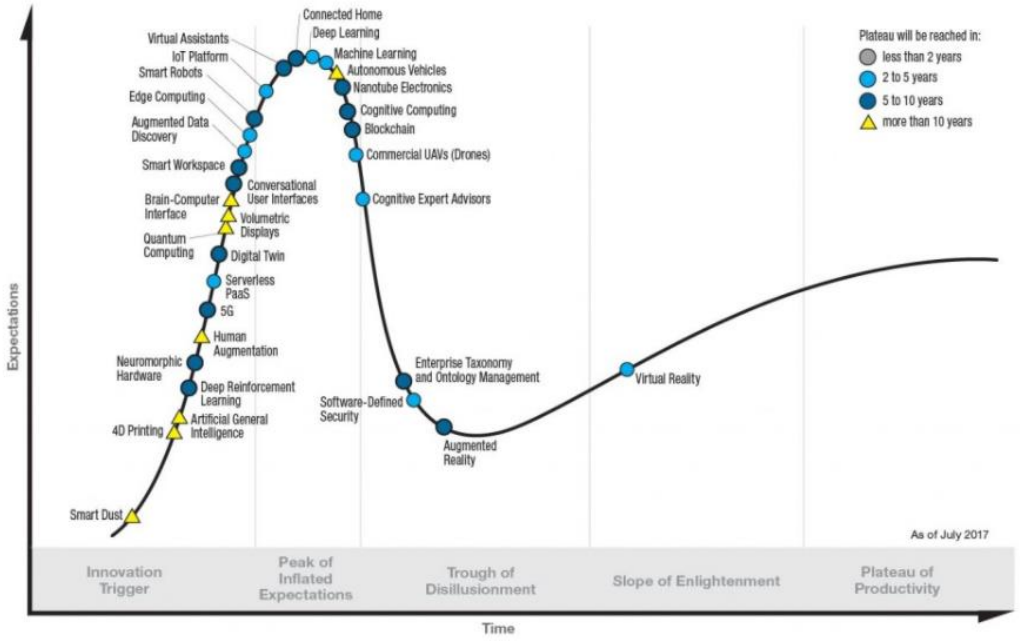


Figura 5. Ciclo de Gartner sobre Tecnologías Emergentes. 2017 [5]

Como se puede observar en la ilustración anterior, la tecnología Blockchain se encuentra prácticamente en la frontera con la categoría "abismo de desilusión", debido posiblemente a la presencia de opiniones adversas a esta tendencia o la publicación de experimentos fallidos sobre la implementación de Blockchain. A pesar de esto, los esfuerzos e interés de prestigiosas firmas, profesionales y estudiantes en esta tendencia parecen indicar que Blockchain no se estancará en esta última etapa, alcanzando la "meseta de productividad" en un margen prudente de tiempo.

Analistas de la misma consultora tecnológica señalan que una gran mayoría de empresas líderes no han llevado a cabo implementaciones significativas basadas en tecnología Blockchain. Estos, además, añaden que uno de los principales motivos de este hecho es la ausencia de algunos de los recursos con los que cuentan las tecnologías actuales, tales como servicios de soporte y desarrollo, APIs, estándares, etc. No obstante, esto no significa que la firma no confíe en Blockchain como tecnología revolucionaria. Una de las áreas que identifica como idóneas para la implantación de este nuevo paradigma es la Gestión de Cadenas de Producción (o *Supply Chain Management*).

La afirmación anterior cobra sentido al comparar las propiedades/requisitos de SCM y Blockchain. La Gestión de la Cadena de Producción constituye una estrategia de operaciones basada en la integración efectiva de todos los agentes que intervienen desde la obtención de la materia prima hasta la entrega del producto al usuario final [6]. Blockchain, con su registro compartido, podría permitir la unificación y coordinación de todos estos flujos de dinero, bienes e información. Este hecho podría suponer importantes innovaciones en industrias tales como el *retail* o la logística.

Aunque SCM sea posiblemente la aplicación potencial de Blockchain más mencionada (aparte de las criptomonedas), investigaciones en multitud de campos se están llevando a cabo en la actualidad. Algunas áreas que destacar en este aspecto son [7]:

- Servicios Sanitarios
  El almacenamiento de los expedientes de pacientes en una red distribuida y descentralizada basada en Blockchain puede traer beneficios en términos de agilidad

y acceso, entre otros. Mecanismos propios de este paradigma podrían ayudar también a mantener la seguridad y confidencialidad de datos médicos.

- Gobierno y Administración Pública
  Estados como EAU, Estonia o Corea del Sur están actualmente trabajando con líderes tecnológicos en el traslado de ciertas competencias a una Blockchain. Esta tecnología también podría emplearse en ámbitos como el sistema electoral, la declaración a Hacienda, el pago de impuestos, etc.
- Identificación y Autenticación digital
  Un claro ejemplo de esto es el sistema *REMME*, basado en la sustitución del sistema de autenticación tradicional "usuario-contraseña" por el almacenamiento de certificados SSL en una Blockchain
- Multimedia
  Actualmente, existen desarrollos de diversos sistemas basados en Blockchain que pretenden administrar la propiedad intelectual y el pago de derechos de autor a artistas de todo tipo, sin necesidad de intermediarios.

Multitud de propuestas y estudios sobre nuevas áreas de aplicabilidad de la Blockchain son publicadas a diario. A pesar de ser sin duda una tendencia prometedora, grandes esfuerzos en término de educación y resolución de desafíos son aún necesarios para hacer de Blockchain una tecnología consolidad e idónea para una gran cantidad de aplicaciones.

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 JUSTIFICACIÓN

A pesar de que la principal motivación/justificación del presente Trabajo de Fin de Máster es meramente divulgativa y académica, su carácter parcialmente práctico podría levantar un interés económico por parte de organizaciones de diversos sectores industriales.

Es cierto que el paradigma tecnológico de la Blockchain está en auge en la actualidad; sin embargo, la cantidad de casos de éxito de esta tendencia en organizaciones concretas es bastante limitada. El hecho de que aún no exista un gran número de empresas que se hayan sumergido de lleno en implementaciones basadas en Blockchain podría potenciar la utilidad de trabajos de análisis y estudio como el aquí reflejado.

En primer lugar, la investigación en profundidad sobre diferentes aspectos de la tecnología podría resultar de interés tanto para profesionales del sector industrial como para docentes. El principal motivo de esto es que el objetivo fundamental de esta primera dimensión del proyecto es describir y aportar cierta perspectiva sobre la Blockchain, proporcionar una especie de "guía" que explore todo aspecto que sea necesario para llevar a cabo cualquier tipo de implementación, añadiendo además ciertas observaciones y omitiendo la porción de información confusa y/o equívoca que abunda en la red. Por tanto, ésta podría ser una fuente de gran utilidad y fiabilidad para cualquier organización que esté tratando de revolucionar su modelo de negocio mediante la incorporación de Blockchain.

Por otro lado, la segunda gran sección del proyecto, con una orientación mayoritariamente práctica, presenta un alto interés divulgativo para profesionales y/o investigadores que estén tratando de integrar Blockchain con otros paradigmas en la actualidad. Dado que gran parte de esta sección se basa en el análisis de la integración de carácter simbiótico entre Blockchain y *Semantic Web*, especialistas de ambos campos son claros destinatarios de la presente tesis.

4.2 OBJETIVOS

El presente Trabajo de Fin de Máster cuenta con tres objetivos fundamentales, evolucionando desde una mera labor de investigación hasta secciones más orientadas a la aplicación industrial de los resultados obtenidos.

En primer lugar, se llevará a cabo una investigación y análisis exhaustivo de Blockchain. Se partirá del estudio de los principios, el funcionamiento básico y las bases de la implementación de esta tecnología, y se analizará su aplicación más extendida hasta la fecha: las criptomonedas. Respecto a este último concepto, se tratará de describir el tipo de economías que genera este nuevo sistema, la aplicación actual de Bitcoin y demás *cryptocurrencies*, las propiedades y el funcionamiento de las distintas transacciones dentro de esta nueva red distribuida, etc.

El segundo objetivo principal de la tesis consistirá en el estudio de la aplicabilidad de todos los conceptos extraídos de la labor de investigación. Para ello, se trabajará con implementaciones de criptomonedas, se hará uso de herramientas existentes para el despliegue de Blockchains y se tratará de analizar la posible aplicación de Blockchain en tándem con otros paradigmas, tales como IoT (*Internet of Things*).

Por último, se analizará la viabilidad de un nuevo caso de uso de la tecnología Blockchain. En concreto, se tratará de analizar la aplicabilidad y viabilidad de la integración simbiótica de esta tecnología con la Web Semántica. Esta última tendencia, presente en el panorama tecnológico desde hace varios años, pretende llevar a cabo una revolución en la WWW, incrementando significativamente la interconexión entre la información presente en ésta y transformando dicha información de modo que pueda ser íntegramente interpretada y analizada mediante máquinas (Sistemas de Información). Por tanto, este último objetivo se basa en el empleo de todos los resultados obtenidos tras las tareas de análisis e investigación iniciales para un propósito industrial.

4.3 METODOLOGÍA

En términos metodológicos, se pretende desarrollar un proyecto de "investigación-acción", pues, en primer lugar, se procederá a conocer el contexto de la tecnología (Blockchain) para, posteriormente, evaluar su implantación y aplicabilidad para satisfacer un caso de uso concreto.

Además, se hará uso de métodos diversos en función de la fase del proyecto a ejecutar en cada momento.

Por un lado, la parte de investigación del Trabajo, centrada en la recopilación y el análisis de información sobre Blockchain, hará uso principalmente de:

- Métodos empíricos para la obtención y elaboración de la información.
- Métodos teóricos para la interpretación conceptual de los datos empíricos obtenidos, tales como el análisis o la síntesis.

Tras la investigación, se aplicará el método "lógico-deductivo", modelando la posible aplicabilidad de la tecnología en función de los conocimientos obtenidos hasta ese momento.

Por otro lado, en cuanto a la redacción de la memoria del proyecto, se empleará una metodología de trabajo "en cascada", pues cada apartado de la misma se completará tras finalizar su parte análoga de la planificación. La memoria será elaborada siguiendo las pautas de formato y estructura establecidas por la universidad.

Con el fin de distribuir las distintas labores a realizar de un modo efectivo, y garantizar que la metodología descrita se ejecuta según lo previsto, se ha elaborado una planificación del TFM en el tiempo. La próxima sección presenta el diagrama de Gantt del proyecto, donde se refleja el tiempo estimado de dedicación a cada tarea de éste.

4.4 PLANIFICACIÓN

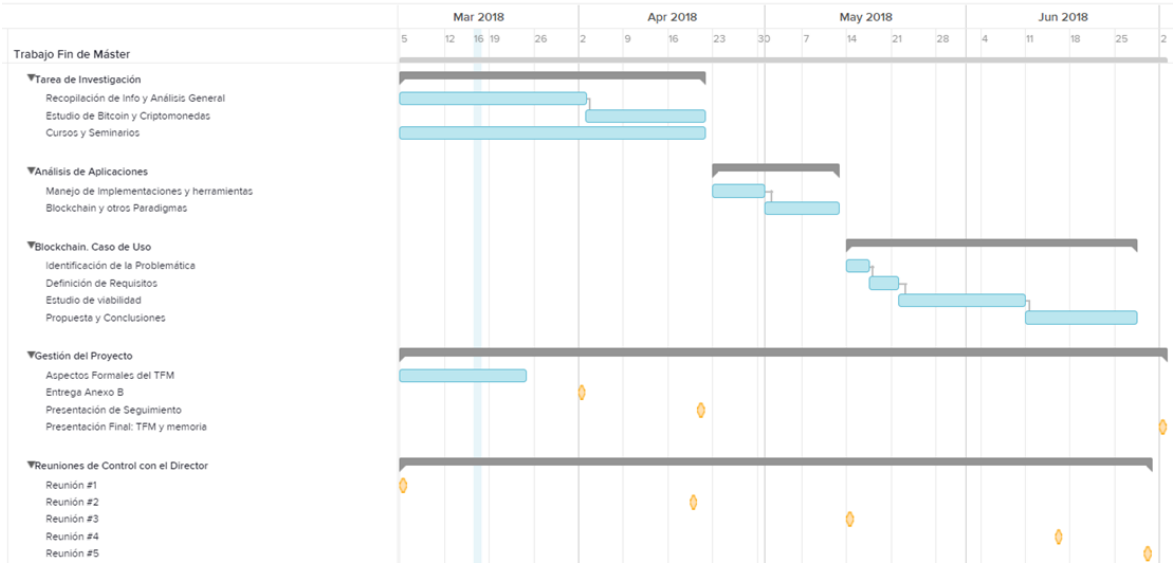


Figura 6. Planificación del TFM. Diagrama de Gantt

Como se puede observar en la ilustración anterior, para la planificación del proyecto se han dividido de manera secuencial las acciones a realizar en tres grandes partes, coincidiendo éstas con los objetivos previamente definidos.

Además, se ha definido un cuarto grupo de tareas destinado a la gestión del TFM. Este período incluye la definición de diversos aspectos formales de la tesis (alcance, objetivos, metodología a emplear, etc.), los principales hitos/entregas a llevar a cabo, y las distintas reuniones de seguimiento con el director del proyecto. Al estar realizando parte del Trabajo de Fin de Máster desde Estados Unidos, algunas reuniones deberán llevarse a cabo de manera telemática.

Por último, cabe señalar que se ha planificado la entrega de la memoria del proyecto para el **2 de julio como parte del Plan de Contingencia de este**. De este modo, se deja un margen de tiempo para entregar el proyecto en caso de que se diera cualquier contratiempo. De hecho, finalmente se ha aprovechado este recurso de contingencia debido a una ligera subestimación inicial de la dimensión del alcance del proyecto, extendiéndose la realización del mismo hasta finales de agosto del 2018.

4.5 ESTIMACIÓN ECONÓMICA

Como se ha descrito hasta el momento, la mayor parte del presente Trabajo de Fin de Máster se basa en una labor de investigación. De hecho, la sección más práctica de este no incluye un gran proceso de implementación ni tiene ninguna dimensión material. Por tanto, se puede concluir que la mayor proporción de costes de la tesis está destinada a cubrir los Recursos Humanos requeridos.

En términos de personal, cabe señalar que el presente proyecto ha contado con la colaboración de un **Consultor Junior** y un **Jefe de Proyecto**. A continuación, se describe un desglose que justifica las horas invertidas por cada uno de los miembros del equipo:

- En primer lugar, se ha estimado que el Consultor Junior (estudiante de ICAI), invertirá un total de 475 horas, distribuidas de la siguiente manera
    - 200 horas invertidas en la familiarización con la tecnología Blockchain y la ejecución y descripción de la labora de investigación llevada a cabo. Esto incluirá tanto trabajo independiente como cierta colaboración con *Drexel College of Computing and Informatics*.
    - 125 horas invertidas en el análisis y empleo de herramientas y plataformas para la implementación de soluciones basadas en Blockchain.
    - 150 horas orientadas al análisis de viabilidad y planteamiento de un posible caso de uso de la tecnología investigada y analizada previamente.
- Por su parte, el Jefe de Proyecto invertirá un total aproximado de 50 horas, destinadas a la realización de diversas tareas de supervisión del proyecto, tales como las reuniones de seguimiento, la revisión del trabajo realizado y la resolución de conflictos.

La información recién descrita, junto con el coste estándar que supone la contratación de las dos categorías mencionadas, permiten estimar el coste del presente proyecto en términos de personal:

Profesional	Tiempo invertido [h]	Precio [€/h]	Coste [€]
Consultor Junior	475	50.0	23,750.00
Jefe de Proyecto	50	65.0	3,250.00
Total RRHH			**27,000.00**

Tabla 1 Coste del Proyecto. RRHH

Por otro lado, la realización del presente Trabajo de Fin de Máster ha supuesto también una ligera inversión en términos de "material de oficina", descrita en la siguiente tabla:

Material	Coste [€]
Licencias *Software*	200.00
Ordenador Portátil	1,000.00
Otros (reprografía, papel, etc.)	100.00
Total	**1,300.00**

Tabla 2 Coste del Proyecto. Material de Oficina

Por lo tanto, se concluye que el coste total que supone la realización de este Trabajo de Fin de Máster es de 28,300.00 €.

Capítulo 5. ESTUDIO DE LA TECNOLOGÍA BLOCKCHAIN

El presente capítulo se centra en la consecución del primer gran objetivo del Trabajo de Fin de Máster. Para ello, se describirá la investigación realizada sobre la tecnología Blockchain, sus principios y elementos fundamentales. Además de una descripción objetiva de los aspectos anteriores, se tratará de aportar un análisis crítico sobre cada uno de ellos.

5.1 INTRODUCCIÓN. FUNCIONALIDAD Y CONCEPTOS BÁSICOS

En esencia, la Blockchain está constituida por un registro común y distribuido que facilita los procesos de grabado de transacciones y seguimiento de activos dentro de una red empresarial [8]. Lejos de ofrecer una definición sobre la tecnología, se tratará de describir la funcionalidad y los aspectos técnicos/funcionales que hay detrás de ésta, dado que se ha encontrado una inmensa variedad de definiciones/descripciones de la Blockchain en la bibliografía empleada.

Esta tecnología ofrece una gran flexibilidad en términos de aplicabilidad, de modo que los activos antes mencionados pueden hacer referencia a un bien tangible (dinero, un coche, una persona) o intangible (información, propiedad intelectual, patentes). En definitiva, cualquier "cosa" de valor puede ser registrada e "intercambiada" en una red Blockchain, reduciendo significativamente el riesgo y el coste de los procesos necesarios para ello.

A lo largo de la historia, se han empleado diversos mecanismos para la realización de transacciones e intercambios de bienes y capitales, tales como el dinero en efectivo, las cartas de crédito o los sistemas bancarios. Todos estos procedimientos contienen, en mayor o menor medida, prestaciones que aportan cierto nivel de protección y confianza a los partidos involucrados.

No obstante, estos medios presentan inconvenientes en términos de eficiencia, seguridad y costes. A esto hay que sumarle el crecimiento exponencial que están experimentando los volúmenes transacciones de datos/bienes en la actualidad. Además, este aumento llegara al clímax con la expansión generalizada del Internet de las Cosas (IoT). Para afrontar con eficacia estos y otros retos, las redes empresariales requieren de una revolución que simplifique la gestión de activos de un modo transparente y fidedigno.

Dentro del sector financiero, esta situación provocó el surgimiento del **Bitcoin**, una "divisa digital", construida sobre los pilares de la Blockchain y lanzada en el 2009 por **Satoshi Nakamoto** (pseudónimo). Un mayor estudio sobre la cripto-moneda se llevará a cabo en futuras secciones.

En definitiva, la idea fundamental de la transformación hacia la Blockchain aparece simplificada en el siguiente diagrama:

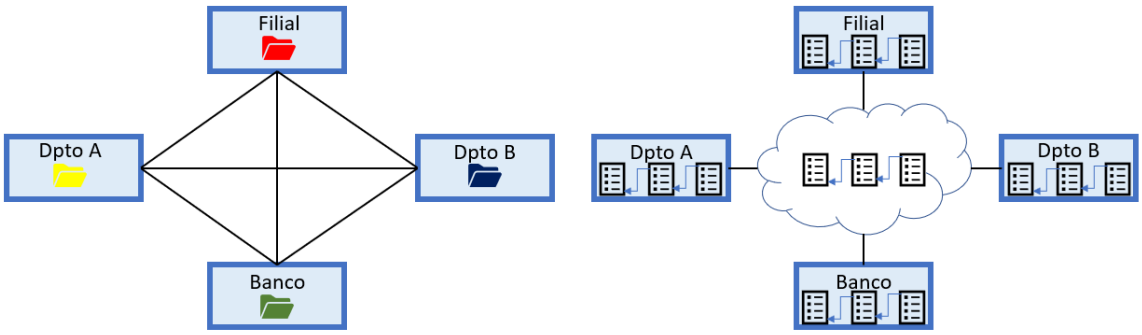


Figura 7. Red Empresarial Tradicional vs Basada en Blockchain

Por un lado, en una red tradicional, el registro y seguimiento de transacciones y/o activos son llevados a cabo de manera independiente por cada uno de los agentes involucrados (departamentos, filiales, entidades bancarias, auditores). Este replicado de registros provoca problemas en términos de eficiencia, costes y complejidad. Además, todas estas transacciones suelen ser gestionadas/aprobadas por una entidad reguladora (bancaria, legal, gubernamental), lo que general una fuerte dependencia de agentes externos a la organización.

Con la evolución hacia una red empresarial basada en la tecnología Blockchain, el seguimiento de activos y el acceso a la información mejora significativamente gracias a la introducción de un registro común constantemente actualizado disponible para todos los agentes involucrados. De este modo, cada participante/nodo puede realizar/recibir una transacción, cuya información se transmitirá al resto de agentes de manera sincronizada y casi instantánea. Además, como se describirá más adelante, la red cuenta con una seria de modelos y mecanismos que garantizan la seguridad, autenticidad y auditoría de las transacciones.

5.1.1 PRINCIPIOS DE BLOCKCHAIN

Tras analizar el trasfondo funcional y tecnológico de la Blockchain, se ha tratado de extraer una serie de principios o fundamentos que caracterizan cualquier implementación de la misma. Esta recopilación pretende describir una serie de conceptos básicos previo a profundizar en los aspectos más técnicos y los elementos comunes en una red Blockchain:

A) ARQUITECTURA DISTRIBUIDA

La transmisión de información/transacciones dentro de la Blockchain se lleva a cabo a través del empleo de un mecanismo *peer-to-peer*, no existe un nodo central. De este modo, se elimina el riesgo provocado por un "*single-point-of-failure"* (punto crítico de fallo) propio de redes centralizadas.

Además, está distribución del poder entre los distintos nodos previene de ataques contra la disponibilidad de la red por parte de alguno de estos. En el caso en que se diera un intento colectivo de actividad "no-lícita" sobre la red, la estructura de la Blockchain garantiza una detección instantánea de esta actividad en toda la red.

Esta descentralización implica la ausencia de intermediarios o el almacenamiento en servidores centrales. El funcionamiento de la Blockchain es el resultado de una colaboración equitativa entre todos los participantes, sin necesidad de un órgano regulador o un agente externo que valide las transacciones. Este factor presenta grandes ventajas en términos de reducción de costes y tiempo; sin embargo, la ausencia total de gobernabilidad puede

suponer un riesgo significativo en función de la situación y el caso de uso concreto de la tecnología Blockchain.

B) TRANSPARENCIA

Como se mencionará posteriormente, la tecnología Blockchain cuenta con propiedades que garantizan un alto grado de privacidad, al permitir que los detalles de cada transacción sean accesibles exclusivamente por los agentes involucrados en la misma.

No obstante, el hecho de que esta red distribuida cuente además con un registro 100% auditable hace de la transparencia un elemento garantizado en toda Blockchain. Toda entrada a dicho registro debe ser previamente validada por el total del sistema. Además, el cambio de un bloque/transacción implica la realización de modificaciones sobre el resto de bloques. Por lo tanto, aparentemente resulta imposible el ocultamiento o manipulación fraudulenta de cualquier transacción dentro de la red [9].

Esta transparencia permite también un ahorro de recursos (ausencia de intermediarios y sistemas antifraude) y una simplificación de la presentación de los datos, factores que facilitan la realización de labores de auditoría tanto propias como a proveedores, así como la evaluación del cumplimiento de estándares y regulaciones.

La transparencia propia de la Blockchain toma un rol fundamental dentro del campo de la Gestión de la Cadena de Producción (o SCM). La alta presencia de la globalización y las estrategias de *outsourcing* en esta área da lugar no solo a un ahorro en costes sino también a un aumento de la complejidad de la *Supply Chain*. Aunque parezca una justificación algo extremista, se han encontrado numerosos casos de cadenas de producción de grandes multinacionales en los que se detectaron agentes/bloques (materia prima, producción, distribución, etc.) de la cadena con práctica sospechosas que podrían incluso financiar guerras/terrorismos en determinadas naciones. Por tanto, la transparencia de la Blockchain puede ayudar a detectar de manera inmediata estos casos, favoreciendo la reputación de la entidad y el cumplimiento de los principios adecuados tanto en términos legales como éticos.

C) DURABILIDAD Y PERSISTENCIA

Decir que la Blockchain es inmutable no es lo suficientemente preciso debido a que, de hecho, esta red distribuida está en constante cambio. Aunque la adición de nuevos bloques a la cadena es continua, sí es cierto que los bloques ya incorporados en la Blockchain no pueden ser modificados *a posteriori*. Como ya se ha mencionado con anterioridad, esta modificación sería posible en teoría, pero los recursos computacionales necesarios para llevar a cabo este tipo de "ataque" sobre la red serían inmensos.

Como ocurre en la WWW (*World Wide Web*), la distribución de bloques idénticos en toda la red proporciona a la tecnología Blockchain una **robustez integrada** [10], al eliminarse la existencia de un único punto de control o de fallo.

Sin embargo, la durabilidad y persistencia de todos estos bloques puede traer ciertos inconvenientes en términos de, por ejemplo, coste de mantenimiento. Se necesita una gran cantidad de recursos (computacionales, de almacenamiento, etc.) para que cada nodo de la red distribuida tenga acceso a su copia del registro compartido. Un ejemplo de respuesta a este aspecto es el operador OP_RETURN introducido por Bitcoin [11]: cualquier transacción que contenga en su *script* este código puede ser "extraída" de la Blockchain, sustituyéndose el contenido total del bloque/transacción por un breve resumen del estado del mismo.

D) PRIVACIDAD

En la actualidad, los datos y/o la información son considerados como activos cruciales en toda organización de cualquier sector. Con el fin de garantizar un uso efectivo y apropiado de dichos *assets*, en los últimos años han surgido diversos organismos y regulaciones dedicados, fundamentalmente, a prevenir violaciones en términos de acceso y privacidad.

La Unión Europea constituye uno de los pioneros en este aspecto con el conjunto de normativas GDPR (*General Data Protection Regulation*), que dictan una serie de pautas sobre cómo las organizaciones deben gestionar los datos personales que poseen.

Uno de los fundamentos de estas normas es el derecho de todo usuario a solicitar a cualquier organización la **eliminación absoluta de todos sus datos** personales en un tiempo determinado. Éste es precisamente el primer punto de conflicto entre Blockchain y privacidad.

En definitiva, la privacidad es un principio/derecho esencial en cualquier sociedad libre. Por tanto, el objetivo a alcanzar consistiría en un control total de los datos personales por parte de todo individuo. Tradicionalmente, bases de datos centralizadas, tanto públicas como privadas, han almacenado todo tipo de información confidencial sobre particulares y/o organizaciones sin notificación previa. Satoshi Nakamoto, fundador(es) del protocolo Bitcoin, implementó la capa de red sin especificación de identidad requerida, de modo que no es necesario proporcionar ningún tipo de dato personal con el fin de descargar el *software* de Bitcoin. No existe ningún tipo de referencia a identidad en las transacciones, debido al hecho de que las **capas de identificación y verificación están separadas de la de transacción** en éste y prácticamente todo protocolo basado en Blockchain.

Las transacciones tradicionales (transferencias, subscripciones, pagos con tarjeta de crédito) han estado caracterizadas por todo lo contrario: modelos centrados en la identidad, lo que ha desembocado en robos/pérdidas de datos masivos por parte de gigantes de la industria como JPMorgan Chase o Sony.

En la Blockchain [12], los participantes tienen la opción de mantener un grado considerable de anonimato, sin verse obligados a proporcionar datos detallados de su identidad ni almacenar los mismos en una BD centralizada y administrada por un tercero. Por lo tanto, los protocolos basados en Blockchain presentan un alto nivel de flexibilidad en términos de privacidad, aportando un mayor control y gestión de datos personales tanto a particulares como a organizaciones.

E) SEGURIDAD

Blockchain ha sido considerado por muchos como un paradigma con grandes virtudes en términos de ciberseguridad. A pesar de que factores como la descentralización o la

transparencia podrían favorecer la protección de los datos en esta red, afirmar la absoluta seguridad de ésta sería insensato y equívoco. Como se ha podido confirmar a lo largo del tiempo, el surgimiento de nuevos estándares de seguridad y tecnologías revolucionarias ha implicado asimismo la aparición de nuevos tipos de ciberataques y amenazas que afrontar.

En cuanto a ciberseguridad en la Blockchain, se podría decir que, *grosso modo*, existen dos amenazas o posibles tipos de ataque: locales o generales.

A diferencia de cualquier infraestructura centralizada, el carácter distribuido de la Blockchain favorece la ineficiencia de ataques locales dentro de la red. Cada nodo de ésta es responsable de la seguridad de sus propias transacciones, seguridad que es implementada de manera independiente al resto de integrantes de la red y puede basarse en multitud de métodos/estándares diferentes. Este contexto difiere significativamente del que se ha dado tradicionalmente, en el que el hacker sólo necesitaba encontrar un punto de acceso a la red de una entidad para, básicamente, hacerse con toda su información.

Con respecto a los ataques de carácter generalizado sobre una red Blockchain, cabe recordar la complejidad que esto implica debido al principio de persistencia de los bloques de la misma. Los esfuerzos que conllevaría este tipo de ataque se verían multiplicados, siendo estos directamente proporcionales a la antigüedad del bloque objetivo dentro de la totalidad de la red.

No obstante, existen multitud de factores adicionales a la hora de analizar la seguridad dentro de la Blockchain. Estos, junto con las amenazas/debilidades que puede presentar esta nueva tecnología, serán objeto de estudio al final del presente capítulo.

F) CONFIANZA

Blockchain está suponiendo una revolución a la hora de aportar confianza en todo aspecto dentro de una red distribuida empresarial. Esto no surge necesariamente a partir de una desconfianza tradicional entre miembros de una red; no obstante, con Blockchain en realidad no se tiene por qué confiar [8], facilitándose así, por ejemplo, transacciones múltiples entre

distintos participantes. Esto es posible gracias a la confluencia de los principios recién mencionados y aspectos adicionales, como los mencionados en el siguiente gráfico:

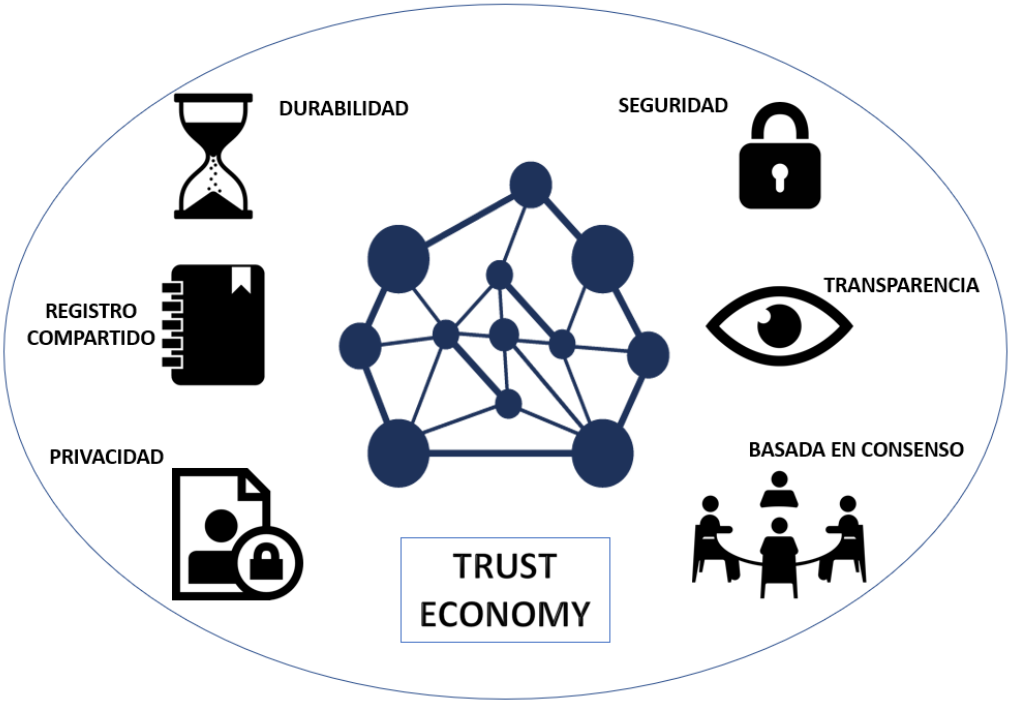


Figura 8. Blockchain como Garante de Confianza

Como ya se ha ido anticipando, los principios de la Blockchain mencionados en este capítulo son, en sí mismos, los grandes facilitadores de confianza de este paradigma. La transparencia, auditabilidad, inmutabilidad y seguridad (entre otros) que aporta este tipo de red distribuido ha dado lugar a lo que hoy se conoce como *trust economy* [13]. Este fenómeno hace posible que cada individuo/participante (no un tercero) decida qué información digital se almacena en la cadena, y el modo en qué se hará uso de la misma.

Fundamentalmente, los costes y esfuerzos en confianza [14] se ven estrictamente recortados debido a la descentralización de la contabilidad mediante el elemento del **registro compartido**. Cualquier miembro de la red puede actualizar el registro, siguiendo la norma dictada por el **protocolo de consenso**, aspecto en el que se profundizará en la siguiente sección del capítulo. De este modo, la comunidad de participantes de la red pasa a constituir globalmente el mecanismo de registro y regulación. En caso de que se diera un intento de

violación del *consensus protocol*, la nueva entrada al registro seria automáticamente rechazada por el total de la red. Esto supone una clara solución, por ejemplo, a la eliminación fraudulenta de libros de cuenta que se ha ejecutado tradicionalmente en industrias de todos los rangos posibles en términos de tamaño y/o poder.

Igual que el surgimiento de Internet y la WWW supuso un cambio radical en la comunicación entre individuos, se podría afirmar que la Blockchain puede traer consigo una nueva revolución en términos de intercambio de información y mecanismos de confianza. Generadores de *trust* tradicionales (bancos, servicios de *rating*, órganos reguladores, cartas de crédito, abogados, etc.) podrían dejar de ser necesarios. No obstante, como ocurre en todo paradigma tecnológico, la ausencia total de **gobernabilidad** resulta complicada y, con frecuencia, conflictiva.

5.1.2 TECNOLOGÍAS Y PROTOCOLOS

A pesar de que el desarrollo de aplicaciones basadas en Blockchain (o Dapps) supone un aspecto importante y necesario dentro de esta nueva tendencia tecnológica, es cierto que la concentración mayoritaria de valor en este paradigma se encuentra en sus **protocolos**.

Los protocolos de la WWW (HTTP, TCP/IP, etc.*)* resultaron fundamentales en la revolución que este paradigma trajo consigo. No obstante, el trascurso del tiempo ha demostrado que el verdadero valor en este caso reside en la capa de aplicaciones y el manejo de datos. Resulta evidente afirmar que los altos niveles de ROI e influencia hoy en día se concentran dentro del área de desarrollo e inversión en aplicaciones. No obstante, no se podría señalar lo mismo en el caso de la Blockchain.

De hecho, el resultado de esta comparación queda invertido en el nuevo paradigma. Una clara señal de este hecho puede extraerse de una de las principales redes Blockchain: Ethereum. Se estima que el valor del protocolo alcanzaba el billón de dólares antes incluso de que se desarrollará una aplicación con cierto nivel de influencia basada en el mismo [15].

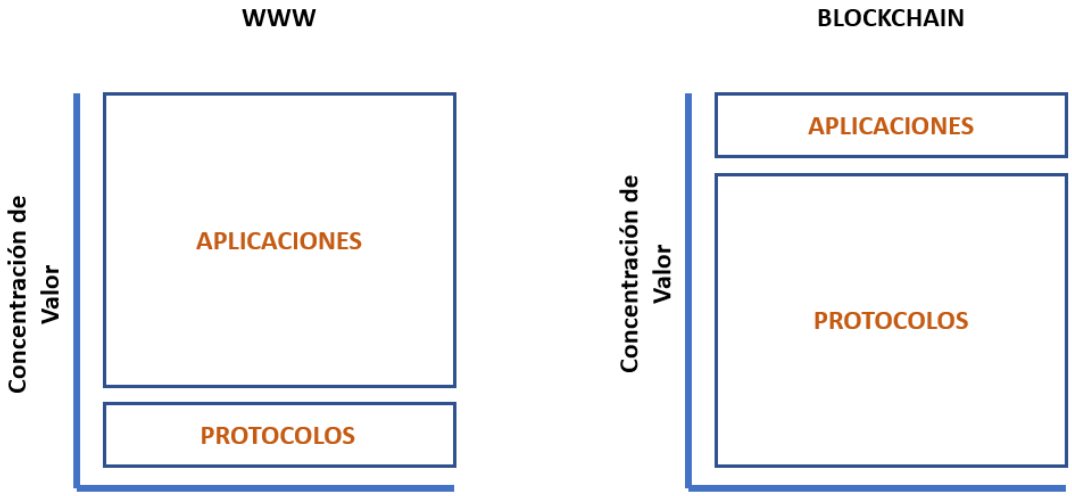


Figura 9. Residencia del Valor en WWW/Blockchain

Esencialmente, existen dos aspectos principales que generan este nivel de valor en los diferentes protocolos (redes) de la Blockchain:

En primer lugar, la e**xistencia de una capa (registro) abierta, en la que los datos son compartidos** y distribuidos alrededor de la red descentralizada, reduce considerablemente cualquier tipo de restricción de entrada. Esta propiedad de todo protocolo Blockchain genera sobre el mismo una atmósfera de creación de aplicaciones/servicios en constante cambio y con altos niveles de competitividad, generándose así altas expectativas en términos de calidad y/o innovación.

Por otro lado, el **carácter "tokenizado"** de estos protocolos también resulta fundamental en la generación de valor por parte de los mismos. La influencia de los *tokens* puede entenderse mediante la descripción del siguiente bucle:

1. Cuando el valor de un *token* sufre una subida, se genera un gran número de especulaciones, provocándose un incremento en el número de interesados por el protocolo.
2. Gran parte de estos interesados invierten en el protocolo y construyen servicios/aplicaciones sobre el mismo, contribuyendo a un nuevo aumento del valor

de éste. Asimismo, esta nueva subida del token capta la atención de nuevos inversores y/o emprendedores, incrementándose tanto las adquisiciones como la innovación sobre esta red (o protocolo).

A nivel financiero/económico, este bucle presenta un **riesgo** importante. En caso de que el interés por el token/criptomoneda creciera mucho más rápido que la oferta de éste provocaría una situación de **burbuja**, desembocando ésta en una caída en picado del *token* (como ya ha ocurrido en múltiples ocasiones).

No obstante, esta **volatilidad es altamente favorable en términos de innovación tecnológica**. Las rápidas subidas del *token* provocan la creación de capital, parte del cual se destina a innovación y desarrollo de aplicaciones. En el caso de los grandes descensos del valor del *token*, se procede a una adopción de la tecnología con fines a largo plazo, con el objetivo de obtener beneficios del servicio ofrecido por la capa de aplicaciones hasta que estos se puedan generar por una nueva subida de la criptomoneda.

Una vez aclarado el valor que aporta esta capa en el paradigma, se procede a señalar los **cinco protocolos principales en el ámbito de la Blockchain**, junto con algunos aspectos claves de su funcionamiento:

- **BITCOIN**

  Sus inicios se remontan al año 2008, cuando un grupo de expertos bajo el pseudónimo **Satoshi Nakamoto** elaboró la primera publicación relacionada con el ámbito de la Blockchain. A pesar de que el análisis en profundidad de esta conocida criptomoneda no se encuentra dentro de los objetivos de la presente tesis, resulta interesante señalar una serie de características fundamentales de esta cadena.

  En primer lugar, uno de los aspectos principales de esta Blockchain es su carácter **público**. Este tipo de redes se conoce también como Blockchain **sin permisos**: cualquier organización/individual puede pasar a formar parte de la red distribuida y comenzar a contribuir en la verificación de bloques.

En contraste, existen las llamadas Blockchain **privadas (o con permisos)**, en las que sólo el propietario de la cadena tiene la autoridad de aceptar o rechazar nuevos integrantes. Este otro tipo de cadenas puede resultar adecuada en casos de uso con requisitos muy estrictos en términos de protección de datos, ya que tanto el acceso a los detalles de transacciones como la participación están más restringidos.

Por otro lado, el protocolo Bitcoin cuenta con una serie de componentes tecnológicos subyacentes [16]: un hash criptográfico, firmado digital, encriptado mediante clave pública y/o privada, distribución P2P (*peer to peer*) y un algoritmo de **consenso POW**, elemento que se describirá en detalle en la siguiente sección del capítulo.

En cuanto a las bases del funcionamiento de esta red distribuida, se puede señalar que coinciden con la explicación genérica e introductoria que se aportó anteriormente en la presente memoria: uno de los miembros de la red anuncia su intención de realizar una nueva transacción, de modo que la creación del nuevo bloque de la cadena requerirá tanto los detalles de la transacción mencionada como una referencia al último bloque incluido. Este último bloque no es conocido en un principio, por lo que un complejo proceso de criptografía (**minado**) será necesario para encontrar esta referencia.

A pesar de que Bitcoin constituye una clara revolución tecnológica, sus propiedades e infraestructura incluyen una serie de **riesgos** a tener en cuenta:

- El hecho de que todo nodo de la red Bitcoin tenga que contener toda la información presente en la red y deba participar en el proceso de verificación, resulta altamente favorable en términos de fiabilidad, pero también genera problemas de **escalabilidad**. De hecho, para muchos la Blockchain de Bitcoin no resulta viable como sistema de procesamientos de pago a nivel mundial.
- Las altas demandas de poder computacional de la red Bitcoin han provocado además problemas de **centralización**. Mientras que el objetivo inicial de esta cadena era la descentralización y la distribución del poder entre los distintos participantes, actualmente la inmensa mayoría de mineros se encuentran en China, donde se emplea la potencia de plantas hidráulicas para llevar a cabo

este proceso de verificación. Esto provoca la vulnerabilidad del estado de esta Blockchain a voluntad de este grupo mayoritario de participantes.

- Finalmente, el valor del *token* de la Bitcoin presenta riesgos de carácter financiero debido a su enorme **volatilidad**. A pesar de que el valor actual de esta criptomoneda es de unos 7000 €, hace apenas un año era de unos 1600 € (una quinta parte, aproximadamente).

Estos riesgos y otros aspectos de diversa índole han provocado la creación de diversos protocolos basados en Blockchain durante los últimos años.

- **ETHEREUM**

Mientras que la comúnmente conocida Bitcoin fue creada con el único fin de optimizar el procesamiento de pagos y transacciones de carácter financiero, Ethereum nació con expectativas significativamente superiores tanto en utilidad como diversidad. Su desarrollo, propuesto inicialmente por **Vitalik Buterin** a finales de 2013, constituye una Blockchain de **uso generalizado**, probablemente la más potente que puede ser empleada actualmente.

A pesar de contar con algunas diferencias respecto a su predecesora Bitcoin, ambos protocolos de Blockchain también comparten características similares: ambas son Blockchain públicas (sin permisos), cuentan con los mismos elementos tecnológicos subyacentes (firma digital, red P2P, etc.) y emplean el algoritmo de consenso POW.

Ethereum cuenta con su propia **criptomoneda nativa**, el **Ether**, que difiere con respecto al bitcoin en multitud de aspectos. En primer lugar, *a priori* la oferta de Ether dentro de esta Blockchain es ilimitada. Además, esta red permite al usuario el desarrollo de **criptomonedas propias** que se ajusten lo máximo posible a sus necesidades.

A pesar de que en el siguiente capítulo de la memoria se profundizará en este protocolo de Blockchain y se analizará una implementación basada en el mismo, cabe señalar que los altos niveles de versatilidad que aporta esta red son posibles gracias, fundamentalmente, a los siguientes elementos [16]:

- La **EVM** (*Ethereum Virtual Machine*), una plataforma Blockchain que permite a cualquier desarrollador la implantación de su propia cadena e, incluso, su propia criptomoneda.
- Los **Smart Contracts**, programas o piezas de *software* que, básicamente, definen una serie de reglas que rigen el flujo de transacciones dentro de la red ("haz esto si se da esta condición"). En la plataforma Ethereum estos contratos están implementados usando su lenguaje de programación propio **Solidity**.
- Las **Dapps**, aplicaciones descentralizadas ejecutadas sobre la Blockchain y contenedoras de Smart Contracts. Toda Dapp debe ser *open-source*, hacer uso de algún tipo de criptomoneda, contar con algún tipo de consenso que regule los cambios en la cadena y emplear algún tipo de algoritmo criptográfico generador de *tokens*.

El hecho de que Ethereum gane a Bitcoin tanto en versatilidad con en velocidad de procesamiento no implica que este protocolo carezca de limitaciones. De hecho, esta Blockchain sigue careciendo de las velocidades y capacidades suficientes para programar un sistema íntegramente sobre la misma. Asimismo, los sistemas de almacenamiento de datos actuales son superiores en varias magnitudes en términos de velocidad. Además, Ethereum no cuenta con muchas prestaciones en términos de privacidad, y es considerado menos seguro que Bitcoin por multitud de expertos.

- **RIPPLE**

Este protocolo comparte numerosas similitudes con Bitcoin y Ethereum. No obstante, cuenta con una serie de características específicas ideadas con el fin de satisfacer su principal caso de uso: **la realización de transacciones financieras globales de un modo rápido y económico**. El usuario puede elegir si realizar esta transferencia empleando la criptomoneda nativa del protocolo, **XRP**, o cualquier tipo de dinero fiduciario (euro, dólar, yen, etc.).

A pesar de que este protocolo ha generado y genera cierto escepticismo por parte de algunos expertos, actualmente algunas de las entidades bancarias líderes a nivel mundial (Santander, Unicredit, Bank of America, etc.) han comenzado a hacer uso de su *suite* **RippleNet** [17], una red que aprovecha las propiedades de XRP y otras tecnologías *open-source* basadas en Blockchain para agilizar estos procesos y mejorar la experiencia de usuario.

- **HYPERLEDGER**

Hyperledger es un proyecto propuesto inicialmente por la Fundación Linux, y al que se han unido desde entonces numerosas entidades líderes de diversos sectores (IBM, Intel, JP Morgan, SAP, etc.) [18].

Esta iniciativa está destinada fundamentalmente a desarrolladores de soluciones *open-source* basadas en Blockchain, aportando estándares y herramientas que faciliten una adopción real e industrial de implantaciones de registros compartidos basados en el paradigma Blockchain. Asimismo, proporciona tecnología modular y APIs que favorezcan la interoperabilidad de las aplicaciones Blockchain y reduzcan significativamente los niveles de complejidad a la hora de implantarlas [8].

La gran novedad de este protocolo/proyecto es la introducción de **permisos**, es decir, globalmente constituye una **Blockchain privada**: sólo los participantes u organizaciones aceptados en el proyecto pueden unirse a la cadena. Además, en este caso los distintos participantes de la red adoptan **distintos roles en el proceso de consenso** en función de diversos factores.

Los dos aspectos recién mencionados resultan adecuados en multitud de aplicaciones industriales, debido a que la mayoría de organizaciones prefieren que su propia información no sea visible por un tercero (ni siquiera por ciertos miembros de la propia entidad). Además, las distintas plataformas y soluciones aportadas por este proyecto aportan niveles de velocidad y escalabilidad adecuados para una gran cantidad y diversidad de casos de uso.

Finalmente, cabe mencionar que Hyperledger **no cuenta con una criptomoneda nativa**. No obstante, está constituido por una gran variedad de proyectos (Fabric, Iroha, etc.) y herramientas (Composer, Cello, etc.) que lo convierten en una alternativa atractiva y prometedora en multitud de sectores empresariales. Alguno de estos proyectos será analizado con mayor detenimiento en el próximo capítulo, dedicado al análisis de implementaciones técnicas de la Blockchain.

- **CORDA**

Este último protocolo ofrece fundamentalmente una tecnología de registro distribuido para fines **financieros**. Fue fundado por el Consorcio R3 en 2015, con integrantes tales como BBVA, JP Morgan o Goldman Sachs [19]. Esta solución Blockchain está diseñada principalmente para operar en las siguientes situaciones:

  - En transacciones sometidas a estrictas medidas regulatorias, mediante la introducción de estas medidas legales en Smart Contracts.
  - Para el establecimiento de acuerdos entre entidades distintas.

Entre sus propiedades, cabe destacar que se trata de una Blockchain privada, no cuenta con una criptomoneda nativa y tiene un protocolo de consenso especial. En este proceso, CORDA mira al conjunto de transacciones, y divide los distintos nodos en roles. De este modo, se delega la validación únicamente a los participantes del acuerdo o la transacción.

5.2 COMPONENTES DE UNA SOLUCIÓN BLOCKCHAIN

Una vez descrito el funcionamiento básico de toda red basada en Blockchain, así como sus principios fundamentales y los protocolos/tecnologías de uso más extendido de este paradigma, es necesario hacer especial hincapié en los seis principales componentes que interoperan durante el funcionamiento de la cadena.

5.2.1 NODOS

Los nodos de cualquier Blockchain constituyen una pieza fundamental para mantener la integridad dentro de la cadena. Esencialmente, se considera nodo a **toda copia de una determinada Blockchain que existe en un ordenador u otro tipo de *hardware*** (servidores, *smartphone*, *tablet*) [20].

Como se puede intuir, el almacenamiento completo del historial de bloques y transacciones de una Blockchain suele requerir una gran cantidad de recursos computacionales disponibles. Por este motivo, ha surgido la diferenciación entre dos tipos de nodos [19]:

- Nodos Completos, aquellos que contienen absolutamente toda la historia de lo que ha ocurrido en la cadena desde su creación.
- Nodos Ligeros, que contienen únicamente una lista parcial de la cadena, y suelen estar linkados de algún modo a un nodo completo.

La cantidad de nodos activos en una Blockchain es directamente proporcional a la seguridad y disponibilidad de la red distribuida debido al hecho de que una distribución global de estos hace casi imposible un ataque que deje a la red fuera de servicio. Por ejemplo, **Bitcoin y Ethereum cuentan con entre 25 y 30 mil nodos completos**, factor que favorece notablemente a mantener la robustez y disponibilidad en estos dos protocolos.

Para configurar y ejecutar un nodo completo basta simplemente con descargar la última versión de su cliente. En el caso de Bitcoin, basta con instalar el “Bitcoin Core”, o “Satoshi Client”, teniendo en cuenta que la instalación de un cliente completo de este protocolo ocupa

un espacio superior a 65 Gb. La siguiente ilustración muestra un ejemplo de ejecución del cliente de Bitcoin en un entorno Linux:

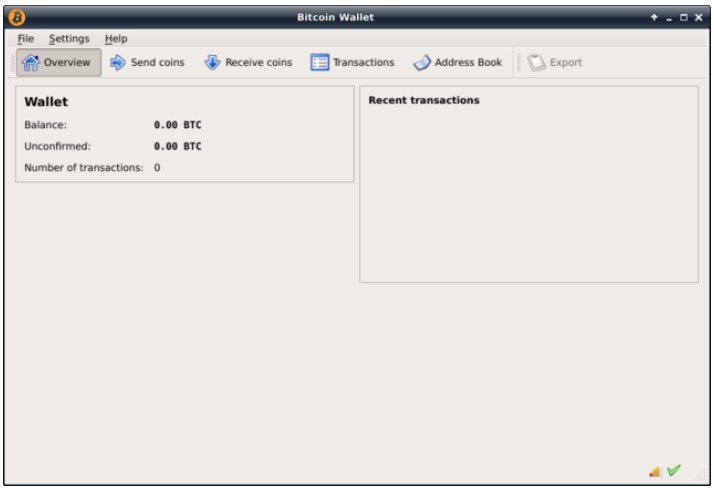


Figura 10. Bitcoin Core – Wallet bajo entorno Linux

Finalmente, cabe aclarar que no todo nodo completo realiza también la ya mencionada labor de minado, como se explicará en la sección dedicada a este importante procedimiento de la Blockchain.

5.2.2 REGISTRO COMPARTIDO

Probablemente el componente más distintivo y fundamental de este paradigma, consiste en un libro/registro distribuido, replicado, sincronizado y común a todos los miembros de la red que constituye una Blockchain. En él aparecen grabadas las distintas transacciones (intercambio de datos u otro tipo de activos) que se han llevado a cabo entre los miembros de la cadena a lo largo del tiempo.

Los participantes de esta red acuerdan y verifican cada una de las actualizaciones de este registro, sin que sea necesaria la existencia de una autoridad o mediador involucrados en este proceso [21]. Cada grabación en el registro tiene, además de la firma criptográfica única ya mencionada anteriormente, un **timestamp** o marca de tiempo que señala el momento de creación del bloque en cuestión.

La siguiente tabla [19] describe las principales diferencias existentes entre este tipo de registro y el tradicional registro/libro centralizado:

Registro Distribuido (Redes basadas en Blockchain)	**Registro Centralizado (Redes Empresariales Tradicionales)**
Información ingresada por consenso	Mayor posibilidad de Conflictos
Inmutabilidad y Persistencia	Fácil Corrupción/Fraude
Carácter Distribuido	Alta vulnerabilidad a fallos (*single point)*
Peer – to – peer	Pasarelas/accesos + entidad central
Verificación mediante Criptografía	Diferentes posibilidades para Verificación
Tantas copias como nodos en la cadena	Número de *back-ups* limitado

Tabla 3. Registros/libros Contables Distribuidos vs Centralizados

Además, los registros distribuidos (*shared ledgers*) suelen clasificarse empleando la misma nomenclatura con la que se discriminan los tipos de redes Blockchain: registros **sin y con permisos**. La integridad es en ambos tipos garantizada por criptografía y reglas matemáticas, no por un elemento humano. No obstante, las dos modalidades presentan también notables diferencias [14].

En primer lugar, en los registros sin permisos cualquier equipo con suficientes recursos computacionales tiene la posibilidad de unirse a la red distribuida. Bitcoin y la mayor parte de criptomonedas cuentan con este tipo de registro.

Por otro lado, los registros con permiso no suelen incluir una criptomoneda nativa. Estos generalmente están constituidos por un grupo de organizaciones que necesitan un libro contable común, pero que son independientes entre ellas y/o no existe una confianza plena entre las mismas (fabricantes y proveedores, etc.). En este tipo de registro los participantes pueden visualizar exclusivamente el contenido del cual tienen autorización.

A pesar de que la tecnología de Registro Distribuido propio de la Blockchain cuenta con una gran cantidad de beneficios, su adopción supone también ciertos riesgos y desafíos [22]:

- Falta de madurez/consolidación.

  El hecho de que Blockchain no sea aún un paradigma con un extenso grado de aplicación en la industria provoca, entre otros, dudas sobre la robustez de sus aplicaciones ante altos volúmenes de transacciones. Además, también existe una preocupación generalizada ante el escaso desarrollo de estándares *software*/*hardware* basados en tecnología Blockchain, un elemento fundamental en implantaciones industriales. Proyectos como Hyperledger están trabajando en dar solución a este último factor.
- Escalabilidad y velocidad en las transacciones.

  Este problema se presenta fundamentalmente en las Blockchain públicas, siendo la red Bitcoin el mejor ejemplo de ello, dado que este protocolo permite procesar sólo entre cuatro y siete transacciones por segundo.
- Desafíos en Gobernabilidad

  A pesar de que Blockchain incluye intrínsicamente la ausencia de necesidad de un elemento central que regule el conjunto de acciones que se desarrollan durante el funcionamiento de la red distribuida, esto trae consigo ciertos conflictos a la hora de aplicar el paradigma para determinados fines o instituciones.
- Desafíos legales y/o regulatorios

  Estos surgen a la hora de intentar incluir estándares industriales, legislaciones y medidas de privacidad sobre ciertas implementaciones de este registro compartido. La regulación de de registros compartidos públicos y sin permisos resulta especialmente compleja debido a la imposibilidad de que exista una entidad legal que controle la red distribuido. En el caso de las redes privadas, la inclusión de un administrador/propietario del sistema resulta más viable, de modo que sí existiría la posibilidad de integrar *frameworks* regulatorios en la cadena.

5.2.3 SMART CONTRACTS

Conceptualmente, los *Smart Contracts* característicos de la tecnología Blockchain comparten numerosas similitudes con los contratos y acuerdos con los que individuales y organizaciones han trabajado tradicionalmente. La principal diferencia es que, en este caso, el contrato está elaborado en un formato 100% digital.

Esencialmente, un *Smart Contract* es un *script* o pieza de *software* integrado en el registro distribuido (en la Blockchain) que incorpora las reglas o condiciones lógicas que marcan el intercambio de activo, las verifica y se ejecuta automáticamente en todos los nodos de la red distribuida [22]. De este modo, se permite una ejecución autónoma de las distintas transacciones según lo estipulado en el contrato, y un almacenamiento de estas en la cadena de bloques.

A pesar de que la mayoría de protocolos de la Blockchain soportan la inclusión de *Smart Contracts*, **Ethereum** es con diferencia la Blockchain con más orientación y flexibilidad en la implementación de este tipo de acuerdo digital. Fundadores de esta Blockchain crearon también el mayor lenguaje de programación orientado a *Smart Contracts*: **Solidity.**

A pesar de que el uso de *Smart Contracts* es fundamental para permitir que las distintas transacciones dentro de la cadena se ejecuten según las reglas de negocio de la entidad sin necesidad de intervención externa, su puesta en marcha conlleva importantes inversiones en términos de tiempo (complejidad) y dinero. Además de elaborar el código del contrato, se debe dar una importancia especial a la auditoría y la **búsqueda de *bugs*** dentro de la implementación realizada, pues un error mínimo en la misma puede tener consecuencias significativamente negativas para el negocio regulado por la cadena en cuestión.

A la complejidad y el carácter crítico propios de cualquier desarrollo de un *Smart Contract* se le ha de sumar el alto coste que esta acción suele suponer. Este coste suele variar entre los $7000 en implementaciones simples hasta los $100000 en las más complejas [23].

La forma de desarrollar (programar) un contrato inteligente debe tener en cuenta además los costes de desplegar y ejecutar el mismo dentro de la Blockchain. La **tarificación de este**

despliegue depende de la complejidad del contrato, midiéndose este factor en función del número de condiciones, operadores, llamadas a funciones, etc.

En el caso de la red Ethereum, esta tarificación se realiza a través del **conteo de gas**, unidad que permite el cálculo del coste de transacciones en Ether (criptomoneda). La suma de todo el coste en gas correspondiente a todas las operaciones/instrucciones ejecutadas durante una transacción constituye el coste en gas de esta. De este modo, se produce gasto/tarificación en dos tipos de situaciones:

- **La creación y despliegue del contrato**, que no deja de ser una transacción más que debe ser distribuida por toda la Blockchain. Esta acción es altamente costosa al ser necesaria la ejecución de todas las operaciones descritas en el *Smart Contract*.
- **Las llamadas al contrato**¸ que tendrán un coste equivalente a la suma de los costes de las operaciones del contrato que se necesitaron ejecutar para realizar la transacción.

Una vez calculado el coste en gas de una transacción, este se multiplica por el valor del gas en Ether (variable), obteniéndose el coste en la criptomoneda. El precio/valor del gas es, técnicamente, una propuesta de un nodo que desea ejecutar una transacción hacia los mineros de los cuales depende la inserción de esta en la Blockchain [24]. El Anexo A de la memoria contiene la tabla de tarificación de las distintas operaciones en Gas, incluida en el artículo de referencia de la red Ethereum: el *Yellow Paper*.

El capítulo siguiente hará referencia y analizará implementaciones concretas de Smart Contracts, señalando además las diferencias en el despliegue y/o almacenamiento del mismo en la Blockchain en función del protocolo empleado.

5.2.4 CONSENSOS

Generalmente, un algoritmo de consenso es un mecanismo empleado para lograr un acuerdo sobre el valor de un dato, aportándose así fiabilidad a una red o *dataset* determinados.

En el caso particular de la Blockchain, la naturaleza distribuida de la red requiere que los distintos nodos o participantes alcancen un consenso con respecto a la validez de nuevas entradas de datos en el registro compartido, mediante el seguimiento de una serie de normas prestablecidas. Por tanto, en esencia, el protocolo de consenso es necesario para **determinar si una transacción es o no legítima**, haciendo uso de un método criptográfico específico designado para esa cadena en concreto. Además, este componente de la Blockchain permite gestionar conflictos existentes cuando, por ejemplo, **dos nodos proponen transacciones distintas sobre un mismo activo** [22].

Previo a la aparición del Bitcoin y la tecnología Blockchain, numerosas arquitecturas distribuidas y descentralizadas fracasaron tras no saber afrontar ciertos desafíos. Probablemente el más destacado de estos es el conocido como **Problema de los Generales Bizantinos (*Byzantine Generals Problem, BGP)***.

Cuatro (n) divisiones del imperio bizantino con sus correspondientes generales acampan alrededor de una ciudad. Existen dos opciones viables de actuación: un ataque simultáneo de todas las divisiones, o una retirada; y esta decisión debe ser decidida y comunicada entre divisiones a través de mensajero. El problema surge cuando aparecen **divisiones traidoras** como la número #4, que envía una decisión distinta a cada una de sus divisiones contiguas (#3 y #1), tal como muestra la siguiente ilustración:

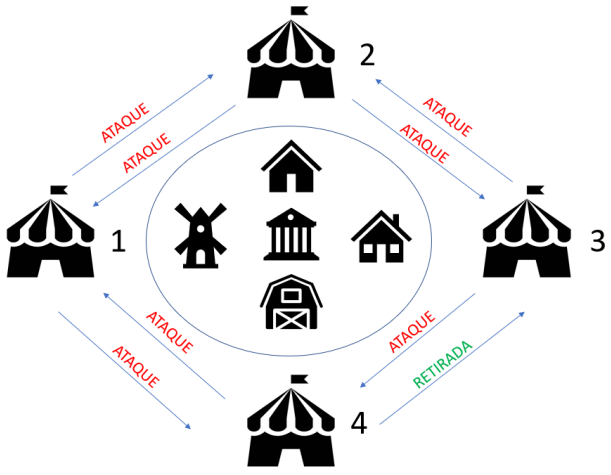


Figura 11 Explicación gráfica del BGP

La situación anterior provoca que, debido a la traición de la cuarta división, la acción que va a ejecutar la división leal #3 sea también una incertidumbre total. Por tanto, las divisiones necesitan un mecanismo de consenso que garantice que todas las unidades leales compartan un mismo plan de acción y, además, la minoría de traidores no provoque que unidades leales adopten el plan erróneo.

Esta situación presenta una evidente analogía con la Blockchain: una enorme red distribuida con multitud de nodos de distinta naturaleza y procedencia. Los mecanismos de consenso constituyen la solución para evitar intentos de manipulación fraudulenta de transacciones dentro de la cadena. Algunos de los principales son [25]:

- **Proof of Work (PoW)**

  Este es el protocolo de consenso empleado en la red de Bitcoin. Como su nombre indica, este mecanismo requiere que los participantes de la red inviertan altos niveles de capacidades de procesamiento. En concreto, consiste en una competición para resolver un reto criptográfico de alta complejidad, proceso conocido como **minado**. La siguiente sección de este capítulo describe con detalle este proceso, fundamental en el paradigma de la Blockchain.

  Aunque se trata de un mecanismo brillante, es cierto que consume altos niveles de potencia computacional, y que este consenso (y el carácter descentralizado de la red) puede verse corrompido por la concentración de mineros en grupos y zonas con recursos eléctricos a bajo coste, por ejemplo.

- **Proof – of – Stake (PoS)**

  En este mecanismo, la probabilidad de que un nodo sea elegido para la creación del siguiente bloque es directamente proporcional a la cantidad de criptomoneda con la que cuente. Una vez el nodo (denominado validador) crea el bloque y verifica la validez de las transacciones, la actualización de la cadena debe ser aceptada por la mayoría de nodos (todos o un grupo de ellos) antes de insertarse en la Blockchain.

  A pesar de que este mecanismo consume un menor nivel de capacidad de procesamiento, éste presenta también algunos riesgos:

  - Existe el mismo riesgo contra el carácter descentralizado de la cadena, aunque el riesgo de que exista un individuo/organización con más de la mitad del total de criptomonedas es menos probable a que lo exista con más de la mitad del poder computacional (economías de escala, pooled miners, etc.).
  - El hecho de que un validador con un mayor nivel de *stake* (porción del total de criptomonedas) tenga más probabilidad de crear el nuevo bloque provoca el riesgo de que se cree un círculo vicioso que sólo beneficie a los nodos más "ricos" de la Blockchain.

- **Proof – of – Activity (PoA)**

  Grosso modo, este mecanismo es una mezcla entre el PoW y el PoS, en el que la recompensa final es repartida entre el minero ganador y el grupo de validadores que hicieron el *commit* final del bloque. Este mecanismo no está siendo empleado por ninguna de las grandes Blockchain en la actualidad.

- **Practical Byzantine Fault Tolerance Algorithm (PBFT)**

  Este último método, el más novedoso, constituye una mejora respecto a los tres anteriores, sobre todo en términos de **baja latencia**. Este factor es de alta importancia para numerosas aplicaciones industriales (cualquier plataforma basada en el flujo continuo de activos digitales), de ahí que sea el empleado por Blockchains como Hyperledger o Ripple. La lógica detrás de este protocolo será explicada con mayor detalle en el futuro apartado dedicado al análisis del proyecto Hyperledger.

5.2.5 TRANSACCIONES

En esencia, una transacción se define como el **cambio de propiedad de un activo (tangible o no) de un individuo/organización a otro**. Dentro del paradigma de la Blockchain, esta definición coincide, siendo este activo una criptomoneda o un bien *tokenizado*. Por lo general, una transacción comienza cuando uno de los dos nodos "anuncia" este cambio de propiedad; este anuncio debe estar firmado digitalmente por este participante e indicar la localización/dirección del beneficiario.

Aunque generalmente se suele asociar el término transacción con transferencias de bienes financieros (como ocurre en la archiconocida Bitcoin), existe una gran variedad de cadenas que permiten la publicación de **transacciones en formato de programas/tareas**, unidades que son almacenadas en lo que se conoce como registros o **bloques.**

Estos bloques se ejecutan, implementan y almacenan en la red distribuida una vez hayan sido validados por los integrantes de la cadena. La siguiente ilustración muestra una versión simplificada de la estructura de los bloques de una Blockchain:

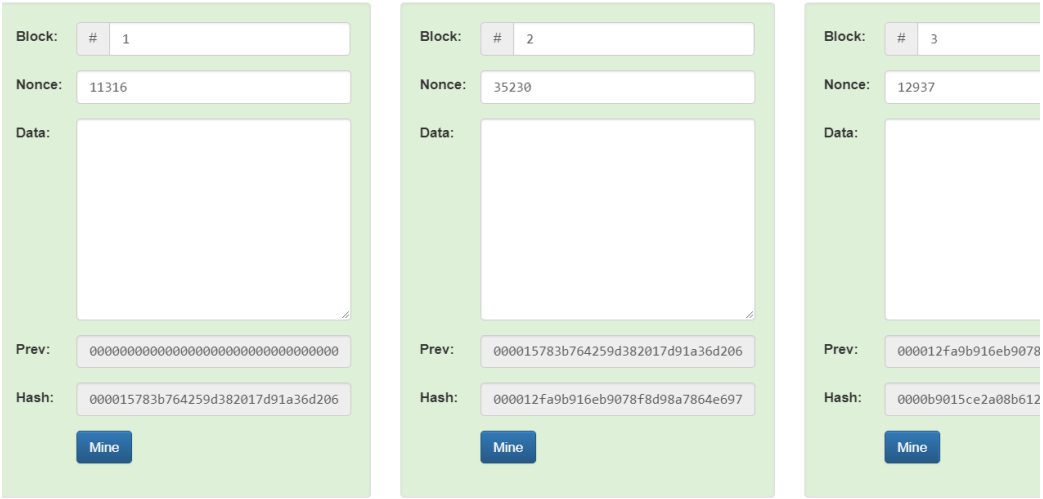


Figura 12. Bloques. Elementos Fundamentales [26]

Por lo tanto, los tres componentes fundamentales presentes en cualquier bloque son:

- El **hash del bloque anterior**
- El **hash del propio bloque,** obtenido tras ejecutar un algoritmo concreto (sha-256, etc.) sobre una serie de parámetros (Nonce+ hash anterior, principalmente).
- Un campo de **datos**, donde se almacenan las distintas transacciones del bloque.

Futuras secciones del proyecto describirán con mayor detalle la estructura de estos bloques y los procedimientos de obtención del hash.

5.2.6 BLOQUES. TIPOS

Aunque ya se ha hecho referencia con anterioridad a los bloques como componente de la Blockchain, resulta interesante realizar una clasificación de estos en función de distintos factores, tales como el resultado del proceso de consenso y minado dentro de la cadena. Dicho esto, dentro del espacio virtual de una Blockchain pueden encontrarse principalmente **tres tipos de bloques [38]:**

En primer lugar, se pueden encontrar **bloques huérfanos**. Un bloque es clasificado de este modo si cumple con todos los requisitos necesarios para ser añadido a la Blockchain pero ha sido rechazado. Esto ocurre **cuando dos mineros crean un bloque al mismo tiempo.** Estos bloques, antes de incorporarse a la Blockchain, deben recorrer toda la red de nodos con el fin de asegurar su validez. Si el resto de nodos reciben dos bloques procedentes de distintos mineros, validaran aquel con un mayor *Proof – of – Work*, es decir, aquel cuyo minado haya consumido una mayor cantidad de recursos computacionales. Cabe señalar que, en el caso de los bloques huérfanos, el **minero en cuestión no recibe ningún tipo de recompensa** por la creación del mismo.

El segundo tipo de bloque, conocido como bloque tío (o *uncle*), está más asociado a la red Ethereum, y es, fundamentalmente, un bloque huérfano por el cual el minero **sí recibe una recompensa**, aunque está es menor a la que recibiría si el bloque se hubiera incorporado en la cadena. Este mecanismo es una buena medida para fomentar el "minado en solitario" frente a las agrupaciones de mineros y la utilización de *hardware* especializado en este proceso. Futuras secciones tratarán con más detalle este fenómeno.

Por último, toda Blockchain cuenta con un **bloque génesis**, el primero creado en la historia de esa cadena, contenedor de todos los fundamentos en los que se basa el funcionamiento de la misma.

5.2.7 PARTICIPANTES

Con el fin de concluir con la descripción de los principales componentes dentro de una red Blockchain, se procede a señalar los distintos tipos de participantes que se pueden encontrar en la misma [22] [8]. Esta clasificación hace mayor **referencia a las redes privadas con permisos**, debido a que una Blockchain pública incluye por definición la ausencia de roles.

En primer lugar, las Blockchain empresariales (privadas) suelen incluir un **administrador/regulador**, rol que incluye el control de nuevos accesos a la red y la supervisión de las transacciones que se dan en la cadena. En ocasiones este individuo/nodo no cuenta con el derecho de ejecutar transacciones.

La mayoría de nodos son **usuarios genéricos** de la red, que cuenta con permiso de acceso a la cadena y propone nuevas transacciones para su inclusión en el *shared ledger*. En la mayoría de las aplicaciones industriales, este tipo de usuario interacciona con el sistema sin realmente percatarse de toda la tecnología Blockchain (registro compartido, consensos, minados, etc.) que se ejecuta por debajo.

En algunas cadenas, existe una serie de **nodos validadores,** encargados de verificar y aceptar/rechazar las propuestas de inclusión de nuevas transacciones en el registro. En el caso de las Blockchain públicas, esta validación es siempre llevada a cabo por algoritmos de consenso descentralizado.

En algunas aplicaciones industriales tokenizadas, existen nodos de red responsables de la **creación de este activo digital**. Este rol es nuevamente típico solo de algunas cadenas privadas, ya que la mayoría de Blockchains públicas cuentan con mecanismos autónomos que generan nuevos *tokens* según una serie de normas predefinidas.

Finalmente, la mayoría de redes empresariales requieren la existencia de **nodos con un rol auditor**, con permisos para visualizar por completo lo que ocurre dentro del registro compartido, aunque no modificarlo.

5.3 ASPECTOS DE SEGURIDAD

Para finalizar el recorrido por los aspectos fundamentales de toda Blockchain, esta sección procede a describir una serie de mecanismos esenciales en su funcionamiento y en la aportación de robustez y seguridad a la red. Además, se concluirá con una breve referencia a algunos de los principales riesgos de ciberseguridad experimentados por este paradigma hasta la fecha.

5.3.1 CRIPTOGRAFÍA Y MINADO

En toda Blockchain, se conoce como minado al proceso a través del cual se verifican las distintas transacciones con el fin de ser añadidas al registro compartido.

Grosso modo, este procedimiento consiste en la inserción de una serie de transacciones en bloques y la resolución de un *puzzle*/algoritmo complejo en términos computacionales. El nodo o **minero que antes consigue resolver el puzzle es aquel con derecho a insertar el siguiente bloque de la cadena y, además, obtener una recompensa**. Estas recompensas pueden ser criptomonedas, el coste de ejecución de las distintas operaciones del Smart Contract contenido en el bloque, etc.

Para entender el conjunto de acciones que existen detrás del proceso de minado, es necesario introducir el concepto de **hash criptográfico** [27]. Toda Blockchain está constituida por una serie de bloques sucesivos vinculados criptográficamente. Además, como ya se comentó anteriormente [Fig. 9], cada bloque contiene básicamente cuatro elementos: el número de bloque, el hash del bloque actual, el hash del bloque anterior y un campo de datos (transacciones, Smart Contracts, etc.). Básicamente, un hash es una cadena larga de caracteres que constituye una huella digital de un conjunto de datos determinados mediante el empleo de una función/algoritmo (SHA256 en Bitcoin, Keccak-256 en Ethereum, etc.). Estos algoritmos presentan ciertas propiedades que resultan de gran utilidad para el correcto funcionamiento de una Blockchain:

1) Son determinísticos, es decir, siempre se obtiene la misma salida si se aplica el algoritmo sobre la misma entrada.
2) La salida del hash es impredecible hasta que no se aplica sobre las entradas el algoritmo en cuestión.
3) Con que una de las entradas varíe muy ligeramente, el algoritmo de hash generará una salida completamente distinta.

Por lo general, **el hash de un bloque se calcula aplicando el algoritmo correspondiente sobre, como mínimo, tres entradas/*inputs*: el número de bloque, el hash del bloque anterior, y el campo de datos**. De este modo los distintos bloques quedan vinculados y la manipulación de cualquier campo de un bloque será detectada tanto por éste como por todos los siguientes (se producen cambios en el valor del hash, y los bloques quedan invalidados). La siguiente figura ilustra este fenómeno:

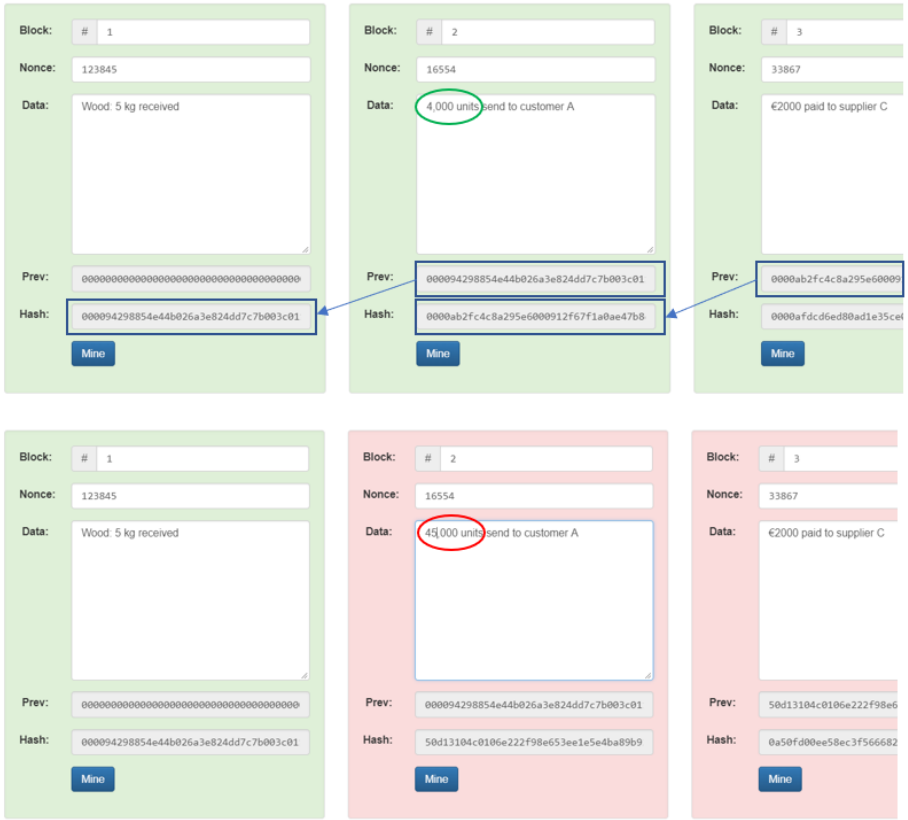


Figura 13. Manipulación de un Bloque de la Cadena

El hecho de que, como se puede observar en la figura, el cambio de cualquier dato en un bloque invalide tanto este como los siguientes garantiza el principio de **persistencia** de la Blockchain.

No obstante, con sólo describir este proceso (aparentemente bastante rápido), no queda clara la necesidad de altos niveles de capacidad de procesamiento para llevar a cabo el proceso de minado. Esto es porque se han obviado alguna de las entradas del algoritmo criptográfico (y elemento de todo bloque) que crea el hash, entre ella el **nonce (*number used only once)*.** Éste es el único elemento del bloque que **el minero puede usar libremente para cambiar el hash del bloque manteniendo el resto de éste intacto.**

En el caso del SHA256, cada hash tiene 64 dígitos hexadecimales, con 16 valores posibles cada uno. Por tanto, existen 16^{64} posibles valores para este hash. No obstante, la Blockchain cuenta con un mecanismo para limitar está gran cantidad de posibilidades, denominado comúnmente como ***target*.** Básicamente, el *target* (varía periódicamente) define el valor máximo que puede tener el *hash*, por lo que el **número de ceros a la izquierda de éste será inversamente proporcional al número de hashes posibles y directamente proporcional al nivel de complejidad de encontrar un hash válido.** Este es de hecho el *core element* del proceso de minado: se está compitiendo por ser el primero que encuentre un nonce que genere un hash válido en función del target marcado por la Blockchain, de modo que quien lo encuentre tendrá el derecho de añadir el siguiente bloque a la cadena y, además recibirá una recompensa.

El nivel de dificultad de este proceso (ceros a la izquierda del *target*) se va ajustando en función de la cantidad total de poder de procesamiento que se esté invirtiendo en el minado en cada momento, con el fin de mantener el ritmo de adición de nuevos bloques más o menos constante [28].

Sin embargo, el **nonce** resulta insuficiente para garantizar el proceso de minado por dos razones principales:

1) Un minero empleando un dispositivo *hardware* estándar sería capaz de iterar el rango completo de posibles valores del nonce en unos 40 segundos, lo que desembocaría en nuevas adiciones de bloques de manera demasiado frecuente.
2) En situaciones con *target* muy bajo, la posibilidad de encontrar un *hash* válido simplemente iterando el nonce es extremadamente baja.

Es por esto por lo que el algoritmo criptográfico que general el hash emplea además otro elemento del bloque en cuestión, un **timestamp** que se actualiza cada segundo, cambiando con esta frecuencia el valor del *hash* en función de los posibles valores del nonce.

Aunque esto resuelve el problema para un minero estándar, no lo hace para un grupo de **pooled miners**. Este término hace referencia a un gran número de nodos que trabajan conjuntamente en el proceso de minado, repartiendo proporcionalmente la recompensa en caso de que uno de ellos encuentre el *hash* vencedor. La siguiente figura presenta un ejemplo de *pooled miners*, una planta llena de dispositivos de minado trabajando colaborativamente:

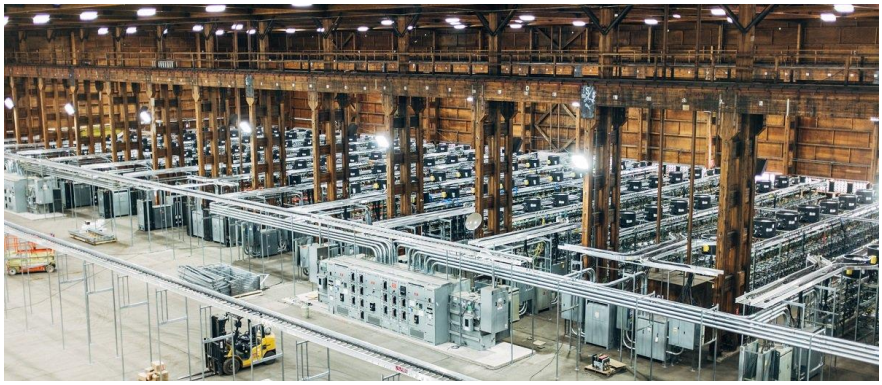


Figura 14. Pooled Miners

Estas agrupaciones son capaces de iterar el rango completo de valores del nonce en fracciones de segundo, por lo que la espera de un segundo para intentar de nuevo tras la actualización del *timestamp* supone un fuerte inconveniente. En estos casos, se procede a **"jugar" con la configuración del campo de datos/transacciones del bloque**.

Antes de ser añadidas a un bloque, el conjunto de transacciones de la cadena queda almacenado en una especie de "memoria previa", el **mempool**. Por lo tanto, los mineros

pueden ir variando las transacciones que escogen del mempool para formar el bloque, siendo éste un modo adicional de ir variando los valores del hash en la búsqueda de uno inferior al target.

Como conclusión, la siguiente figura señala los elementos que permiten a los mineros la resolución del *puzzle* criptográfico que se acaba de describir:

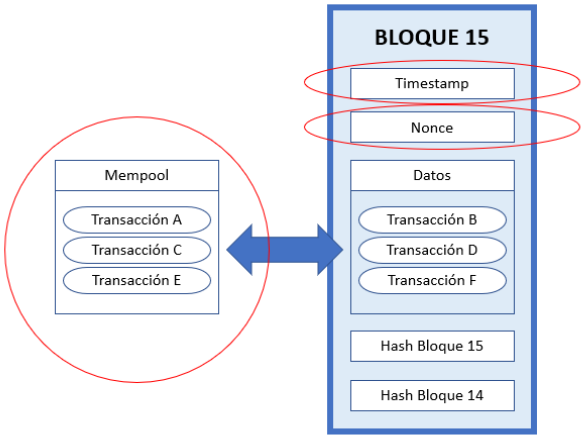


Figura 15. Minado. Elementos Principales

5.3.2 RIESGOS DE SEGURIDAD EN LA BLOCKCHAIN

Hasta el momento, se han señalado múltiples atributos que hacen de esta tecnología una opción idónea para una rápida detección de intentos de manipulación fraudulenta, una verificación descentralizada que previene abusos de poder por parte de terceros o administradores, etc.

Los altos niveles de dependencia en la tecnología y la red Internet de cualquier sistema IT a día de hoy han traído grandes beneficios pero también nuevas y frecuentes oportunidades de ataque para los cibercriminales. Por supuesto, la tecnología Blockchain no se libra de este último factor, y podría presentar ciertas vulnerabilidades en términos de ciberseguridad.

En primer lugar, la tecnología **Smart Contract supone una considerable vulnerabilidad** del Blockchain en términos de ciberseguridad. Se estima que, en una pieza de *software*

estándar, existen entre 15 y 50 *bugs* por cada 1000 líneas de código. En el caso de los *Smart Contracts*, una mínima cantidad de estos errores puede tener consecuencias catastróficas. El ejemplo más claro de este riesgo fue el **ataque a DAO**, una organización descentralizada implantada sobre el protocolo Ethereum. En este caso, el cibercriminal aprovecho la existencia de un *bug* dentro de un *Smart Contract* para sustraer la significante cantidad de 60 millones de Ether [29] o, lo que es lo mismo, 800 millones de euros de acuerdo al valor de la criptomoneda en aquella época. De hecho, en los últimos años numerosas organizaciones han aprovechado este nicho de mercado, y actualmente se pueden encontrar una gran variedad de entidades que ofrecen servicios de verificación y auditoría del código de estos contratos digitales.

Por otro lado, otro *threat* que corre cierto riesgo de ataque es la **manipulación del proceso de consenso** [22] para la verificación de transacciones, pieza clave del carácter descentralizado de toda aplicación basada en Blockchain. Esta manipulación puede darse, por ejemplo, en el caso de que el cibercriminal consiguiera acceder a la cadena, obtener credenciales y multiplicarse hasta controlar el 51% del poder de procesamiento de la red.

Además, hasta la fecha se han registrado numerosos ataques en redes Blockchain que entrarían dentro de tipologías tradicionales, con ligeras variaciones que permiten al cibercriminal lograr sus objetivos. Caben destacar los siguientes tipos de ciberataques convencionales:

- **PHISING**

  Este ataque, basado en ingeniería social, es uno de los más comunes dentro del mundo Blockchain. El crimen típico se basa en el registro de páginas web con dominios similares a sitios de acceso a bitcoin wallets u otro tipo de cadenas, pero que en realidad sirven al cibercriminal para la roba de datos, credenciales, criptomonedas, etc.

- **MALWARE**

  Numerosos atacantes han recurrido al *ransomware*, un *software* que permite bloquear el dispositivo de la víctima y exigirle un rescate (en criptomonedas, en este caso) a cambio de recuperar su información y acceso.

- **DDoS (Distributed Denial of Service)**

  Ethereum ha sido posiblemente la mayor víctima de este tipo de ataques. En concreto, el atacante introduce una operación en el *Smart Contract* con un bajo precio en Gas, pero que requiere altos niveles de capacidad de procesamiento. Al hacer esto, el rendimiento total de la red se ve perjudicado, y el proceso de minado altamente ralentizado.

No obstante, a pesar de todo lo recién mencionado, cabe también señalar que, aunque sí se ha conseguido hackear la capa de acceso a la tecnología del registro distribuido, la tecnología en el *core* de los distintos protocolos del paradigma Blockchain no se ha visto aún comprometida hasta la fecha.

Capítulo 6. APLICACIONES E IMPLEMENTACIONES DE LA BLOCKCHAIN

Tras haber descrito y profundizado en los aspectos fundamentales que constituyen el paradigma de la Blockchain, este capítulo presenta una primera toma de contacto con el proceso de implementación de aplicaciones descentralizadas basadas en esta tecnología.

Se procederá a detallar algo más dos de los protocolos Blockchain con mayor uso hoy en día, y se concluirá esta sección con una referencia tanto a áreas que están apostando fuertemente por la Blockchain como a otros campos en los que la aplicación de este paradigma resulta prometedora.

6.1 ANÁLISIS DE LAS TECNOLOGÍAS

En esta primera gran sección del capítulo se procede a analizar y describir más detalladamente dos tecnologías Blockchain que han despertado el interés de multitud de desarrolladores y/o organizaciones: la red Ethereum y el proyecto Hyperledger.

En primer lugar, se describirán los aspectos fundamentales de la red Ethereum, probablemente el segundo protocolo de Blockchain con más expansión en la actualidad, tras la archiconocida Bitcoin. De entre estos aspectos, se hará énfasis en los *Smart Contracts* y los mecanismos de consenso que emplea la red. Tras esto, se procederá a la descripción del proceso de implementación de aplicaciones (conocidas como Dapps) sobre esta Blockchain.

Respecto al proyecto Hyperledger, iniciado y mantenido por organizaciones de prestigio a nivel internacional, se describirán tantos sus aspectos funcionales fundamentales como las herramientas y plataformas que ofrece al desarrollador para simplificar significativamente la creación de soluciones basadas en Blockchain.

6.1.1 ETHEREUM

Grosso modo, Ethereum es una Blockchain **pública (sin permisos)**, programable y con una gran versatilidad en términos de **aplicabilidad**. A día de hoy, probablemente se podría señalas como la Blockchain pública con mayor potencial por sus altas prestaciones de flexibilidad, latencia y escalabilidad.

Como ya se mencionó anteriormente, aunque las latencias en esta Blockchain son significativamente menores que las de la Bitcoin, implantar un sistema completo sobre este protocolo o cadena provocaría igualmente problemas de velocidad. Por lo tanto, en la actualidad **gran parte de la lógica debe implementarse *off – chain*** con el fin de asegurar un correcto funcionamiento del sistema.

La otra gran diferencia entre Ethereum y Bitcoin es el **objetivo o propósito fundamental** de cada una de las cadenas. Mientras que Bitcoin fundamentalmente ofrece una red P2P que permite la realización y trazabilidad de transferencias monetarias *online*, Ethereum se centra en la **ejecución de programas/código dentro de cualquier aplicación descentralizada.**

Al igual que el resto de protocolos o Blockchains, Ethereum asegura la ya mencionada ***trust economy***. Cada **transacción que es incorporada a la cadena está firmada criptográficamente** por la cuenta del creador de la misma. Absolutamente todos los elementos de la Blockchain, incluidos los programas, están firmados.

Al igual que Bitcoin, Ethereum cuenta con su **criptomoneda nativa: el Ether**. Como ya se menciono en la investigación general sobre Blockchain, en la red Ethereum los mineros consumen sus recursos computacionales para ganar Ether; esta criptomoneda además, con la ayuda del **Gas (tarificación)**, es el medio que tienen los desarrolladores de aplicaciones para pagar las tasas correspondientes a la ejecución de transacciones y la creación de programas (*Smart Contracts*) dentro de la red.

La red Ethereum cuenta con un **inmenso número de cuentas asociadas**. Cada una des estas está posee un número de cuenta y un balance. Además, alguien (el protocolo de la cuenta) tendrá una **clave privada** que le permite realizar transacciones. También es común que una

cuenta cuente con código asociado que es ejecutado automáticamente ante la recepción de un pago. Estas cuentas cuentan con una serie de **rasgos característicos [24]:**

- En la red Ethereum, es común encontrar **cuentas que existen únicamente por el código que contienen** y que, por tanto, son comúnmente invocadas como si de funciones se trataran. Este código puede contener además variables, cuyos valores persisten a lo largo de las distintas llamadas.
- Cuando una cuenta existe por su balance y la posibilidad de ejecutar transacciones mediante el uso de la clave, es denominada **EOA** (*Externally Operated Account*). Si existe por si código y variables, es denominado contrato (o ***Contract***).
- Cada vez que un usuario desea ejecutar una transacción, inicia un ciclo desde una cuenta EOA. Los mensajes procedentes de esta cuenta ponen en funcionamiento (si procede) al *Contract* correspondiente que, a su vez, puede llamar a otras cuentas contrato. La red Ethereum gestiona todo este proceso hasta que la transacción inicial quede resuelta. Cabe recordar que todo esto ocurre simultáneamente en todos los nodos de la cadena, que también almacenan los datos de todas las cuentas Ethereum.

La siguiente ilustración describe el ciclo recién mencionado:

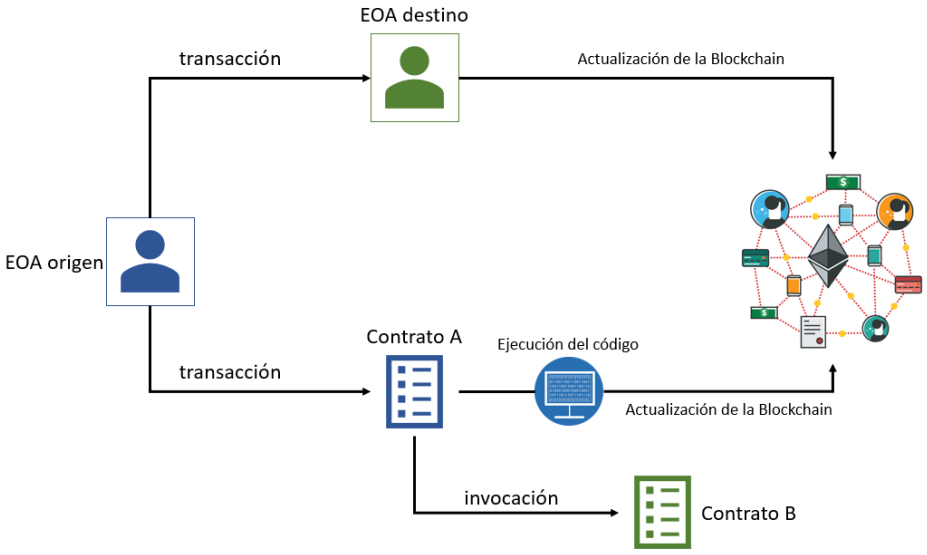


Figura 16. Ethereum. Tipos de Transacciones

Como se puede intuir a partir de lo recién descrito, los ***Smart Contracts*** constituyen un aspecto fundamental dentro de la red Ethereum. Un *Smart Contract* no es más que una pieza de *software*, un programa, que facilita y regula el intercambio de dinero, contenido o cualquier tipo de activo. Dentro de una Blockchain, estos contratos inteligentes se **ejecutan automáticamente cuando se dan ciertas condiciones** en la cadena. Además, al estar implantados dentro de la cadena, cualquier posibilidad de censura/fraude/intervención queda totalmente descartada y las condiciones descritas en este están blindadas.

Actualmente, cuando la tecnología Blockchain no ha alcanzado un gran nivel de aplicación en sistemas e infraestructuras IT, los *Smart Contracts* aportan una gran ventaja al **permitir iniciar procesos *off – chain***, cualquiera que pueda iniciar un programa estándar: envío de e-mails, impresión de libros, etc. Además, la ejecución de un *Smart Contract* puede ser **activada también por una condición externa**, de modo que éste estará ligado a eventos externos, ejecutándose al recibir una transacción firmada (encriptada) definiendo en lo que consistió dicho evento.

Mientras que todas las Blockchains son capaces de procesar código, la mayoría presenta significantes limitaciones en este aspecto. Sin embargo, Ethereum ofrece una gran versatilidad en este aspecto, gracias fundamentalmente a su lenguaje de descripción de contratos **Solidity,** que comparte bastantes similitudes con JavaScript. Tras compilarse, el código en Solidity es condensado en instrucciones u *opcodes*, interpretables y ejecutables por la *Virtual Machine* de Ethereum, conocida como **EVM**. Cabe mencionar que existe otro lenguaje de descripción de *Contracts* en Ethereum: ***Serpent*** (con similitudes con Python), aunque su grado de adopción es mucho menor que el de Solidity.

Con respecto al ámbito de los *Smart Contracts* dentro de Ethereum, cabe señalar dos conceptos adicionales [24]:

- Las **DAO (*Decentralized Autonomous Organization*)**, que consiste en la creación de una entidad íntegramente en la Blockchain, y se regula exclusivamente según lo codificado en ella. Por lo tanto, una DAO estaría constituida por una infinidad de *Smart Contracts* de alta complejidad, sería autosuficiente y gestionaría activos

digitales. No obstante, esta modalidad presenta numerosos desafíos en términos legales y/o éticos.

- El **Oráculo (*Oracle*)**, componente que aporta al *Smart Contract* la información que necesita exterior a la cadena. Suelen comunicar el mundo real con el digital. Ejemplos de oráculos son: sensores de un dispositivo IoT, *web services* que aportan información adaptada al formato que el contrato "entiende", etc.

El siguiente diagrama refleja el flujo de datos y acciones típico en toda aplicación descentralizada basada en Ethereum:

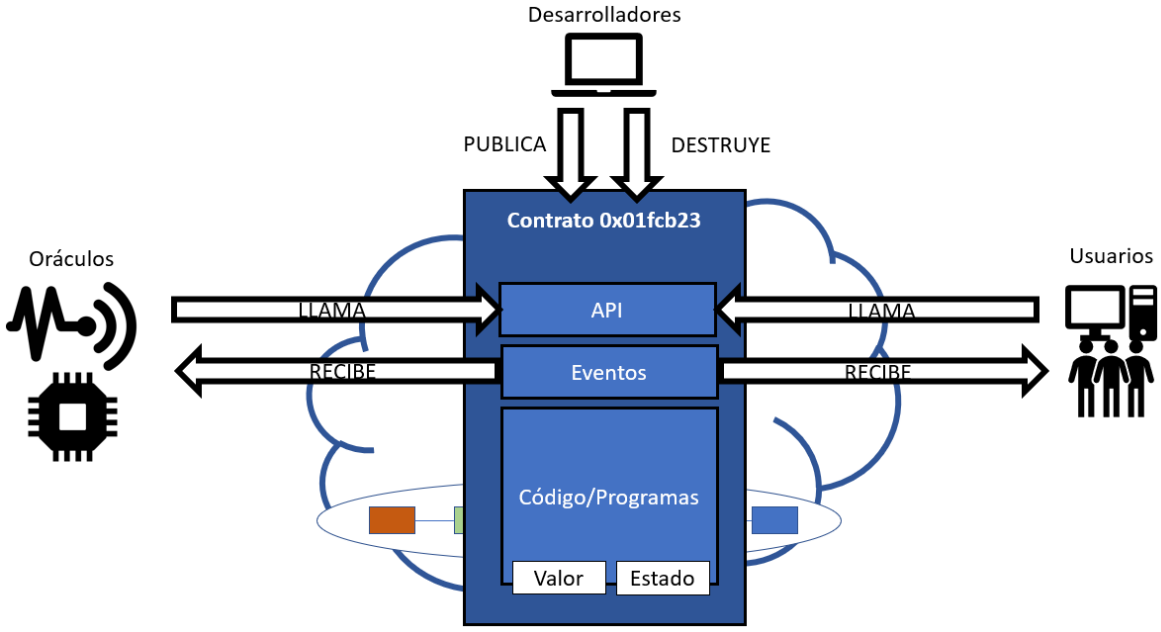


Figura 17. Intercambios de datos en una Dapp

Otro aspecto importante de la red Ethereum es el **mecanismo de consenso** que emplea, es decir, cómo consigue la red un acuerdo sobre el estado/aspecto global de la cadena entre todos los nodos. Actualmente, Ethereum hace uso de ***Proof of Work (PoW)*** que, como ya se ha mencionado anteriormente, consiste en la proposición de un bloque (con un nonce) más la noción que tiene el nodo sobre el estado global de la cadena. Normalmente se producen un gran número de propuestas conflictivas (con distinta perspectiva) en el mismo momento: cada uno de estos nodos incorpora el bloque creado en su Blockchain local, por lo que se

comparan estas copias de la Blockchain, de modo que los demás **nodos suelen apostar por el que haya propuesto la versión de la Blockchain "más larga".**

Como ya se mencionó con anterioridad, PoW conlleva un alto consumo de recursos computacionales y, por tanto, electricidad. Este hecho ha llevado a la creación de agrupaciones de mineros, o ***pool miners***, concentrados sobre todo en áreas con baja tarificación eléctrica. Esto supone un problema, ya que se corre el **riesgo de centralización del poder de minado sobre la red**. Ethereum trata de mitigar este riesgo mediante dos mecanismos [37]:

1) Reduciendo las pérdidas debidas a bloques huérfanos (que no pertenecen a la cadena principal), accidente que se da con más frecuencia en mineros independientes. Esto se consigue mediante una técnica conocida como **GHOST**.
2) El empleo de un **PoW resistente a ASICs**, *hardware* especializado únicamente en el proceso de minado. De este modo, los mineros *standard* se vuelven más competitivos y el proceso de minado más justo.

Cabe señalar también que la Ethereum Foundation está invirtiendo fuertes esfuerzos en la modificación del consenso empleado hacia el conocido como ***Proof – of – Stake*** **(PoS)**, en en castellano prueba de **participación**. Como ya se ha introducido anteriormente, este mecanismo consiste en la siguiente serie de acciones:

1) Los nodos de la red cuentan con una cantidad de criptomoneda, y están dispuestos a congelar parte de ésta (una participación).
2) Existe un algoritmo que decide **pseudo – aleatoriamente** que nodo (o validador) será el encargado de incorporar el siguiente bloque a la cadena. Si este bloque finalmente es incorporado con éxito en la Blockchain, el nodo creador cobrara las tasas de las transacciones descritas en el bloque.
3) Al crear el bloque el **resto de nodos procede a la validación** de la nueva actualización en función de un *set* de reglas (el bloque no incluye transacciones sin fondos, apunta correctamente al bloque anterior, etc.). Con el fin de fomentar la unanimidad y la eficacia de esta validación, aquellos nodos que hubieran validado

un bloque que finalmente resultó ser incorrecto correrán el riesgo de perder parte de su participación.

Previo a concluir con los aspectos relativos al mecanismo de consenso, cabe describir brevemente un fenómeno que ocurre en toda Blockchain y recibe el nombre de ***fork***, que consiste en la **bifurcación de la cadena**, es decir, **la creación de una nueva Blockchain a partir de un bloque de la Blockchain principal**. Además, esta bifurcación o *fork* puede ser de dos tipos:

- Suave (o *soft*), que consiste en un cambio del protocolo o las reglas de la cadena en el que se crean bloques que **siguen siendo válidos también en la "cadena *legacy*"** (el *software* anterior a la bifurcación). No obstante, este **nuevo *software* no admite el formato del antiguo**, por lo que si algún nodo no actualizado intenta crear un bloque, se producirá una división de la cadena en otras dos que tendrán en común todo lo construido hasta el momento del *fork*.
- Duro (o *hard*), en el que el **cambio del protocolo de la cadena es considerado como inválido en la Blockchain original**. Todos los nodos deberían actualizar el *software* para funcionar según las nuevas reglas. No obstante, en el caso de que existan nodos que intenten operar según el *software* anterior, se producirá una división de la cadena que puede dar lugar, en ocasiones, al nacimiento de una nueva criptomoneda.

Por último, dentro del ámbito de la **criptografía**, tanto en Bitcoin como en Blockchain las cuentas cuentan con una clave asociada y, además, toda acción ejecutada desde una cuenta debe estar firmada. Ambas Blockchains emplean **ECDSA** (*Elliptic Curve Digital Signature Algorithm*) para la generación del **par de sus claves públicas/privadas**. Obtener la clave pública a partir de la privada es trivial, mientras que realizar el proceso inverso es altamente complejo. De este modo, el usuario/cuenta de la cadena puede compartir su clave pública por distintos motivos sin tener que preocuparse por la seguridad de su clave privada.

En cuanto al algoritmo para calcular el ***hash*** de cada bloque, cabe señalar que Bitcoin emplea SHA – 256, Ethereum hace uso del algoritmo **Keccak – 256**.

6.1.1.1 Proceso de Implementación de Dapps sobre la red Ethereum

El desarrollo sobre Ethereum (y cualquier otra Blockchain) se conoce habitualmente como **código descentralizado**. *Grosso modo*, este término indica que los desarrollos están alojados de un modo idéntico en miles de equipos (o nodos) y se ejecuta en paralelo en todos ellos.

Las aplicaciones creadas siguiendo esta modalidad de desarrollo reciben el nombre de **Dapps**, y presentan las siguientes propiedades [24]:

- Una Dapp "pura" debería, en teoría contar con un ***backend* íntegramente descentralizado**, aportando a la implementación altos niveles de robustez y confianza.
- Los límites de rendimiento y capacidad de las Blockchains actuales provocan que en ocasiones parte del *backend* esté alojado fuera de la cadena, en entornos centralizados.
- El ***frontend*** de una Dapp (interfaz, gráficos, páginas webs, etc.) suele ser ***off-chain***, alojándose también en entornos centralizados como servidores o servicios *cloud*.

El desarrollo íntegro de una Dapp incluye varios procesos, algunos de complejidad elevada, que han de ser bien entendidos desde el primer momento. Los **pasos a seguir para la construcción de una Dapp**, a gran escala, serían los siguientes [38]:

1. Preparación del "proyecto"
2. Programación/codificación de los *Smart Contracts*
3. Desarrollo del *frontend* de la Dapp
4. Preparación de la Dapp para su visualización *online*: *hosting* (y dominio).

La **CREACIÓN/PREPARACIÓN DEL PROYECTO** correspondiente a una Dapp comparte muchas similitudes con la de una aplicación web *standard*.

En primer lugar cabe señalar que para el desarrollo de JavaScript en el lado del servidor se va a hacer uso del archiconocido **Node.js**, por lo que será necesario descargar su última versión. La principal utilidad del Node.js en una aplicación Blockchain es lo que se conoce

como **Event Loop**, que permite la realización de operaciones I/O de un modo asíncrono, permitiendo escalar grandes volúmenes de clientes (nodos) rápidamente. Tras esto, se debe proceder a instalar e iniciar **Truffle**¸ un *framework* que permite la creación, testeo y compilación de *Smart Contracts* en la red Ethereum.

Tras finalizar estos pasos, el proyecto contará con los ficheros esenciales que permitirán el desarrollo de Dapps. Para completar esta primera parte, basta con introducir los siguientes **comandos en el cmd**, dentro de la carpeta del proyecto:

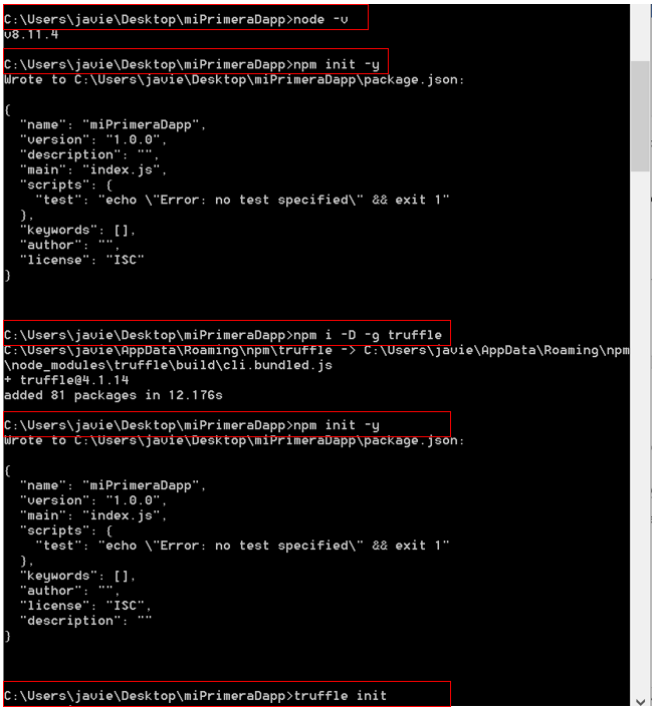


Figura 18. Comandos Iniciales. Node.js y Truffle

Como se puede observar en la captura anterior, es recomendable el uso de **npm** (*Node Package Manager)*, un gestor de paquetes que simplifica el trabajo con node.js al permitir el acceso a librerías en una sola línea de código y la fácil administración de los distintos módulos de la Dapp.

Tras este paso inicial, se puede optar por la instalación de más herramientas para el desarrollo *front-end* mediante el empleo de npm. Este es un paso libre, ya que se pueden emplear

herramientas que simplifiquen este desarrollo o no. No obstante, es altamente recomendable la instalación de **Web3**, una librería de JavaScript que facilita la interacción entre la interfaz de usuario y el *Smart Contract* que define la lógica de la Dapp.

La estructura de la "carpeta proyecto" antes de comenzara a implementar la lógica de la Dapp debería contar con los siguientes componentes:

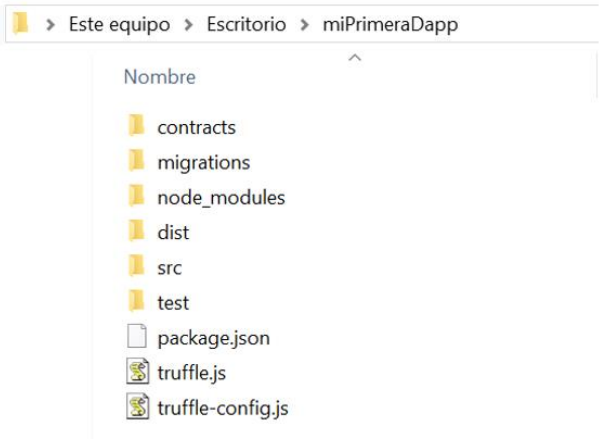


Figura 19. Preparación del Proyecto

Es en este momento se procede a **implementar, compilar y testear** el/los *Smart Contracts* que definirán la lógica/reglas de la Dapp empleando el lenguaje **Solidity**. Este proceso se describirá detalladamente más adelante.

Tras esto, se procede al **desarrollo del *front-end*** de la Dapp. Como ya se ha comentado, para esta parte del proceso se puede usar un amplio rango de *frameworks* de JavaScript según las preferencias y la experiencia del desarrollador. También se podría desarrollar con JavaScript plano. Ejemplos de *frameworks* de gran utilidad en este tipo de aplicaciones son React.js con webpack.

Una vez creados los ficheros .css, .js y .html del *frontend* de la aplicación, se debe proceder a la **conexión con el Smart Contract**. Para llevar a cabo este paso existen dos posibilidades:

A) **Usar directamente truffle** para implantar el contrato y obtener los datos. No obstante, está opción requiere la posibilidad de consumo de mucha memoria, al ser necesaria la descarga de toda la Blockchain Ethereum.

B) **Implementar el contrato dentro de un IDE** como **Remix**, por ejemplo, e interactuar con el *Smart Contract* mediante el uso de la dirección de su ABI (*Application Binary Interface*), componente que permite la lectura y el envío de datos entre el contrato y el exterior. A gran escala, la realización de esta alternativa consta de los siguientes pasos:

   a. Crear el contracto dentro del IDE, compilarlo y testearlo.

   b. Obtener la dirección y la ABI del *Smart Contract*.

   c. Conectar el *frontend* y el *Contract* mediante la inclusión de varias instrucciones de Web3 en el index.js del proyecto.

En este punto, se puede afirmar que se ha concluido con el desarrollo de la Dapp. En caso de querer trasladar la aplicación a la www, un servicio de ***hosting*** (IPFS, por ejemplo, gratuita y con *hosting* descentralizado) y **un DNS para crear un dominio** propio, en caso de querer darle un aspecto semántico a la URL.

6.1.1.2 Descripción de Smart Contracts en Solidity

Como ya se ha mencionado, una parte fundamental en el desarrollo de una Dapp es la descripción, compilación y testeo de *Smart Contracts*.

Esta sección pretende aportar una introducción a este proceso, señalando las herramientas básicas necesarias para su consecución, y analizando dos ejemplos de contratos desarrollados en Solidity: uno basado en documentación disponible *online* para tener una primera toma de contacto con las antes citadas herramientas, y el otro de implementación propia.

El proceso de ***testing*** en Solidity tiene una gran importancia. De hecho, junto con la creación de este lenguaje se implantó un completo ecosistema de programación con soporte para procesos de testeo hasta el mínimo nivel de detalle.

No obstante, este proceso es considerablemente complejo al requerir conocimientos sobre desarrollo de *testbenches*, entre otros. Por tanto, se ha optado por centrarse exclusivamente en los procesos de implementación y compilación del contrato.

Para este análisis, se ha hecho uso, fundamentalmente, de dos herramientas:

- **Remix**, un IDE de Solidity disponible *online*. Esta herramienta permite la compilación y ejecución de *Smart Contracts*, aportando mensajes de gran utilidad en el caso de aparición de errores o *warnings* en el código. Cuenta además con un *debugger* integrado, y permite la implantación directa de *Smart Contracts* sobre la siguiente herramienta.
- **Metamask**, una extensión compatible con varios navegadores que ofrece una *virtual wallet* para Ethereum y sus redes de prueba, y permite interactuar con Dapps y/o *Smart Contracts* sin necesidad de descargar la Blockchain completa o realizar instalaciones adicionales.

Con el fin de, en primer lugar, llevar a cabo un proceso de familiarización con estas dos herramientas, se ha codificado un *Smart Contract* basado significativamente en un contrato disponible en la web de Solidity [41]. El código de éste, junto con una breve descripción, están disponibles en el ANEXO B de esta memoria. Fundamentalmente, **el contrato gestiona un proceso de elecciones**, contando con las siguientes **prestaciones**:

- Los tres elementos principales son: los votantes, el administrador o moderador, y los distintos candidatos a ser elegidos.
- El candidato puede votar, o delegar su voto sobre otra persona.
- El moderador es el responsable de otorgar el derecho a voto a los votantes (direcciones de la Blockchain).

Para compilar y ejecutar el contrato en Remix, basta con seguir los siguientes pasos:

1. Descargar e instalar las dos herramientas (extensión de metamask para Chrome en este caso).

2. Tras abrir el IDE *online* Remix, aparecen una serie de opciones sobre compilación, *testing*, etc. En principio, para **compilar** el contrato programado se pueden mantener los ajustes por defectos. Bastaría con pulsar la opción "Start to compile" en la pestaña de compilación, como se muestra en la siguiente captura de pantalla:

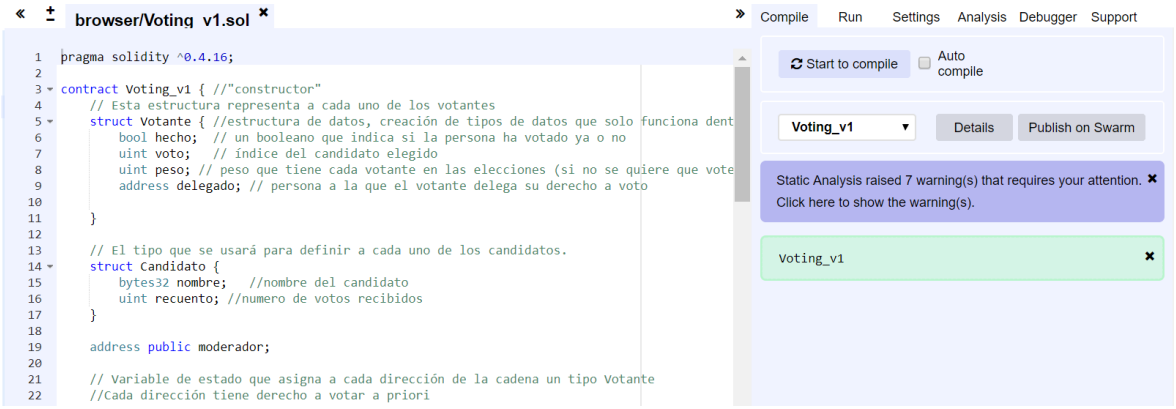


Figura 20. Compilación de un Smart Contract en Remix

Una vez compilado el código, este IDE proporciona además un **estimado de la tarificación** del contrato, es decir, el número aproximado de Gas que supondría la ejecución de la transacción (el contrato) en la red Ethereum. En este caso, la estimación mencionada aparece reflejada en la siguiente captura:

Figura 21. Estimación de la Tarificación en Remix

Como se puede observar, esta estimación refleja tanto costes de ejecución y depósito del contrato, como aquellos correspondientes a la ejecución de las distintas funciones y constructores del este.

3. Tras obtener una compilación sin errores, se puede proceder a **ejecutar** el contrato en la red Ethereum. No obstante, esto requiere el consumo de cierta cantidad de Ether. Como se va a trabajar en la red privada de prueba de Ethereum **Ropsten**, bastará con crear una cuenta Ethereum con la ayuda de metamask, y solicitar una pequeña cantidad de criptomoneda gratuita, para uso exclusivo en esta red de prueba. Tras seguir los sencillos pasos de instalación y registro en metamask, se habrá creado **una cartera cifrada con una contraseña a elegir por el usuario.** El aspecto de la pantalla principal de esta cartera tras haber solucionado el Ether gratuito será el siguiente:

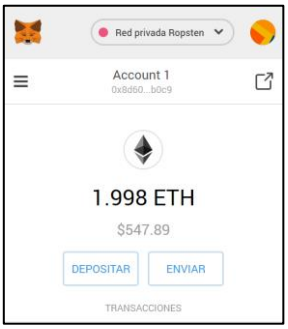


Figura 22. Metamask. Cartera Ethereum

4. Para implantar el *Smart Contract* será necesario escoger la opción "*deploy"* de la pestaña de ejecución del IDE. Antes de esto, será necesario especificar una serie de factores:
    a. La red sobre la que se quiere ejecutar el contrato: la red Ropsten de prueba en este caso.
    b. La cuenta desde la que se extraerán la tarifa correspondiente a la implantación del contrato, disponible en la ventana principal de Metamask.
    c. El Gas que se está dispuesto a pagar para la transacción, con el fin de incentivar y agilizar el proceso.

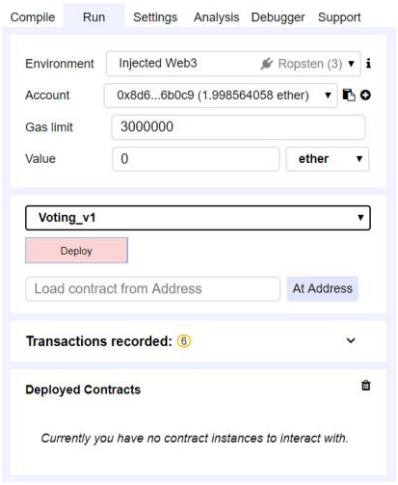


Figura 23. Ejecución del contrato en Remix

5. Tras esto, Metamask preguntará al usuario si desea confirmar la transacción, informándole del coste que esto conlleva (en $ y Ether). En caso de que el usuario dé el OK, y cuente con fondos suficientes, la transacción será aprobada y el contrato incluido en la red de pruebas de Ethereum.

Una vez finalizado todo el proceso, resulta interesante echar un vistazo a los detalles de la transacción correspondiente a la implantación del contrato en la cadena. Esto es posible gracias a que la herramienta Metamask tiene conexión con **etherscan**, que permite realizar labores de exploración dentro de la red Ethereum:

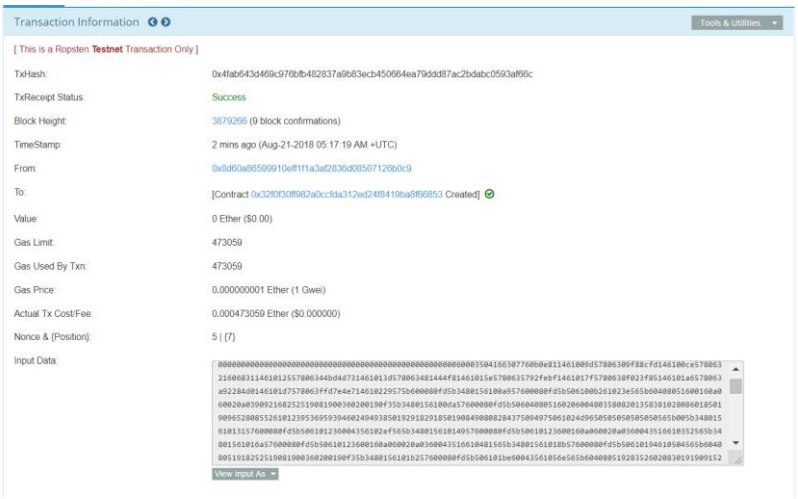


Figura 24. Detalles de una Transacción. Etherscan

Una vez realizada esta primera toma de contacto con el proceso de compilación e implantación de Smart Contracts en la Blockchain, se procede a describir el trabajo realizado relativo a otra importante aplicabilidad del desarrollo de contratos en Solidity: **la creación de criptomonedas/tokens personalizados**.

En concreto, se ha implementado el *Smart Contract* relativo a la **descripción de la criptomoneda Icaicoin**, cuya abreviatura o símbolo será **ICC**. Para la programación de la misma, se han consultado diversas webs y foros de la comunidad Ethereum (Ethereum Wiki, Quora) [43], aunque el "producto final" es resultado de comprobaciones y experimentaciones propias hasta lograr el funcionamiento deseado: contar con un *token* virtual que se pueda implantar en la red Ethereum, y permita transferencias entre cuentas/direcciones.

Respecto a la experiencia de programar en Solidity, cabe destacar su considerable grado de similitud con otros lenguajes de programación, sobre todo en términos de concepto. Los elementos más novedosos, personalmente hablando, de este lenguaje han sido:

- Los **eventos**. Cada vez que se llama a un evento, éste crea la información a almacenar en el log de transacciones. Además, este elemento permite la comunicación con aplicaciones clientes u otras Dapps. Estos logs, por lo general, persisten sin modificación alguna durante todo el período de vida de la cadena en cuestión. En el caso de la criptomoneda creada, se han descrito dos eventos para crear logs cada vez que se realice una transferencia del *token* o una concesión de una cuenta a otra.
- Los **mapeos** hacen referencia a **una estructura tipo clave – valor**. Se declaran como cualquier otro tipo de variable: ***mapping (_KeyType => _ValueType) mapName***, de modo que el mapName será del tipo *_ValueType*.

El resto de componentes son altamente similares a los empleados para creación de clases, funciones/métodos y atributos/variables en lenguajes como Java. Las reglas de herencia son también similares.

A continuación se presenta el código (comentado) implementado para la descripción de la Icaicoin (ICC):

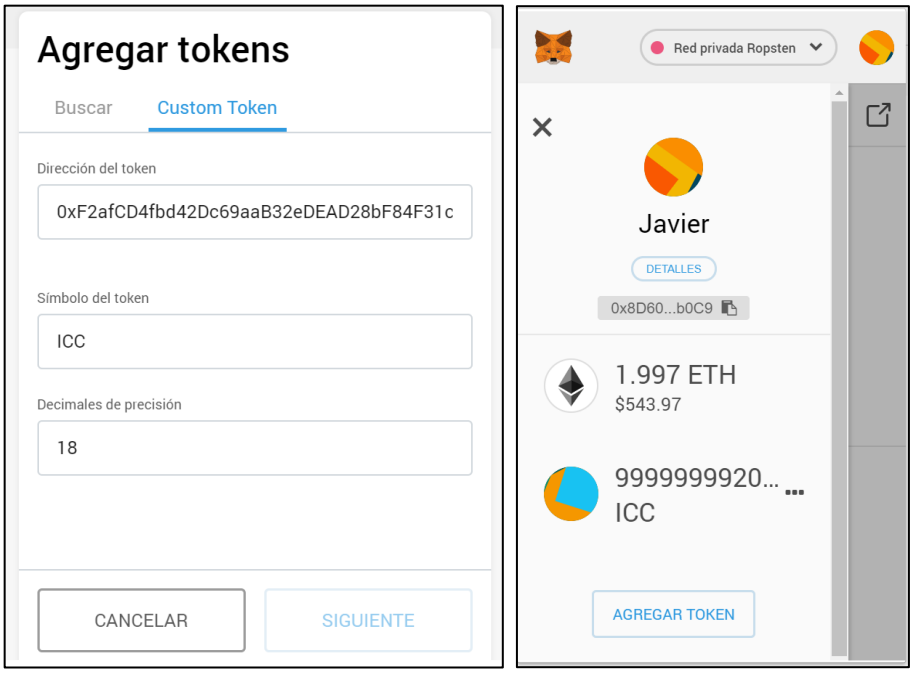
```
pragma solidity ^0.4.4;

//Primero creamos una especie de "clase genérica"
//indicando qué funcionalidad debe incluir todo token

contract Token {

    // función que devuelva la cantidad total de tokens existentes
    function totalSupply() constant returns (uint256 supply) {}

    //función para obtener el balance de un usuario (_owner), que devuelve el valor de balance
    function balanceOf(address _owner) constant returns (uint256 balance) {}


    //función para transferir una cantidad de token _value a la dirección _to. Nos devuelve true si se ha realizado
    //con éxito
    function transfer(address _to, uint256 _value) returns (bool success) {}

    //una función similar a la anterior, pero indicando también la dirección del emisor de la transferencia
    function transferFrom(address _from, address _to, uint256 _value) returns (bool success) {}

    //una función por la cual una dirección (msg.sender) autoriza a otra _spender a gastar una cantidad _value
    function approve(address _spender, uint256 _value) returns (bool success) {}

    //está última función es muy común en el mundo token. Sirve para saber cuanto saldo de una "cesión" le queda
    //a la cuenta receptora de esta cesión. Más o menos
    function allowance(address _owner, address _spender) constant returns (uint256 remaining) {}

    //se crean una serie de eventos para la creación de los logs de las distintas transacciones
    //esto permite, entre otras cosas, comunicación del contrato con otros o con cualquier cosa conectada
    //a este por una API
    event Transfer(address indexed _from, address indexed _to, uint256 _value);
    event Approval(address indexed _owner, address indexed _spender, uint256 _value);

}
//creamos otro contrato (o clase) que herede de la anterior, en el que redefinimos las funciones declaradas en Token
contract StandardToken is Token {

    function transfer(address _to, uint256 _value) returns (bool success) {
        //se realiza una comprobación de si el emisor tiene fondos suficientes y se quiere enviar
        //una cantidad de tokens superior a cero. Si es así, se realiza la transferencia
        if (balances[msg.sender] >= _value && _value > 0) {
            balances[msg.sender] -= _value;
            balances[_to] += _value;
            Transfer(msg.sender, _to, _value);
            return true;
        } else {
            return false;
        }
    }
    //una cuenta/dirección desea transferir dinero pero desde otra cuenta distinta.
    function transferFrom(address _from, address _to, uint256 _value) returns (bool success) {
        //esto seré posible si la cuenta origen tiene fondos y los cedidos al que llama a la función son suficientes
        if (balances[_from] >= _value && allowed[_from][msg.sender] >= _value && _value > 0) {
            balances[_to] += _value;
            balances[_from] -= _value;
            allowed[_from][msg.sender] -= _value; //se resta de lo que le queda concedido al emisor del mensaje
            Transfer(_from, _to, _value);
            return true;
        } else {
            return false;
        }
    }
```

Figura 25. Código Icaicoin. Parte 1/2

```
    function balanceOf(address _owner) constant returns (uint256 balance) {
        return balances[_owner];
    }

    // Permite al spender extraer de la cuenta del que llama a la función, x veces hasta alcanzar la cantidad máxima _value
    function approve(address _spender, uint256 _value) returns (bool success) {
        allowed[msg.sender][_spender] = _value;
        Approval(msg.sender, _spender, _value);
        return true;
    }

    //devuelve la cantidad de dinero que sigue pudiendo extraer spender de owner según la Approval que hayan hecho
    function allowance(address _owner, address _spender) constant returns (uint256 remaining) {
      return allowed[_owner][_spender];
    }

    mapping (address => uint256) balances; //el uso de la función mapping de solidity permite que balances[x] haga referencia
    //al balance de la dirección x. Otra opción sería usar x.balance (así no me funcionaba)

    //el propietario de una cuenta aprueba que otra cuenta le quite una cantidad determinada
    mapping (address => mapping (address => uint256)) allowed;

    uint256 public totalSupply;
}


//creamos un contrato/clase de nuestra criptomoneda, que heredará toda la funcionalidad de StandardToken
contract Icaicoin is StandardToken {

    //función que lanza una excepción en caso de que se mande Ether a la dirección del contrato de Icaicoin
    function () {

        throw;
    }

    //variables públicas del token

    string public name;                   //nombre de la criptomoneda
    uint8 public decimals;                //a cuantos decimales aproximar
    string public symbol;                 //el identificador de la criptomoneda. Como BTC, ETH,etc
    string public version = '1.0';       //estándar para control de versiones


    //iniciador de la criptomoneda
    function Icaicoin(
        ) {
        //El balance del total existente que se asigna en un principio al creador del token
        balances[msg.sender] = 10000000000000000000000000000000;
        //el total de Icaicoins que habrá en el mercado. inicialmente, se las asignaos todas al creador
        totalSupply = 10000000000000000000000000000000 ;
        name = "Icaicoin";
        decimals = 18;
        symbol = "ICC";
    }

}
```

Figura 26. Código Icaicoin. Parte 2/2

Tras codificar el fichero recién mostrado, se volvieron a emplear Remix IDE y Metamask para compilar y ejecutar el mismo. Tras desplegar el contrato en la red Ropsten de prueba de Ethereum, Metamask proporciona información sobre el mismo. De entre ésta, se hizo uso de la dirección del contrato para el testeo/comprobación del funcionamiento del *token* creado gracias a la funcionalidad de esta misma herramienta:



Como se puede observar en las capturas anteriores, tras añadir el token, el balance del usuario (el creador, en este caso) en **ICC aparece disponible en la *virtual wallet*** de Metamask. Nótese que la cantidad reflejada es menor al *totalSupply* porque en el momento de realización de captura se habían realizado ya varias pruebas sobre el nuevo *token*.

Para la realización de estas **pruebas**, se hizo uso de otro equipo desde el que se creo una cuenta en Metamask y se agregó la criptomoneda ICC. Después de esto, se probó a transferir dinero de una cuenta a otra para verificar el correcto funcionamiento de la moneda. Las capturas correspondientes a estas transferencias, así como detalles más específicos de la criptomoneda proporcionados por etherscan, están disponibles en el ANEXO C de la memoria.

6.1.2 HYPERLEDGER

Como ya se introdujo con anterioridad, Hyperledger es un proyecto iniciado por la **Fundación Linux** a finales de 2015, y del que ya forman parte más de 200 organizaciones mundiales (Accenture, BBVA, IBM, etc.). Surgió como resultado del interés que la tecnología Blockchain levanto en diversas entidades, que generaron una alianza y trabajan en aumentar la popularidad y **estandarización** de este paradigma.

Uno de los fundamentos de este proyecto es la utilización de tecnología íntegramente ***open-source***. El *software* propietario y los sistemas *legacy* cuentan con licencias restrictivas y costosas, y sólo puede ser modificado o actualizado por el *vendor* en cuestión. Por otro lado, el código abierto puede ser descargado, visualizado y/o modificado libremente. Ésta última alternativa presenta importantes **beneficios**, entre los que destacan:

- Ausencia de exclusividad impuesta por el *vendor*, lo que provoca un aumento de la competitividad y las exigencias entre los mismos.
- Soluciones con altos niveles de calidad y customización.
- Ahorro significativo en términos de TCO (*Total Cost of Ownership*).
- Promoción de la interoperabilidad, es decir, la posibilidad de interacción e integración con programas de otras organizaciones o incluso Blockchains.

A pesar de que Hyperledger cuenta con diferentes plataformas y herramientas para la implementación de Blockchains en función de las necesidades de cada organización, y estos medios presentan también diferencias a nivel técnico, existen una serie de rasgos comunes a todos ellos. Los **principios de diseño** presentes en toda solución Blockchain implementada dentro del ecosistema Hyperledger son [36]:

A) <u>Modularidad</u>

Hyperledger desarrolla una gran cantidad de *frameworks* constituidos por bloques comunes que pueden ser reutilizados. La creación de estos bloques/módulos permite la implementación de aplicaciones que se ajusten de un modo idóneo a unos requisitos concretos, a partir de combinaciones de estos. El **desarrollo de módulos** tiene una gran

popularidad en grandes organizaciones ya que, una vez ajustados del modo deseado, pueden **ser reutilizados** en multitud de aplicativos y arquitecturas.

B) Seguridad

En aplicaciones industriales, la tecnología del registro distribuido es empleada para el **procesamiento de transacciones y datos con alto valor y sensibilidad**. Por tanto, la creación de tecnología altamente segura es clave para favorecer la expansión de la Blockchain. Los diferentes algoritmos, mecanismos criptográficos y protocolos de Hyperledger son revisados y auditados por expertos en seguridad y la comunidad *open-source*.

C) Interoperabilidad

Aunque puede ser que la tecnología Blockchain no se encuentre aún en ese grado de madurez, en un futuro próximo **distintas cadenas deberán ser capaces de comunicarse e intercambiar información**, generando redes potentes con mayor complejidad. Es por lo que en Hyperledger se hace un esfuerzo por implementar *Smart Contracts* y aplicaciones portables para distintas cadenas de bloques.

D) Ausencia de Criptomoneda propia/nativa

El proyecto Hyperledger es independiente de cualquier criptomoneda o *token*, y no existen planes de implantar una moneda nativa. Sin embargo, sí es posible la creación de *tokens* en aplicaciones para la gestión de un activo digital.

E) Tecnología API

Los distintos proyectos de Hyperledger proporcionan una serie de APIs que permiten la interoperabilidad entre sistemas, y sirven de interfaz entre clientes externos y la infraestructura *core* del registro Hyperledger.

6.1.2.1 Ecosistema Hyperledger. Frameworks y Herramientas

Los fundadores del proyecto Hyperledger eran conscientes de la complejidad existente tanto en el ámbito de la Blockchain como en las necesidades de un mundo empresarial cada vez más exigente. De ahí que la estrategia Hyperledger haya creado un amplio rango de Blockchains empresariales (*frameworks*), herramientas y librerías siguiendo sus principios de diseño. El ecosistema resultante de todo esto aparece sintetizado en el siguiente esquema:

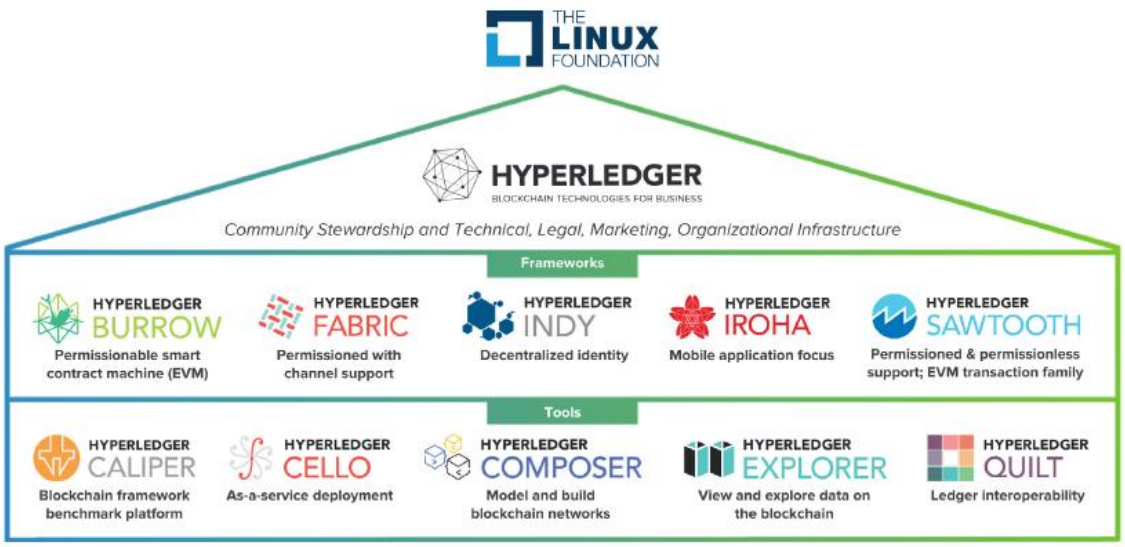


Figura 27. Ecosistema Hyperledger [36]

A pesar de que más adelante se profundizará algo más en alguno de estos recursos, cabe hacer una breve referencia a todos los ***frameworks* de registro distribuido** que proporciona Hyperledger:

- **Hyperledger BURROW**, un cliente Blockchain privado (con permisos) que cuenta con un intérprete de *Smart Contracts* que **soporta *Solidity***, al estar éste implementado parcialmente según las especificaciones de la EVM de Ethereum.
- **Hyperledger FABRIC**, una plataforma que permite la implantación de registros distribuidos según una arquitectura modular, aportando **altos niveles de flexibilidad**. Sus principales contribuciones vienen de la mano de IBM, y es una de las Blockchains privadas más empleadas al permitir el despliegue de *Smart Contracts*

(*chaincodes*) en cualquier lenguaje o decidir qué tipo de consenso emplear entre una gran variedad de mecanismos disponibles.

- **Hyperledger INDY**, un registro compartido que ofrece herramientas, librerías y componentes reutilizables centrados en la implantación de la **identidad digital descentralizada.** Emplea el "*Zero Knowledge Protocol*", que se centra en aportar la mínima cantidad de información posible en procesos que requieran esta identidad digital.
- **Hyperledger IROHA**, con un enfoque también orientado parcialmente a la identidad digital, busca la simplicidad y modularidad para facilitar su incorporación en infraestructuras empresariales. Soporta *Smart Contracts* desarrollados en Java.
- **Hyperledger SAWTOOTH**, cuyas aportaciones principales provienen de **Intel**. Permite la construcción, implantación y ejecución de registros compartidos. Soporta *Solidity* e incorpora un nuevo mecanismo de consenso, **PoET**, mucho más eficiente que PoW en términos de consumo de recursos computacionales.

A pesar de que, a día de hoy, el proyecto más conocido de Hyperledger es el *framework* Fabric, en este caso se ha decido analizar algo más en detalle el proyecto SAWTOOTH, por el hecho de que no se han encontrado tantas referencias a éste en la Web y a simple vista resultó ser igual de interesante y versátil que el primero. La siguiente sección de la memoria describirá la toma de contacto llevada a cabo con esta Blockchain.

Por otro lado, cabe señalar una serie de ideas principales en relación a las herramientas que apoyan este proceso de creación de aplicaciones Blockchain:

- **Hyperledger CALIPER**, una herramienta que permite **comprobar el rendimiento** de cualquier cadena mediante el uso de un *dataset* que contiene una gran variedad de casos de uso ya implementados. Entre los indicadores de rendimiento están la utilización, la latencia en transacciones, el número de transacciones por segundo, etc.
- **Hyperledger CELLO,** que pretende transformar la Blockchain en un ***pay-as-you-go***, aportando mecanismos para la provisión de recursos en función de la demanda o las necesidades de la cadena en cada momento.

- **Hyperledger COMPOSER**, un *set* de herramientas que simplifica el desarrollo de aplicaciones descentralizadas e implementaciones Blockchain, **abstrayendo al usuario de la parte de codificación**. De este modo, el usuario únicamente se centra en diseñar la lógica de activos digitales, nodos y transacciones desde un punto de vista funcional.
- **Hyperledger Explorer**, una especie de **visor** que contiene **paneles** con información sobre los bloques, los *logs* de cada nodo, estadísticas, transacciones, ejecuciones de *Smart Contracts*, etc.
- **Hyperledger QUILT**, cuyos esfuerzos se centran en facilitar la **interoperabilidad entre Blockchains** mediante **ILP** (*Interledger Protocol*), protocolo creado inicialmente por Ripple y destinado a la transferencia de valor entre redes distribuidas y no – distribuidas.

6.1.2.2 Implementaciones en Hyperledger

Con el fin de comprender todo lo qué hay dentro de estos *frameworks*/proyectos de Hyperledger, se ha decidido analizar la arquitectura y la implementación de aplicaciones de uno de ellos: **Hyperledger Sawtooth**.

Como la mayoría de *frameworks* del proyecto Hyperledger, Sawtooth es una plataforma que permite el desarrollo de aplicaciones en Blockchain, con un enfoque mayoritario a soluciones empresariales y **procesos B2B** (*Business to Business)*. Está diseñado de tal forma que simplifica significativamente el proceso de desarrollo, gracias a elementos como los módulos, la ejecución en paralelo, el soporte multi – lingüe y mecanismos de gobierno.

Este *framework* cuenta con una **arquitectura basada en capas**, estableciéndose una clara separación entre el **núcleo** y la **capa de aplicaciones.** Cabe señalar que el núcleo es la parte encargada de la creación, mantenimiento y actualización de la Blockchain, así como de la comunicación con el resto de nodos existentes en la red.

Esta separación núcleo – aplicaciones facilita el proceso de desarrollo de soluciones sin que sea necesario prestar atención a factores como la seguridad de la red o los protocolos de comunicación entre nodos.

Para terminar de entender la arquitectura de este *framework*, cabe describir otros tres componentes del mismo:

- Un **SDK** (*Software Development Kit)*, para una gran variedad de lenguajes de programación: Go, JavaScript, C++, Python, etc.
- Una capa denominada **Validador**, que gestiona las comunicaciones entre el núcleo Sawtooth y la capa de aplicaciones.
- Una **REST API** propia, que gestiona las comunicaciones con el "mundo real". Esto facilita significativamente la comunicación de cada nodo Blockchain con soluciones del lado cliente (páginas web, aplicaciones móviles) y *hardware* especializado (sensores, dispositivos IoT) mediante el uso del protocolo HTTP.

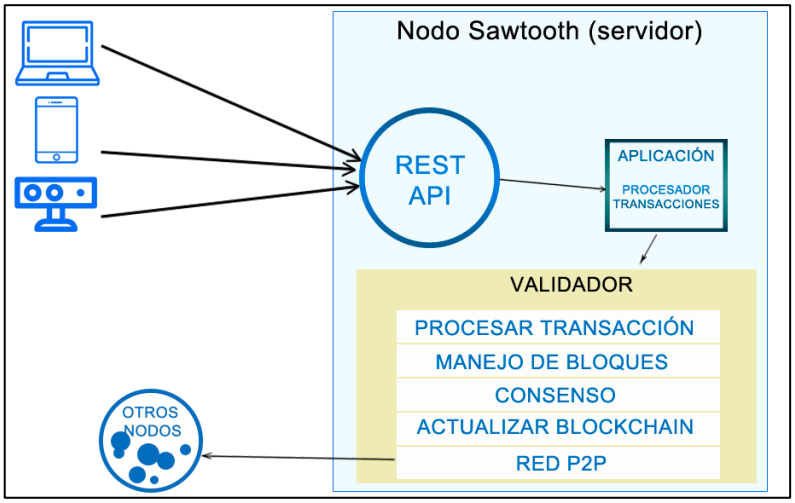


Figura 28. Flujo de datos con Hyperledger Sawtooth

Al igual que el *framework* Fabric, Sawtooth es una Blockchain privada (con permisos) y proporciona a los desarrollos la posibilidad de definir el **algoritmo de consenso de entre una gran variedad de opciones disponibles**: PoW, PoS, BFT. Sawtooth ofrece la

posibilidad de usar un algoritmo no mencionado hasta el momento: el **PoET (*Proof of Elapsed Time*)**, en el que el consenso se alcanza siguiendo los siguientes pasos [42]:

- Cada uno de los participantes/nodos de la cadena espera una cantidad de tiempo determinada aleatoriamente.
- El nodo que termine antes de esperar pasa a ser el elegido para la creación del siguiente bloque.

La versatilidad y altas prestaciones de esta plataforma convierten a Sawtooth en una alternativa viable para multitud de **casos de uso**. Por ejemplo, a continuación se presenta el planteamiento en alto nivel de una **aplicación que guarde y gestione los derechos de autor**.

Para ello, en primer lugar, se trata de simular un escenario típico de la aplicación:

1. María compone y graba una canción.
2. La autora graba la canción en un formato digital.
3. El organismo regulatorio correspondiente asegura que la canción pertenece a María mediante la emisión de un certificado digital de propiedad.
4. La aplicación, diseñada sobre Sawtooth, guarda en la Blockchain una copia encriptada de la canción de Mike junto con el certificado, un *timestamp* y los datos del autor.

Una vez planteado el escenario típico, se procede a describir la arquitectura básica de la solución. Los componentes necesarios para la implantación de este proyecto serían, fundamentalmente: un equipo trabajando en Ubuntu, la plataforma Hyperledger Sawtooth, *un backend* (aplicación lado servidor) y un *frontend* (aplicación lado cliente):

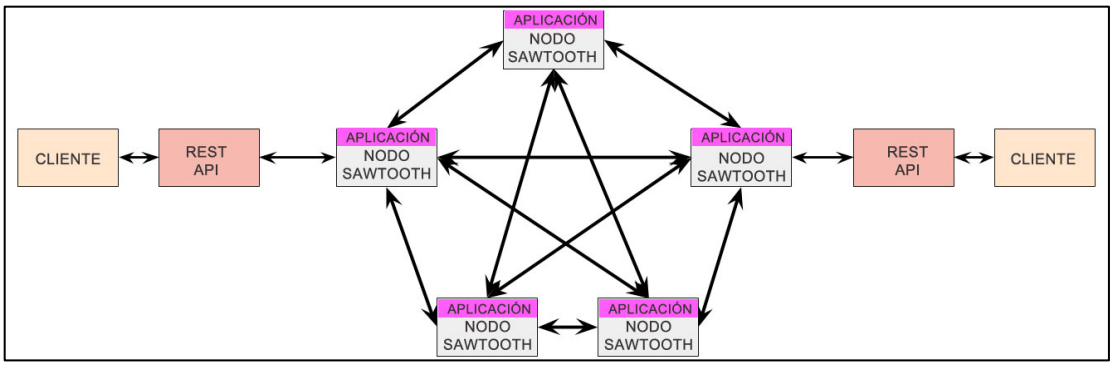


Figura 29. Arquitectura básica. Solución en Sawtooth

Por un lado, la **aplicación en el lado del servidor** deberá contar con:

- Un programa (o contrato) que permita el registro de usuarios. Cada vez que un usuario se registre, se generarán las claves pública y privada, necesarias para la identificación del usuario en la Blockchain.
- Un *script* que permita la subida y encriptado de los trabajos (y certificados) por parte de los usuarios.
- Una rutina que envíe todos los datos al validador de Sawtooth.
- Un rastreador público de Blockchain.

Por otro lado, la aplicación del lado cliente contaría con una estructura muy básica, estando compuesta por los siguientes elementos:

- Una página principal, con formularios de registro e inicio de sesión.
- Un área personal para cada artista, con información confidencial (datos personales, trabajos, claves) y una sección para la subida de contenido.
- Un rastreador público de Blockchain.

6.1.3 ETHEREUM VS HYPERLEDGER. COMPARACIÓN DE LAS TECNOLOGÍAS

Para concluir con el análisis realizado sobre las tecnologías Blockchain Ethereum e Hyperledger, se aporta una comparación a gran escala de las mismas, reflejada en la siguiente tabla:

Aspecto	Ethereum	Hyperledger (Sawtooth)
Uso	Genérico	Empresarial
Gobierno	Desarrolladores de la red	Fundación Linux
Tipo de Blockchain	Mayoritariamente pública (sin permisos) Quorum: Blockchain privada	Privada (con permisos)
Consenso	PoW En transición hacia PoS	A elección del desarrollador (PoW, PoS, PoET, etc.)
Smart Contracts	En Solidity	En diversos lenguajes SDK soporta Solidity, Java, Go, Python, etc.
Moneda	Ether Posibilidad de creación de otras monedas y *tokens* vía *Smart Contracts*	No tiene moneda nativa Posibilidad de creación de moneda y tokens vía *chaincode*

Tabla 4. Ethereum vs Hyperledger

6.2 CAMPOS DE APLICACIÓN DE LA BLOCKCHAIN

Esta segunda sección del Trabajo de Fin de Máster concluye con una presentación de algunas de las principales áreas/industrias en las que la tecnología Blockchain está dejando huella hasta el momento.

6.2.1 SECTOR FINANCIERO

Esta área es, sin duda, la que ha mostrado más interés (y a la vez rechazo) por el paradigma Blockchain en los últimos años. Numerosos expertos señalan que lo que esta tecnología traerá y está trayendo a la banca y los servicios financieros es equiparable a lo que Internet trajo a los medios de comunicación.

Tradicionalmente, los distintos mecanismos que hacían funcionar al sistema financiero, poderosos intermediarios que consolidan el capital y ejercen influencias que pueden llegar a imponer economías de monopolio, han provocado una ralentización significativa de todo el proceso, una adición de costes y una generación ingente de beneficios para ellos mismos.

Esta situación, junto con el hecho de que, tecnológicamente, numerosas entidades cuentan con infraestructuras obsoletas, han motivado a *managers* y emprendedores en la búsqueda de nuevas soluciones Blockchain que creen valor en este sector. Existen fundamentalmente seis razones que hacen de esta transformación un paso necesario [12]:

1) **Certificación/Atestación**. La Blockchain permite la negociación entre partidos sin que sea necesaria la existencia de confianza entre los mismos. Además, en caso de que esta confianza fuese necesaria, esta tecnología permite la verificación de un individual/entidad a través de una combinación de su historial dentro de la cadena, datos de solvencia y otros indicadores socioeconómicos.
2) **Costes**. El ahorro de billones de € en intermediarios (reguladores, auditores, consultores, etc.) permitiría aumentar las inversiones destinadas a la mejora del modelo de negocio y la accesibilidad universal del sistema bancario/financiero.

3) **Velocidad**. A pesar de que numerosos expertos han criticado las carencias de la red Bitcoin para el soporte de los millones de transacciones financieras que se dan a diario, existen otras Blockchains (Bitcoin Lightning Network, Ripple) capaces de reducir los tiempos de resolución de transacciones a fracciones de segundo.
4) **Gestión de Riesgos**. Fundamentalmente, la Blockchain promete optimizar la gestión de riesgos mediante la mitigación de la mayoría de los mismos: riesgos de resolución de transacciones, riesgos asociados a las contrapartes, etc.
5) **Innovación**. La red Bitcoin fue inicialmente diseñada para el movimiento de esta criptomoneda exclusivamente, no el de cualquier otro tipo de activo (financiero o no). No obstante, esta y otras Blockchains admiten un alto nivel de experimentación, lo que permite al paradigma ser capaz de resolver un gran rango de necesidades dentro de cualquier sector.
6) **Tecnología *Open-Source***. Por lo general, las infraestructuras IT de una entidad u organismo financiero han estado basadas en sistemas *legacy*, con altos costes de mantenimiento y licencias y numerosos problemas de interoperabilidad. Por el contrario, la tecnología Blockchain es 100% *open-source*, lo que permite una absoluta constancia en innovación y mejora.

Por estos y otros motivos, durante los últimos años han surgido numerosas aplicaciones de la Blockchain en este sector. Muchas de ellas están enfocadas a la **realización de transferencias internacionales,** una operación lenta y costosa si se realiza a través del sistema bancario tradicional. Esto es debido a que, en este tipo de operaciones, el paso del dinero entre bancos y/o divisas se realiza de un modo bastante manual en el que son necesarios diversos intermediarios. El empleo de una red Blockchain para estas transferencias permitiría reducir los tiempos de espera de varios días a menos de diez minutos, y los costes (comisiones) de mínimos de 40/50 € a unos céntimos. **Ripple** es probablemente la Blockchain más representativa dentro de esta actividad, aunque ciertas entidades bancarias (Banco Santander, entre otros) llevan años trabajando en sus propias implementaciones Blockchain con este mismo fin.

Además, como ya se ha mencionado, la tecnología subyacente de este paradigma refuerza la trazabilidad y persistencia de las transacciones, aspectos cruciales para el proceso de **KYC (“*Know Your Customer*”)**, por el cual las compañías pueden identificar a sus clientes, evitando la participación accidental en blanqueos de dinero y/o financiación de actividades ilegítimas (guerras, terrorismo, etc.).

6.2.2 CADENAS DE PRODUCCIÓN. SUPPLY CHAIN MANAGEMENT

Actualmente, los últimos avances tecnológicos y las altas exigencias del consumidor han llevado a que la mayoría de las compañías se enfrenten a la creación de productos altamente complejos y sofisticados, constituidos por una gran cantidad de componentes. Todo esto desemboca en la implantación de cadenas de producción (*supply chains*) enormes tanto en número de actores implicados como en cantidad de procesos a ejecutar diariamente.

Con el fin de garantizar un correcto funcionamiento de toda cadena de producción, existen factores fundamentales a garantizar, tales como la comunicación constante entre actores/participantes, la máxima transparencia posible y la optimización de los tiempos de duración de los distintos procesos implicados. La tecnología Blockchain y, en especial, su tecnología de registro distribuido y descentralizado puede favorecer el sector del SCM mediante el aprovechamiento los siguientes aspectos [30]:

- Empleando time-stamps, trazas y automatización sobre las distintas transacciones, permitiendo una **auditoría total de la evolución del activo en tiempo real**.
- **Minimizando la participación de intermediarios**, tales como bancos o aseguradoras. Esto resulta de gran interés, por ejemplo, para cadenas que cuentan con participantes dispersos por todo el mundo, en países con monedas y regulaciones distintas.
- Configurando e incluyendo ***Smart Contracts* que permitan la automatización de procesos repetitivos y estandarizados** como los envíos y la facturación.
- Facilitando la escalabilidad, tanto en clientes como en proveedores, mediante la asignación de IDs digitales (incorporación de nuevos bloques a la cadena).

- La transparencia y persistencia de todas las transacciones dentro de la cadena permiten también que el **proceso de contratación de proveedores sigue los principios de la entidad**, evitando la colaboración con participantes fraudulentos o con actividades no éticas.

SAP es un ejemplo de la multitud de gigantes tecnológicos que han encontrado en la Blockchain un gran potencial para la optimización de las cadenas de producción. Entre sus proyectos, destaca su intento por aplicar este paradigma en el **sector agrícola**. La creciente preocupación por parte del consumidor sobre la procedencia y calidad de los alimentos a consumir (comide ecológica, productos de corral, etc.) hacen que éste sea un campo de aplicación con altos requerimientos en términos de transparencia y trazabilidad, aspectos asegurados en una Blockchain. En concreto, SAP trata de integrar la tecnología Blockchain con otras plataformas de rastreo y SCM ya existentes, de modo que esta nueva capa genere una visión compartida y global que incluya la actividad de todos los actores implicados "desde la granja al consumidor" [31].

Aunque es evidente que la Blockchain puede traer cuantiosas mejoras al área del SCM, esta revolución llegará a su *súmmum* cuando este paradigma se integre con otros que están causando un impacto significante en esta misma área, tales como la robótica, el *machine learning*, impresión 3D, etc. La siguiente ilustración presenta las aportaciones principales de la tecnología Blockchain a la cadena de producción:

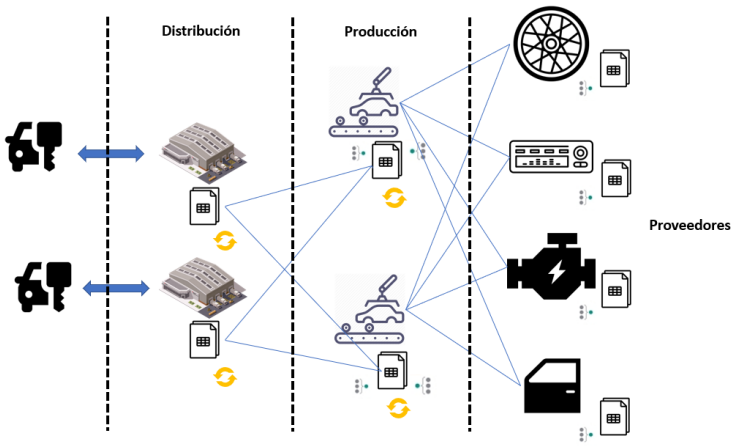


Figura 30. Aplicación de Blockchain a la Industria Automovilística

6.2.3 FORMACIÓN Y EDUCACIÓN

A pesar de que el sector financiero y el SCM son probablemente los que, a día de hoy, han experimentado una mayor acogida y un mayor número de casos de éxito, existen infinidad de campos compatibles con este paradigma: la sanidad, el sector inmobiliario, las entidades caritativas/ONG, la protección de datos, etc. La educación superior y la formación de profesionales constituyen dos áreas que se verían beneficiadas con el surgimiento de aplicaciones emergentes basadas en Blockchain.

A pesar de que la implantación de soluciones Blockchain para la optimización del **sector universitario** se encuentran aún en un estado de madurez reducido, cabe identificar las áreas que podrían experimentar un impacto positivo con la adaptación de este paradigma:

- La Blockchain permitiría el **remplazo definitivo de la documentación en papel**, propensa a problemas tales como el fraude o la pérdida. Las certificaciones/reconocimientos podrían almacenarse de un modo seguro en la cadena, permitiendo la automatización de ciertos procesos y una reducción en costes y tiempo de gestión.
- El **almacenamiento de toda información personal/académica** sobre alumnos/docentes en la Blockchain permitiría eliminar la necesidad de validación de credenciales y gestionar la propiedad intelectual de un modo eficiente y sencillo.
- Este paradigma podría usarse también como garante de la autoría y calidad del material académico, como se señalará en la siguiente gran sección de la tesis.

Además, numerosos centros docentes han comenzado a implantar nodos Blockchain con **fines académicos, de investigación y de gestión**. La **UP Comillas** ha sido, de hecho, pionera en este aspecto, con la implantación del primer nodo universitario Blockchain de España, conectado a la Red Alastria (Bankia, Telefónica, BBVA, Santander, Everis, Endesa, etc.) [32]. El nodo, cuyas altas necesidades de computación serán soportadas por los ocho servidores de ICAI (192 cores de proceso, 96 TB de memoria), se empleará con el fin de alcanzar fundamentalmente los siguientes objetivos:

- Una transformación en los mecanismos de **intercambio de información entre alumnos y universidades**, permitiendo una agilización y simplificación de procedimientos como la solicitud de notas o los traslados de expedientes.
- Una fuerte **innovación en el campo y las metodologías de investigación**, basado en la descentralización las experimentaciones, la elaboración de estudios conjuntos distribuidos por parte de varias universidades/organizaciones, etc.

La siguiente figura presenta el nodo instalado en la universidad. Cabe señalar que este pertenece a una Blockchain semipública con permisos, basada en Ethereum y **Quorum** (versión de Ethereum orientada al mundo empresarial, desarrollada por JP Morgan).

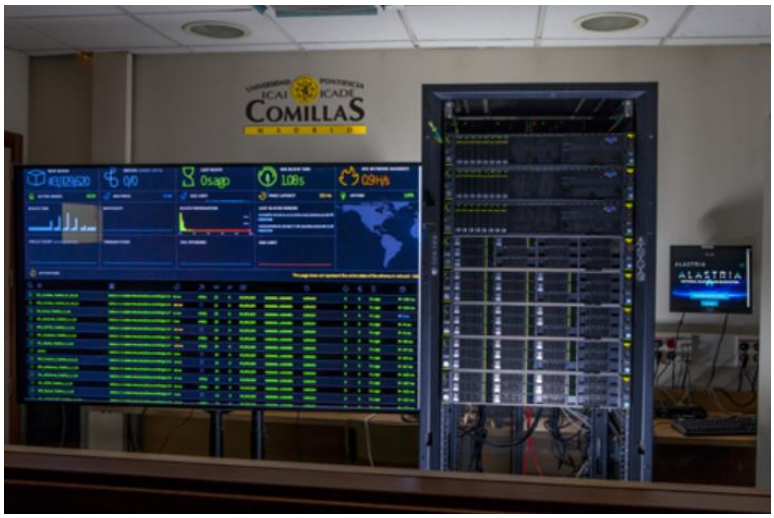


Figura 31. Nodo Blockchain instalado en la UP Comillas – ICAI [32]

Por otro lado, ya existen grandes organizaciones que también han apostado en la Blockchain para innovar en el campo de la **formación y el aprendizaje continuo de sus profesionales**. Por ejemplo, **BBVA** España y Argentina ha recurrido a la **tokenización** de las horas de formación cursadas y/o impartidas. Inicialmente, la plataforma asigna un número determinado de *tokens* a cada empleado, que podrá usar para la realización de cursos, y recuperar mediante la aportación de horas docencia a otros compañeros. Este modelo de "tokenomics" resulta de gran utilidad a la hora de emplear la Blockchain en multitud de campos y, en este caso, garantiza que todos los profesionales del banco sean parte activa del proceso de formación [33].

6.2.4 ASUNTOS DE GOBIERNO

Existen aspectos de la tecnología Blockchain ya mencionados en numerosas ocasiones (transparencia, ausencia de terceros, descentralización) que hacen de éste un paradigma idóneo para las labores administrativas y gubernamentales. El despliegue de Blockchains para estos fines podría constituir el inicio del fin de problemas como los lentos y complejos procesos burocráticos, la corrupción y los altos costes administrativos.

Una de las áreas de gobierno que se vería más beneficiadas con la adopción de la Blockchain sería el **sistema electoral**. Esta tecnología aportaría todas las propiedades necesarias para la implantación de una plataforma de votación [34]: tolerancia a fallos, imposibilidad de cambiar el pasado (durabilidad), altos niveles de seguridad ante ataques, fácil recuperación en caso de que estos ocurran, acceso a un recuento de votos común desde todos los nodos, etc.

Además del proceso electoral, la Blockchain podría ser de gran utilidad en la realización de **labores legislativas y regulatorias**. Un ejemplo de proyecto con este fin es **RegChain**, una Blockchain basada en *Proof of Concept* (PoC) y una serie de Smart Contracts que permiten la generación de informes regulatorios sobre procesos que manejen altos volúmenes de datos.

Numerosos países se han puesto ya manos a la obra en la elaboración de su "estrategia Blockchain". Cabe destacar el caso de **Estonia** [35], que con su proyecto E-Estonia ha revolucionado las áreas de:

- **Seguridad**, con su KSI Blockchain, una red empleada en todo el mundo y que permite verificar que otras redes/sistemas/datos no han sido comprometidos y/o manipulados, manteniendo además la privacidad de la información. Esta tecnología, capaz de soportar hasta 10^{12} transacciones por segundo, está siendo utilizada, entre otros, por el Ministerio de Justicia y el cuerpo de policía del país.
- **Sanidad**, mediante el empleo de la misma Blockchain (KSI). El historial de cada paciente del sistema sanitario estonio está almacenado de un modo seguro en la cadena, y es accesible exclusivamente por el personal autorizado.

6.2.5 INTERNET OF THINGS (IOT)

El paradigma del Internet de las Cosas (IoT) consiste, fundamentalmente, en la creación de redes de dispositivos (sensores, *gadgets*, etc.) capaces de transmitir y recibir información a través de internet (HTTP, FTP, etc.).

Cada uno de estos dispositivos puede actuar como cliente o como cliente – servidor (nodo), en función del poder de procesamiento y/o la capacidad de almacenamiento de datos que tenga el mismo.

No obstante, hasta el día de hoy este paradigma se ha enfrentado a diversos **desafíos** tecnológicos [40] en términos de **seguridad**, conectividad, compatibilidad y **longevidad**, *standards* y **generación de conclusiones** a partir de la información recolectada.

La combinación de este paradigma con la Blockchain podría dar lugar a un **IoT descentralizado**, en el que la cadena constituiría el *framework* para facilitar el procesamiento de transacciones y la **coordinación entre dispositivos** correlacionados. Además, se podría garantizar que los **datos empleados para un posterior análisis son precisos** y no han sido alterados, gracias al carácter incorruptible del registro, distribuido y mantenido por los diversos nodos de la red distribuida. De este modo, se puede llegar incluso a la creación de una **red distribuida de dispositivos autosuficientes.**

Un **ejemplo** estándar de la aplicabilidad de la Blockchain en este paradigma sería la creación de una red de electrodomésticos capaces de comunicarse entre sí, mandar datos sobre el consumo real de energía a lo largo del día, notificar averías, pagar la factura de la electricidad, etc.

En realidad, el modo de implementar esta red no difiere en absoluto con el proceso de creación de cualquier otro tipo de aplicación en la Blockchain:

- En primer lugar, es necesario generar una identidad para cada dispositivo, conectarlo a uno/varios nodos de la red Blockchain y programar el *frontend* (cliente Web, aplicación web, etc.) para poder acceder a los datos.

- Cada aparato en una casa tendría la posibilidad de transmitir/enviar información a un **enrutador central** que, a su vez, estaría conectado a la red de nodos de la Blockchain para la realización de las transacciones necesarias.

La arquitectura global de este ejemplo tendría el siguiente aspecto, en alto nivel:

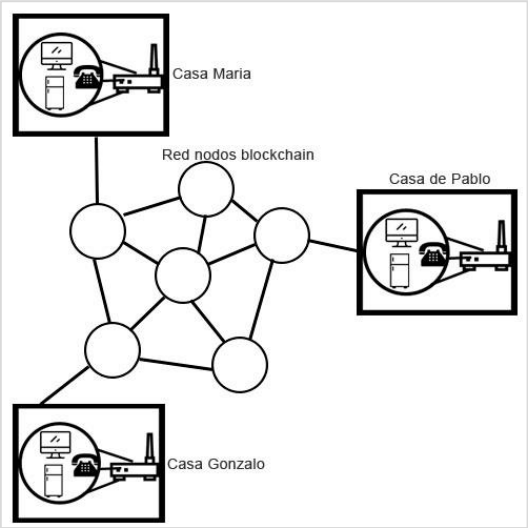


Figura 32. Integración Blockchain – IoT

Entre las **ventajas** que aporta la resolución de este ejemplo mediante Blockchain, cabe destacar la **inmutabilidad de los datos**. De este modo, cualquier avería, el gasto real de electricidad, las horas de funcionamiento…quedan registrados sin que ninguna de las partes pueda manipularlo de un modo fraudulento.

Además, la incorporación de ***Smart Contracts*** en este tipo de aplicaciones resolvería uno de los problemas que las redes IoT están experimentando actualmente: análisis de información imprecisos e inválidos por la presencia de excesivos *outliers*, *flaws*, etc. en los datos recolectados. Esta situación podría mejorar con la incorporación de un *Smart Contract* en contacto con los dispositivos reales, que fuera modelando y filtrando los datos de entrada a los sistemas analíticos.

Actualmente, existen numerosos proyectos y entidades trabajando en la integración de estos dos paradigmas, tales como IOTA, RuffChain o Modum.

Capítulo 7. PROPUESTA DE CASO DE USO. BLOCKCHAIN Y OTROS PARADIGMAS

La última sección de contenido del Trabajo de Fin de Máster se centra en la descripción con un mayor nivel de detalle de la integración de la Blockchain con otro paradigma tecnológico. El Internet de las Cosas (IoT) es, sin lugar a dudas, el paradigma tecnológico del que más se está hablando a la hora de aplicar la Blockchain. No obstante, éste no es el único, por lo que se ha decidido orientar este estudio a una temática menos investigada hasta la fecha.

En esta ocasión se ha optado por analizar la posibilidad de establecer una **relación simbiótica entre la Blockchain y la Web Semántica**, un paradigma nacido hace bastante tiempo pero sin un nivel de implantación muy avanzado.

Para la descripción de esta labor de análisis, en primer lugar se hará una descripción del paradigma *Semantic Web* para, una vez comprendidos una serie de conceptos relevantes, proponer el modo en el que éste y la cadena de bloques pueden integrarse con el fin de mejorar numerosos campos/ejemplos de aplicación.

7.1 INTRODUCCIÓN A LA WEB SEMÁNTICA Y LAS ONTOLOGÍAS

A lo largo del tiempo, el acceso y manejo de la información se ha realizado de distintas maneras. Antes de la aparición de la *World Wide Web* (WWW), existías núcleos aislados de información (bibliotecas, librerías privadas, laboratorios, etc.), en lo que ésta estaba disponible mayoritariamente en un formato físico.

Con la llegada del WWW y lo que comúnmente se conoce como **Web 1.0**, se conseguía un gran avance en términos de accesibilidad de la información gracias a la aparición del **hipervínculo**, que ofrecía al usuario la posibilidad de acceder a miles de documentos interconectados a través de un simple “clic”. La **Web 2.0**, el boom de las aplicaciones

(Facebook, LinkedIn, etc.) vino acompañada de la creación de nuevos núcleos ricos en información, aunque la conexión entre estos núcleos es normalmente escasa e insuficiente.

La visualización de la información en la era Web es posible gracias a la tecnología **HTML**, que no es más que un lenguaje que permite describir al equipo o al *browser* cómo representar la información accesible al usuario. No obstante, esta tecnología no incluye ningún mecanismo que permita la **comprensión del contenido** a mostrar.

Es precisamente este aspecto la principal motivación del nacimiento de la **Web 3.0:** si se consigue que los **equipos entiendan el contenido, estos podrán conocer al usuario y, por tanto, se convertirán en un actor activo de la navegación**. En el paradigma de la Web 3.0, o ***Semantic Web,*** no se busca establecer una interconexión entre *websites* o documentos, sino entre **datos o hechos**.

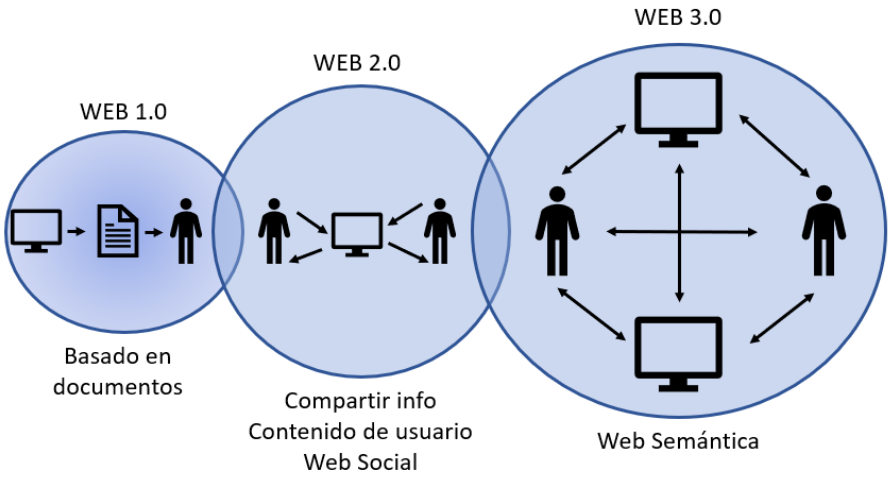


Figura 33. Evolución del Paradigma Web

En las siguientes secciones se señalarán los aspectos fundamentales para entender el paradigma de la Web Semántica, una corriente presente en el panorama tecnológico durante más de 20 años, pero que no ha terminado de asentarse debido a la presencia de varios desafíos. Esta explicación ayudará a comprender mejor la forma en la que la Blockchain puede colaborar en la resolución de estos retos.

7.1.1 LA WEB DE HOY Y LA PROPUESTA POR *SEMANTIC WEB*

La Web, accesible hoy en día por la mayoría de la población, cuenta con representaciones de absolutamente todo, de un modo u otro. Además, los motores de búsqueda (Google, Yahoo, etc.) facilitan significativamente el acceso a todo este contenido.

Aparte de ser una fuente casi ilimitada de contenido e información, la "Web de hoy" facilita numerosas actividades diarias, tales como la realización de compras, la planificación de viajes, las comunicaciones con distintos propósitos, etc. No obstante, navegando por la red se pueden identificar también algunos **aspectos a mejorar**:

- La Web ha alcanzado un tamaño inmenso, con infinidad de replicaciones del mismo contenido, resultando parte de estas inválidas.
- El modo en el cual está codificada la mayor parte de la WWW hoy en día impide que la máquina pueda reproducir y comprender los contenidos de los distintos sitios.
- Los motores de búsqueda, aunque de gran utilidad, no suelen ofrecer resultados personalizados a cada usuario, sino que estos están basados en pesos o incentivos económicos.

La mayor parte de estos problemas tienen el mismo origen: **la ausencia de una expresión del significado** en las implementaciones del contenido web actual. Este contenido se presenta mediante el empleo de formatos (HTML) e interfaces (párrafos, formularios) que proveen una representación gráfica entendible por personas pero no por máquinas, haciendo casi **imposible la automatización de tareas mediante *software* en substitución del humano** [44].

La **Web Semántica** ha buscado, desde su nacimiento, acabar con las limitaciones recién mencionadas mediante la **introducción de descripciones explícitas de concepto/significado del contenido de la Web**, haciéndolos procesables con máquinas y facilitando una clasificación, estructuración e interconexión de todos estos recursos.

Las siguientes secciones describirán los recursos conceptuales y tecnológicos que la *Semantic Web* ha propuesto para la consecución de sus objetivos.

7.1.2 URLS, URIS Y NAMESPACES

En esta primera sección de carácter técnico se procede a describir brevemente una serie de herramientas básicas empleadas constantemente en la Web Semántica, e indispensables para comprender correctamente la tecnología RDF: las URLs, URNs, URIs e IRIs.

En primer lugar, el término **URI (*"Universal Resource Identifier")*** hace referencia a una secuencia de caracteres que identifica un recurso, pudiendo ser éste físico o abstracto. Cuando la URI representa exclusivamente un recurso (libro, artículo, etc.), independientemente de su localización, ésta recibe el nombre **de URN (*Universal Resource Name***, poco empleados). Por otro lado, cuando la URI describe tanto un recurso como la localización del mismo, recibe el nombre de **URL (*"Universal Resource Locator"***) [45].

Con el fin de adaptar el empleo de todo tipo de carácter (latino y no latino) para la descripción de URIs surge el término **IRI (*Internationalized Resource Identifier*)**. No obstante, se recomienda el uso exclusivo de caracteres soportados por URI siempre que se pueda, para evitar problemas de incompatibilidades.

Por último, cabe mencionar uno de los aspectos fundamentales de la Web Semántica, declarado actualmente como una *best practice* de diseño: la introducción de **metadatos** en el contenido web. Por lo general, un dominio/concepto es definido por un *set* de elementos/metadatos. Debido a la posibilidad de que existan algunos elementos de este *set* que coincidan en nombre con los de otro, surge la necesidad de diferenciar estos dominios mediante el empleo de **namespaces**. La forma general (en XML) de emplear y asignar valores a estos sets de atributos se muestra en la siguiente figura:

```
<?xml version="1.0"?>

xmlns:persona="Url donde tengamos el set de metadatos"

<persona:nombre>Javier</persona:nombre>
<persona:edad>24</persona:edad>
```

Figura 34. Namespace. Empleo en XML

7.1.3 RESOURCE DESCRIPTION FRAMEWORK (RDF)

En esta sección se explicará conceptualmente en los que consiste RDF para, después, mostrar un ejemplo para comprender la idea detrás de este *framework* y su significante utilidad.

Esencialmente, RDF es un lenguaje o modelo para la **representación de recursos**, entendiéndose como recurso a cualquier aspecto que pueda localizarse vía una URL. La unidad básica para la descripción de recursos recibe el nombre de **triplete RDF**. Esta unidad es, en realidad, representada mediante **grafos** y recibe el nombre de triplete porque consta fundamentalmente de tres elementos: **sujeto**, propiedad o **predicado** y valor u **objeto**.

El estándar RDF es altamente compatible con representaciones tradicionales: **cualquier dato relacional puede ser también representado en forma de triplete**, como se ilustra en la siguiente figura:

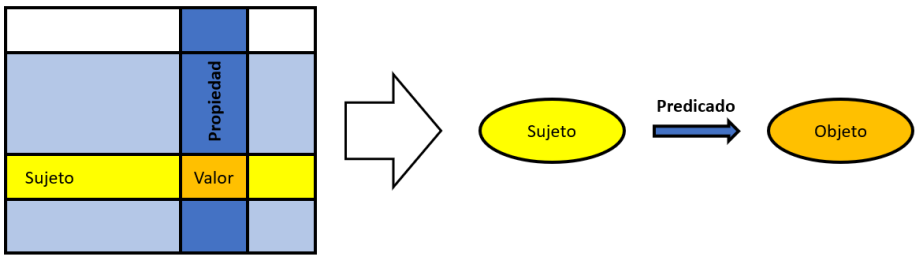


Figura 35. Adaptación de tuplas a tripletes RDF

Uno de los grandes potenciales de esta representación es la **gran diversidad de combinaciones** que acepta: se pueden agrupar varios tripletes que compartan el mismo sujeto, realizarse uniones de tripletes mediante el uso de un predicado, etc.

Además, cabe mencionar que, por lo general, los elementos del triplete RDF suelen representarse del siguiente modo:

- El Sujeto, mediante el empleo de PREFFIX (una especie de alias) o URIs completas.
- El Predicado, mediante el empleo de PREFFIX (una especie de alias) o URIs completas.

- El Objeto, mediante el empleo de URIs o literales (especificando el tipo de dato, o no).

Es crucial que la representación de estos tres elementos no genere ningún tipo de ambigüedad. Existen ocasiones en los que el sujeto o el objeto pueden estar representados en forma de ***blank node***, un nodo vacío debido a desconocimiento sobre el recurso o requisitos de anonimato, entre otros.

Una vez introducidos los conceptos básicos de este *framework*, se debe hacer referencia a la existencia de distintos **formatos posibles para su representación**. Entre estos, destacan:

- **Turtle**, un formato basado en texto enfocado sobre todo a la comprensión humana.
- **RDF/XML**, la serialización oficial de RDF para XML.
- **RDFa**, un mecanismo para embeber RDF dentro de archivos HTML.

La siguiente figura muestra un ejemplo de código basado en uno de estos formatos [48]:

```
<div vocab="http://schema.org/" typeof="Person">
  <span property="name">George Bush</span>, the
  <span property="disambiguatingDescription">41st President of the United
  States</span>
  is the father of
  <div property="children" typeof="Person">
    <span property="name">George W. Bush</span>, the
    <span property="disambiguatingDescription">43rd President of the United
    States</span>.
  </div>
</div>
```

Figura 36. Ejemplo código HTML con uso de RDFa

Si se desea almacenar un gran número de tripletes, guardar todos ellos en un fichero de texto empleando un formato como Turtle o RDF/XML (serialización) no es la solución más eficiente o conveniente. Este problema ha provocado el surgimiento de ***software* de almacenamiento optimizado para el soporte de tripletes**. Fundamentalmente, estos sistemas indexan la información y deciden qué datos guardar en memoria y cuándo, evitando en numerosas ocasiones el almacenamiento repetido del mismo recurso. Estos sistemas reciben el nombre de ***triple stores***.

7.1.4 SPARQL (SIMPLE PROTOCOL AND RDF QUERY LANGUAGE)

Como puede intuirse, SPARQL es básicamente un lenguaje para **descripción de consultas o *queries* para RDF**, aunque actualmente existe también un estándar que soporta OWL.

SPARQL es tanto un lenguaje de consulta como un protocolo de acceso a una base de datos RDF, lo que implica que permite al usuario el acceso a grafos RDF u otras fuentes de datos que puedan ser convertidas al estándar anterior mediante el uso de *middlewares*.

Este lenguaje permite la elaboración de consultas constituidas por triples de RDF, conjunciones (&&) y/o disyunciones (||), de modo que el motor SPARQL devolverá el *set* de triples del modelo consultado que coincida con esta consulta. Las *queries* de este estándar son muy similares a las empleadas para consulta de bases de datos relacionales (SQL), como se puede apreciar en la siguiente figura [46]:

```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
PREFIX books:<http://www.dummies.com/books#>

SELECT  ?book
WHERE {
         ?book rdf:type books:Books .
         ?book books:author
            http://me.jtpollock.us/foaf.rdf#me .
       }
ORDER BY ?book
```

Figura 37. Ejemplo de query en SPARQL

Como se puede intuir, esta consulta busca libros cuyo autor es el mismo que el de la misma consulta (especificándose a sí mismo mediante su URI). El resultado dado por el motor SPARQL estará constituido por las URIs de todos los libros que coincidan con la condición.

7.1.5 ONTOLOGÍAS Y OWL (WEB ONTOLOGY LANGUAGE)

Antes de pasar directamente a la explicación de estos dos aspectos fundamentales, es necesario hacer una breve referencia al estándar **RDFS** (**RDF *Schema***). RDFS es un lenguaje empleado para la **descripción de vocabularios**, siendo estos *sets* de términos (categorías), equivalentes a los ya descritos namespaces.

Estos vocabularios, que no dejan de ser *sets* de tripletes, pueden ser reutilizados y almacenados empleando estándares como RDFS y OWL.

El lenguaje RDFS es, en realidad, otro set de propiedades, que permite la creación de:

- Nuevas **clases** de recursos, lo que permite una mayor asignación de metadatos
- **Propiedades**, en las cuales habrá que especificar un **dominio** (el tipo de sujeto que contará con esa propiedad) y un **rango** (el tipo de objeto/valor al que podrá hacer referencia la propiedad).

La siguiente figura pretende servir de explicación complementaria a lo recién descrito, para aportar una visión introductoria a todo lo que envuelve RDFS:

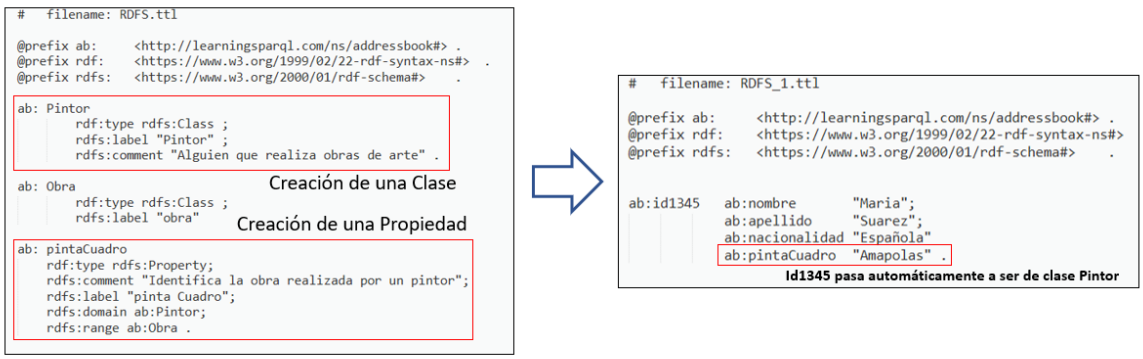


Figura 38. RDFS. Utilidad

Como se puede observar en la figura, usando procesadores que soporten los estándares de *Semantic* Web, si se añade una propiedad de una clase a los metadatos de un recurso, este recurso se asociará automáticamente a dicha clase.

Gracias a las tecnologías de la Web Semántica, **la información se va estructurando automáticamente,** facilitando significativamente los procesos de inferencia en estructuras de datos complejas.

En este contexto, y sobre el estándar RFDS, surge el lenguaje **OWL**, que de nuevo vuelve a constar de una serie de tripletes RDF, consultables por tanto mediante lenguajes como SPARQL. Antes de hacer referencia al lenguaje en sí, se debe introducir el término **ontología**, que hace referencia a **jerarquías** complejas de conceptos con atributos y relaciones entre distintos vocabularios o clases, definiendo una terminología **consensuada** para definir redes de datos interrelacionados. Por lo general, cada ontología describe **dominios muy específicos**, como áreas de investigación o ramas artísticas concretas, facilitando la compartición de datos entre distintos actores e instituciones.

El lenguaje **OWL** ofrece una funcionalidad adicional a RDFS mediante la introducción de **elementos que expresan relación** entre sujetos o recursos. Ejemplos de estos elementos son:

- Owl:SymmetricProperty. Como alternativa a la creación de una propiedad mediante RDFS, este elemento indica "mutualidad" entre los dos nodos que une este predicado (ejemplos: es cónyuge, es familiar, es amigo, etc.).
- Owl:inverseOf. Habitualmente empleado para expresar que una clase es complementaria u opuesta a otra (ejemplo: médico/paciente).

Esta funcionalidad añadida potencia aún más los mecanismos de inferencia y aprendizaje automático ya mencionados: no es necesario explicar absolutamente todo sobre cada dato o recurso.

El objetivo detrás de todo esto es que la *Semantic Web* esté creada por infinidad de nodos interconectados mediante clases, propiedades y relaciones predefinidas por consenso por los autores de cada ontología. La **adopción de ontologías comunes** es imprescindible para que todos los actores dentro de la Web Semántica (creadores o consumidores) tengan la posibilidad de trabajar autónomamente estando seguros de que todas las piezas encajan.

7.1.6 LINKED DATA

Esta primera sección del capítulo concluye con el aglutinamiento de todos los mecanismos recién descritos para la constitución de lo que se conoce como *Linked Data*, un conjunto de *best practices* para la construcción de una infraestructura que facilite la distribución y compartición de datos en la Web.

Esta optimización de la conexión entre datos/aspectos en la red podría traer consigo beneficios tales como incrementos en usabilidad, interoperabilidad y productividad, provocado este último fenómeno por la **capacidad de las maquinas de explotar el dato**.

El conjunto de buenas prácticas propuestas en el *framework* del Linked Data puede sintetizarse en los siguientes **principios**:

- Uso de **URIs** para el nombramiento y la identificación de aspectos en la Web.
- Estas URIs deben estar basadas en el protocolo **HTTP** (no FTP, *mailto*, SMTP, etc.). De este modo, el usuario puede acceder a más información sobre el contenido, aumentándose la interoperabilidad.
- Cuando alguien (usuario o máquina) acceda a una URI, debería encontrar **información útil y representada siguiendo algún estándar** (grafo RDF, SPARQL, etc.), para que sea comprensible tanto por humanos como por *software*.
- Para promover la integridad de la Web Semántica, cada una de estas descripciones debería además incluir links a otras URIs relacionadas con el tema/aspecto en cuestión.

Actualmente existen numerosos proyectos que promueven la participación masiva para la creación de una web de datos interconectados. Cabe destacas la iniciativa ***Linked Open Data***, en la que participan entidades como Wikipedia, la BBC o Thomson Reuters. Mediante el volcado de sus datos en la red, y la publicación de una serie de repositorios de datos traducidos a RDF, se está generando un espacio global de datos sobre multitud de aspectos (personas, publicaciones, música, genes, etc.). Actualmente se estima que proyectos como éste han generado unos 32000 millones de tripletes RDF enlazados entre ellos [47].

7.2 BLOCKCHAIN Y SEMANTIC WEB

A pesar de tener un potencial inmenso y servir de utilidad para tendencias tecnológicas actuales (*Machine Learning*, AI), la Web Semántica no ha experimentado los niveles deseados de aceptación y adopción desde su surgimiento hace ya varios años. Esta sección pretende aportar una serie de planteamientos e ideas sobre cómo la Blockchain podría impulsar la Semantic Web. Asimismo, se plantearán una serie de propuestas que resuelven desafíos actuales de la cadena de bloques mediante la introducción de elementos de la Web Semántica.

El establecimiento de esta relación simbiótica entre ambos paradigmas puede llevarse a cabo de distintas maneras [50]. La investigación y análisis realizado se ha centrado en las siguientes tres:

A) <u>MAPEO DEL CONTENIDO DE LA BLOCKCHAIN EN ESTÁNDARES DE *SEMANTIC WEB*</u>

Una forma clara de integrar ambos paradigmas siguiendo este marco referencial es la **creación de ontologías que describan las estructuras internas de las distintas redes Blockchain existentes**. La consecución de este objetivo podría traer beneficios, tales como:

- La creación y compartición de información general sobre todas estas Blockchains, tanto para personas como para elementos *software*
- Permitir la reutilización de ciertos dominios
- Establecer una separación entre lo que es el dominio nativo de cada Blockchain y su correspondiente dominio operacional
- Etc.

Aunque pudiese parecer que este tipo de propuestas no provocarían un fuerte impacto sobre la tecnología Blockchain, se considera que acciones como ésta podrían simplificar el aprendizaje y entendimiento del paradigma, pudiendo provocar un incremento en la atracción por parte de individuales e instituciones.

Desde el nacimiento de las aplicaciones basadas en cualquier red Blockchain, ha existido la necesidad de explorar el contenido y/u operaciones de la misma. Esta necesidad está actualmente cubierta por los ***Blockchain Browsers***, como el ya mencionado y empleado etherscan. Todas estas herramientas **hacen uso de bases de datos relacionales** o basadas en un esquema clave – valor, no en estándares semánticos como RDF.

La **creación de ontologías completas que representaran el dominio y relaciones existentes dentro de una Blockchain** traería consigo numerosos beneficios aunque también conlleva la realización de acciones con cierta complejidad. Los principales aspectos a considerar dentro de esta propuesta serían:

- En caso de querer incluir aspectos operacionales dentro de las ontologías, se deberían tener en cuenta los **requisitos de anonimato y privacidad** de la red en cuestión. No obstante, este aspecto puede gestionarse fácilmente con el empleo de los ya mencionados ***blank nodes***.
- La creación de ***Blockchain Browsers*** con información almacenada en formatos semánticos (RDF, OWL) convertiría dichos datos en recursos comprensibles por máquinas, elemento imprescindible si se produce la expansión esperada del paradigma Blockchain.
- Aunque la creación de todas estas ontologías parezca una tarea compleja y tediosa, los mecanismos de **inferencia y aprendizaje automático** propios de procesadores compatibles con *Semantic Web* simplificarían significativamente el proceso.

En definitiva, la aplicación de OWL u otras tecnologías para el mapeo y la descripción de contenido Blockchain traería importantes ventajas en términos de aprendizaje e **interoperabilidad**. Se espera que, en un futuro, existan conexiones constantes entre Blockchains distintas e incluso se creen "Blockchains de Blockchains". Este ecosistema incluye intrínsicamente la necesidad de mapeo de las distintas cadenas para lograr estas vinculaciones.

Además, en el caso de lograrse está representación de las operaciones dentro de una cadena en forma de ontología, se podría numerosas posibilidades a la tecnología Blockchain. Por

ejemplo, mediante la inclusión de **clases, propiedades y relaciones específicas a una ontología** se podría aplicar la cadena de bloques en sectores que hoy en día suponían un desafío por los altos requerimientos en términos de regulación o cualquier otro aspecto formal.

B) ALMACENAMIENTO DE CONTENIDO RDF DIRECTAMENTE EN LA BLOCKCHAIN

Este es, probablemente, el modo de integración que beneficie en mayor medida al paradigma de la Web Semántica.

Actualmente, la creación y publicación de ontologías en la WWW cuenta con diversos desafíos. Entre estos, en este contexto cabría destacar:

- Centralización. La mayoría de ontologías se encuentran almacenados en un pequeño número de repositorios.
- Consistencia. Muchos de estos repositorios son de libre acceso y editado, por lo que cualquiera puede modificar los datos relativos a una ontología.
- Tolerancia a fallos. Debido a la concentración de toda la información en escasos repositorios, se da el riesgo del *single-point-of-failure*.
- Interoperabilidad. Todos estos repositorios no están interconectados, por lo que la colaboración es tediosa y con frecuencia se pueden encontrar ontologías repetidas y distintas en estos "almacenes".

Una vez descritos los problemas anteriores y teniendo en cuenta todo lo descrito hasta el momento en el TFM, resulta casi evidente que la tecnología Blockchain podría aportar (*a* priori) enormes beneficios al mundo de las ontologías. La creación de una cadena de bloques basada en un **registro único, distribuido y descentralizado de ontologías** acabaría con todos los riesgos antes mencionados. Todo cliente de la Blockchain contaría con su copia del registro o repositorio común, por lo que los problemas de interoperabilidad y colaboración desaparecerían por completo. Además, toda modificación por una ontología sería verificada por los miembros de la red antes de realizar una actualización sobre el

repositorio, facilitando así intentos de modificación fraudulenta o errónea de recursos semánticos.

No obstante, el almacenamiento dentro de una Blockchain de ontologías y, en general, cualquier recurso basado en tripletes RDF, requiere la realización de un análisis de viabilidad previo para el análisis de dos riesgos fundamentalmente [49]:

- Problemas de **compatibilidad entre los requerimientos de tamaño y velocidad** de ficheros de este tipo y las prestaciones actuales de las Blockchains.
- La introducción, modificación, consulta, etc. de ontologías dentro de, por ejemplo, la red Ethereum, supondrían un **cierto coste en Gas y Ether**. Por tanto, sería necesario realizar también un **análisis de viabilidad financiera** de estas propuestas, comparando el gasto antes mencionado con el que supondría el empleo de una Blockchain privada, por ejemplo.

Teniendo en cuenta estos aspectos, se debería proceder a decidir el **modo en el que se va a almacenar la información semántica dentro de la cadena de bloques**. Como ha sido la red más analizada hasta la fecha, la siguiente tabla aporta las distintas alternativas que serían posibles si se construyera el repositorio de ontologías sobre Ethereum:

LOCALIZACIÓN	DESCRIPCIÓN	PROS	CONTRAS
Campo "datos" del Bloque	-	No necesita tener una longitud fija y no se puede modificar	Es caro y la información se almacena en formato hexadecimal
Smart Contracts	Almacenamiento en forma de clave-valor o formato de texto	No necesita tener una longitud fija y es de fácil acceso	Es caro y la información podría modificarse con cierta facilidad
Logs de Eventos	Almacenamiento en el histórico de transacciones	Alternativa más económica	El histórico puede perderse, la durabilidad no estaría garantizada

Tabla 5. Formas de almacenamiento de contenido RDF en Ethereum

C) CREACIÓN DESDE CERO DE REDES BLOCKCHAIN BASADAS EN ESTÁNDARES DE *SEMANTIC WEB*

La última alternativa para la integración de estos dos paradigmas consiste en la creación de un *fork*/bifurcación de una de las redes actuales, o directamente una nueva Blockchain, cuyos **protocolos de intercambio de datos internos y operacionales estén basados en estándares semánticos**, soportando tripletes RDF.

En definitiva, el análisis y estudio de proyectos basados en cualquiera de estas tres alternativas podrían con mucha probabilidad aportar beneficios a ambos paradigmas. La Blockchain constituye la plataforma perfecta para impulsar la expansión de la *Semantic Web* y sus principios, incorporando a los *datasets* y ontologías un aspecto fundamental: **confianza**. Asimismo, el uso de estándares semánticos promueve buenas prácticas como el empleo de formatos de representación e intercambio de datos comunes haciendo uso del *framework* RDF y aportando a la tecnología dos aspectos de gran importancia: **interoperabilidad y estandarización**. El empleo de los principios de *Linked Data* en aplicaciones basadas en Blockchain podría facilitar significativamente el proceso de expansión que está experimentando la tecnología en la actualidad; algunas *best practices* recomendadas en este escenario serían:

- Toda Blockchain y *Smart Contract* debería contar con una identificación única e irrepetible.
- El contenido de un *Smart Contract* debe estar implementado en un formato entendible por elementos *software* (*bots*).
- La creación de ontologías especializadas podría resolver las necesidades de gobernabilidad de algunas Dapps, mediante la inclusión de metadatos de las mismas en los distintos *Smart Contracts* de una infraestructura, por ejemplo.
- Las cadenas de bloques deben contar con protocolos que soporten *request* y *response* solicitando y aportando datos en RDF.
- Las *queries* en SPARQL deberían poder involucrar a distintas cadenas interconectadas a la vez.

7.2.1 IMPLANTACIÓN DE LA PROPUESTA

Esta última sección pretende concluir el capítulo aportando una visión más práctica sobre las alternativas existentes para la integración entre Blockchain y *Semantic Web*, y el tipo de casos de uso que ésta puede satisfacer.

7.2.1.1 Ejemplo de Aplicación I: Sistema de Gestión de Ontologías

Como ya se ha mencionado en la sección anterior, la expansión y utilidad de la red semántica se verían beneficiadas con el empleo de la Blockchain para el almacenamiento, acceso y modificación de las distintas ontologías disponibles en la red. No obstante, cabe recordar que esta aplicación traía consigo una serie de desafíos, a destacar:

- Las restricciones en términos de velocidad de procesamiento y consulta que hoy en día siguen presentando la mayoría de las Blockchain.
- Los costes (Gas en Ethereum) que supondrían las consultas/modificaciones de ontologías sobre la Blockchain.

Teniendo en cuenta estos riesgos y el estado de madurez actual de la tecnología Blockchain, se considera que la solución óptima para la gestión de las ontologías estaría basada en una arquitectura similar a la representada a continuación:

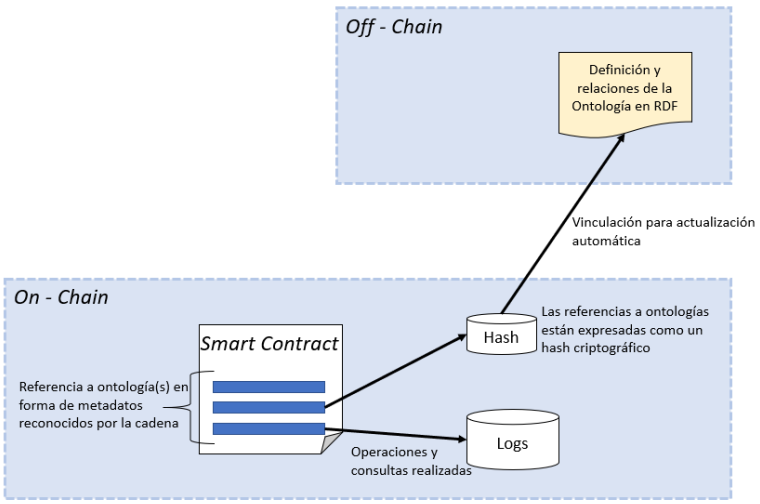


Figura 39. Gestión de Ontologías en la Blockchain

El esquema básico de arquitectura recién mostrado se basa principalmente en dos principios:

- Mantener la descripción completa de las ontologías en un repositorio *off-chain* y almacenar *hashes* que hagan referencia a estas en la Blockchain.
- Adaptar los protocolos de intercambio de datos para que soporten estándares basados en tripletes RDF.

La parte del aplicativo localizado dentro de la Blockchain estará basada fundamentalmente en la creación y manejo de *Smart Contracts* con funciones que permitan el acceso al repositorio de *hashes* que referencian a las ontologías y la realización de operaciones sobre las mismas. Por tanto, estos contratos deben ser capaces de comunicarse con el repositorio *off-chain* para realizar las actualizaciones pertinentes sobre la información semántica.

El diagrama mostrado es una mera representación en alto nivel del *backend* del aplicativo. Además de esto, se debería definir una arquitectura *frontend* y una serie de componentes que permitieran la comunicación entre ambas, abstrayendo al usuario de la complejidad técnica sobre la que estaría construida la plataforma en cuestión.

Lo que se consigue con este planteamiento es una "autogestión" del repositorio de ontologías *off-chain*. Los usuarios tendrían acceso exclusivamente a la plataforma implantada para realizar sus tareas de consulta o modificación de ontologías. De este modo, el repositorio que contiene la estructura completa de ontologías estaría disponible en todo momento, evitándose además riesgos de ejecución de modificaciones no autorizadas o creación de ontologías repetidas.

Además, se ha considerado mantener el repositorio de ontologías fuera de la Blockchain con el fin de garantizar los requisitos en términos de latencia: los datos a gestionar y almacenar por los contratos son, de este modo, significativamente más ligeros (operaciones, *queries*, breves secciones de texto). Una posible opción para el *hosting* de este repositorio sería el empleo de **IPFS** (*Inter Planetary File System*), una solución empleada para el referenciado de grandes ficheros a Dapps, que soporta el uso del repositorio de *hash* que apunten al recurso real descrito en el diagrama de arquitectura.

7.2.1.2 Ejemplo de Aplicación II: Semantic Blockchain y SCM

Anteriormente, se ha mencionado la gran compatibilidad existente a la hora de incorporar el paradigma de la **Blockchain** en infraestructuras para la gestión de cadenas de producción. Además, numerosas entidades de *retail* u organizaciones que cuenten con sistemas de SCM hacen hoy uso de principios propios del ***Linked Data***, empleando ontologías predefinidas para estandarizar la terminología empleada en cada uno de los procesos por los que pasa el producto en cuestión, obteniéndose así importantes beneficios en términos de **cumplimiento de regulaciones e interoperabilidad.**

Si ambos paradigmas son compatibles y útiles con el sector del *SCM*, es evidente que una combinación simultánea de ambos en estos sistemas optimizaría varios aspectos, a destacar:

- La Blockchain resultaría de gran ayuda para el registro de balances y transferencias de activos a lo largo de toda la cadena de un modo fiable y transparente. Un modo sencillo de llevar a cabo esto sería la *tokenización* de la cadena de producción.
- Por otro lado, si el flujo del producto a través de los distintos participantes de la cadena (proveedores, productores, distribuidores, etc.) se replicará también en un formato RDF, se simplificaría el proceso de trazabilidad y/o autenticidad de los productos, pudiéndose conocer en todo momento el origen del producto final. La interpretación de estos datos expresados en forma de ontología resulta más sencilla que la comprensión de los datos sin procesar dentro de la Blockchain, pudiéndose generar certificados de proveniencia y/o autenticidad más fácilmente.
- La traducción del flujo de datos dentro de la cadena en un formato *machine – readable* permitiría la realización de tareas de análisis sobre el rendimiento de la cadena, permitiendo esto la obtención de ideas para la optimización del comportamiento operativo de todo el proceso.

En definitiva, la relación a tener en cuenta entre estos dos paradigmas es claramente simbiótica, la Blockchain aportaría el impulso que *Semantic Web* necesita, y la introducción de elementos semánticos a la cadena de bloques reduciría su complejidad y facilitaría el procesado y análisis de datos relativos a todas las operaciones ejecutadas en la misma.

Capítulo 8. CONCLUSIONES Y TRABAJOS FUTUROS

El surgimiento de la tecnología Blockchain puede ya señalarse, sin ninguna duda, como un hito fundamental en la historia de la tecnología y el mundo empresarial. Tras la realización de este Trabajo de Fin de Máster, resulta obvio afirmar el hecho de que este paradigma constituye mucho más que un mecanismo para la creación/transferencia de bienes financieros.

La Blockchain se sitúa en el *core* de la cuarta revolución industrial, aportando una solución tecnológica descentralizada para compartir y distribuir datos de un modo autónomo entre un inmenso número de participantes, sin necesidad de que exista una confianza previa entre los mismos. De este modo, el paradigma presenta un gran potencial a la hora de conectar millones de personas (*online* o no), optimizar la eficiencia de un amplio rango de sectores empresariales y, en general, reinventar la realidad tecnológica a través de una mejora brutal en la gestión de activos.

A pesar de que, inicialmente, la tecnología contaba con partidarios y detractores a partes iguales, actualmente la mayoría de organizaciones líderes en multitud de sectores están invirtiendo y apostando por el potencial de la revolución Blockchain. No obstante, aún se aprecia un carácter bastante experimental en la mayoría de proyectos llevados a cabo.

Se puede concluir, además, que la consecución de los objetivos del proyecto ha sido generalmente satisfactoria. Se ha llevado a cabo una investigación general de la tendencia tecnológica, abarcando todos los aspectos claves del paradigma y realizando una serie de aportaciones propias.

Tras esto, se ha dedicado parte de la tesis al análisis de algunas de las tecnologías y soluciones Blockchain con un mayor grado de implantación en la actualidad, tratando y planteando aspectos teórico-prácticos e implementando *Smart Contracts* con el fin de lograr una familiarización inicial con distintas soluciones *software* y lenguajes de programación.

Además, como definía otro de los objetivos del trabajo, se ha estudiado la integración de la Blockchain con otros paradigmas, profundizando en el establecimiento de una relación simbiótica entre esta y la Web Semántica, y aportándose un modelo inicial de arquitectura que resuelve los desafíos a los que se enfrenta su principal elemento: las ontologías.

Uno de los aspectos más poderosos a la hora de implantar soluciones basadas en Blockchain es, sin duda, la posibilidad de gestionar prácticamente cualquier activo físico mediante su *tokenización*. No obstante, esta transformación a activo digital debe realizarse de un modo adecuado, y la estructura interna de la solución debe customizarse para satisfacer los requisitos del escenario en cuestión, siendo necesaria la toma de decisiones sobre una serie de aspectos tales como: el tipo de cadena, la estructura del dato, el protocolo de consenso, la frecuencia de creación de bloque, le introducción o no de mecanismos de privacidad, etc.

A pesar de que el trabajo extraído de este proyecto permite concluir en la necesidad de apostar de lleno por este paradigma, la Blockchain se enfrenta también a una serie de riesgos y desafíos que podrían truncar su rápida expansión. Aunque existen una serie de desafíos tecnológicos a afrontar por la Blockchain, no se consideran excesivamente preocupantes, debido al mero hecho de que a diario surgen soluciones y alternativas que están progresivamente dando solución a estos problemas (latencias, excesivo consumo eléctrico, etc.). No obstante, no se puede señalar lo mismo sobre el ritmo que sigue la capacidad de adaptación humana y administrativa.

Es por esto por lo que las posturas reacias de ciertos órganos de gobierno y entidades regulatorias se consideran como el mayor riesgo que podría provocar una frenada en la expansión de la Blockchain. La mayoría de la población tiene una idea muy vaga sobre lo que es la Bitcoin, y la mayoría de profesionales no técnicos (o financieros) no han oído hablar de la tecnología Blockchain. Es fundamental que se promueva la comprensión del paradigma para que, de este modo, órganos centrales y regulatorios no lo vean como un enemigo, sino como un gran aliado. Una estrategia para conseguir este acercamiento es la inversión de esfuerzos en la estandarización del paradigma, esencial para su uso empresarial. El proyecto Hyperledger posee probablemente el liderazgo en este aspecto.

El atractivo de la tecnología Blockchain y la situación de constante cambio en la que se encuentra provocan que surjan infinidad de posibles continuaciones al presente TFM. Entre estas, resultaría especialmente interesante tratar en un futuro los siguientes aspectos:

- Una de las dos implementaciones de *Smart Contracts* realizadas podría aprovecharse para la creación de una Dapp completa, con una arquitectura *backend*, una aplicación cliente y los elementos necesarios para la interconexión de estas. Inicialmente, esta labor estaba incluida dentro de uno de los objetivos del proyecto; no obstante, estos fueron definidos antes de adquirir conciencia sobre la complejidad que supone la creación de aplicaciones basadas en Blockchain, por lo que el alcance del proyecto se sobredimensionó ligeramente.
- Resultaría interesante realizar una labor de investigación sobre la aplicabilidad de la Blockchain en otro paradigma tecnológico: la Inteligencia Artificial. Se intuye que de esta "colaboración" se podría extraer una nueva relación simbiótica como la ya obtenida entre la cadena de bloques y la Web Semántica.
- Debido al interés que ha despertado el proceso de *tokenización* para la gestión de activos digitales, resultaría interesante analizar la viabilidad de aplicar este mecanismo para la gestión de, por ejemplo, la subasta eléctrica. La transformación del recurso energético en un *token* auto-gestionable por una infraestructura Blockchain supondría una gran innovación para el sector Energía.

La realidad del mundo actual se caracteriza por unos niveles de interconexión enormes: los volúmenes de datos /comunicaciones distribuidos y compartidos entre millones de usuarios no paran de crecer. Blockchain promete convertirse en la alternativa óptima para la gestión de todos estos recursos. No obstante, la infinidad de escenarios y necesidades actuales provocan que resulte imposible la existencia de una única Blockchain que satisfaga todo este rango de requisitos; es por ello por lo que, con mucha probabilidad, el futuro de este paradigma esté caracterizado por la existencia de multitud de cadenas de bloques y protocolos de comunicación que garanticen una óptima interoperabilidad y vinculación entre las mismas.

Capítulo 9. BIBLIOGRAFÍA

[4] Linares, H. (2017, July 19). Blockchain: Explicación técnica para no-técnicos. Retrieved from https://es.linkedin.com/pulse/blockchain-explicación-técnica-para-no-técnicos-héctor-linaresHerrero

[5] El Economista.es. (n.d.). Dinero: Qué es - Diccionario de Economía. Retrieved from http://www.eleconomista.es/diccionario-de-economia/dinero

[6] Malanov, A. (2017, August 18). Six main disadvantages of Bitcoin and the blockchain. Retrieved from https://www.kaspersky.com/blog/bitcoin-blockchain-issues/18019/.

[7] Stanford Computer Science. (n.d.). Digital Currencies. Retrieved from https://cs.stanford.edu/people/eroberts/cs181/projects/2010-11/DigitalCurrencies/disadvantages/index.html

[8] Panetta, K. (2017, October 12). Top Trends in the Gartner Hype Cycle for Emerging Technologies, 2017. https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017/

[9] Bustos, J. (2017, November 11). Information Aspects of Supply Chain Management. CCI Drexel University.

[10] Marr, B. (2018, March 20). 35 Amazing Real-World Examples Of How Blockchain Is Changing Our World. https://www.forbes.com/sites/bernardmarr/2018/01/22/35-amazing-real-world-examples-of-how-blockchain-is-changing-our-world/#24d957f843b5

[11] *Blockchain for Dummies*. (2017). Hoboken, NJ: John Wiley and sons.

[12] Myler, L. (2017, November 09). Transparent Transactions: How Blockchain Payments Can Make Life Easier For B2B Companies. Retrieved from https://www.forbes.com/sites/larrymyler/2017/11/09/transparent-transactions-how-blockchain-payments-can-make-life-easier-for-b2b-companies/#458f0e1170b5

[13] Wild, S. (n.d.). What is Blockchain? The Technology Explained. Retrieved from https://thehub.smsglobal.com/the-rise-of-cryptocurrencies-and-blockchain

[14] Saglimbeni, A. (2015, 2016). Thesis: The Blockchain as a Financial Market Infrastructure. Politecnico di Milano.

[15] Tapscott, D., & Tapscott, A. (2018). *Blockchain revolution: How the technology behind bitcoin and other cryptocurrencies is changing the world*. United Kingdom: Portfolio Penguin.

[16] Piscini, E., Hyman, G., & Henry, W. (n.d.). *Blockchain: Trust Economy*. In *Deloitte-UK-Tech-Trends-2017*. https://www2.deloitte.com/content/dam/Deloitte/uk/Documents/technology/deloitte-uk-tech-trends-2017-blockchain.pdf

[17] Casey, M. J. (2018, May 18). In blockchain we trust. Retrieved from https://www.technologyreview.com/s/610781/in-blockchain-we-trust/

[18] Monegro, J. (n.d.). Fat Protocols | Union Square Ventures. Retrieved from http://www.usv.com/blog/fat-protocols

[19] E. (2018, May 19). A Comparison Between 5 Major Blockchain Protocols – edChain – Medium. Retrieved from https://medium.com/edchain/a-comparison-between-5-major-blockchain-protocols-b8a6a46f8b1f

[20] Marquer, S. (2017, November 25). World's Biggest Banks are Members of RippleNet. Retrieved from https://ripple.com/insights/the-worlds-biggest-banks-lead-the-blockchain-charge/

[21] Members. (n.d.). Retrieved from https://www.hyperledger.org/members

[22] Martinazzi, S. (2016, 2017). Thesis: The age of FinTech. Providing a liquid and efficient secondary market for security-based crowdfunding with Distributed Ledger Technologies. Politecnico di Milano.

[23] Blockchain Nodes | What Are Nodes and How Do They Work? (n.d.). Retrieved from https://www.worldcryptoindex.com/how-nodes-work/

[24] Brakeville, S., & Perepa, B. (2018, March 18). IBM Blockchain basics: Introduction to distributed ledgers. From

https://www.ibm.com/developerworks/cloud/library/cl-blockchain-basics-intro-bluemix-trs/index.html

[25] World Bank Group. (n.d.). DLT and Blockchain. Retrieved from http://documents.worldbank.org/curated/en/177911513714062215/pdf/122140-WP-PUBLIC-Distributed-Ledger-Technology-and-Blockchain-Fintech-Notes.pdf

[26] I. (2018, March 25). Problems & Costs of Smart Contract Development – iOlite – Medium. Retrieved from https://medium.com/@iolite/problems-costs-of-smart-contract-development-649d88eedd1f

[27] Diedrich, H. (2016). *Ethereum: Blockchains, digital assets, smart contracts, decentralized autonomous organizations*. Erscheinungsort nicht ermittelbar: Wildfire Publishing.

[28] Boaventura, A. (2018, April 12). Demystifying Blockchain and Consensus Mechanisms - Everything You Wanted to Know But Were Never... Retrieved from https://medium.com/oracledevs/demystifying-blockchain-and-consensus-mechanisms-everything-you-wanted-to-know-but-were-never-aabe62145128

[29] Brownworth, A. Blockchain. From https://anders.com/blockchain/blockchain.html

[30] Eremenko, K. (2018, May 03). How does Bitcoin / Blockchain Mining work? – The Startup – Medium. Retrieved from https://medium.com/swlh/how-does-bitcoin-blockchain-mining-work-36db1c5cb55d

[31] Kelleher, J. (2018, July 05). Bitcoin Mining. Retrieved from https://www.investopedia.com/terms/b/bitcoin-mining.asp

[32] Piscini, E., & Kehoe, L. (n.d.). *Blockchain and Cybersecurity. Let's Discuss* (Tech.). Deloitte EMEA Grid Blockchain Lab.

[33] Ganeriwalla, A., Casey, M., Srikhrishna, P., Bender, J., & Gstettner, S. (n.d.). Does Your Supply Chain Need a Blockchain? Retrieved from https://www.bcg.com/publications/2018/does-your-supply-chain-need-blockchain.aspx

[34] SAP Launches Blockchain Supply Chain Initiative. (2018, May 14). Retrieved from https://www.coindesk.com/sap-launches-new-blockchain-supply-chain-initiative/

[35] Nieto, J. M. (n.d.). Comillas presenta el primer nodo universitario 'blockchain'. Retrieved from https://www.comillas.edu/es/noticias-comillas/16882-comillas-presenta-el-primer-nodo-universitario-blockchain

[36] Fresno, B. G. (2018, June 07). BBVA da a sus empleados el control de su formación con un sistema basado en 'tokens' | BBVA. Retrieved from https://www.bbva.com/es/bbva-da-empleados-control-formacion-sistema-basado-tokens/

[37] Meola, A. (2017, September 28). The growing list of applications and use cases of blockchain technology in business & life. Retrieved from https://www.businessinsider.com/blockchain-technology-applications-use-cases-2017-9?IR=T

[38] We have built a digital society and so can you. (n.d.). Retrieved from https://e-estonia.com/

[39] Hyperledger Technical Steering and Marketing Committees (Ed.). (n.d.). *Hyperledger. The Whitepaper*. Linux Foundation.

[40] Ethereum Homestead. From http://www.ethdocs.org/en/latest/frequently-asked-questions/frequently-asked-questions.html#how-will-ethereum-combat-centralisation-of-mining-pools

[41] Orphan, Uncle & Genesis Blocks Explained. (2018, July 29). Retrieved from https://www.mycryptopedia.com/orphan-uncle-genesis-blocks-explained/

[42] Light, J. (2017, September 25). The differences between a hard fork, a soft fork, and a chain split, and what they mean for the... Retrieved from https://medium.com/@lightcoin/the-differences-between-a-hard-fork-a-soft-fork-and-a-chain-split-and-what-they-mean-for-the-769273f358c9

[43] Banafa, A. (n.d.). Three Major Challenges Facing IoT. Retrieved from https://iot.ieee.org/newsletter/march-2017/three-major-challenges-facing-iot.html

[44] Solidity. Retrieved from https://solidity.readthedocs.io/en/v0.4.21/index.html

[45] Rilee, K. (2018, February 20). Understanding Hyperledger Sawtooth - Proof of Elapsed Time. Retrieved from https://medium.com/kokster/understanding-hyperledger-sawtooth-proof-of-elapsed-time-e0c303577ec1

[46] ERC20 Token Standard. (n.d.). Retrieved from https://theethereum.wiki/w/index.php/ERC20_Token_Standard

[47] Castells, P. (n.d.). *La Web Semántica* (Escuela Politécnica Superior. Universidad Autónoma de Madrid, Ed.) [Scholarly project]. Retrieved from http://arantxa.ii.uam.es/~castells/publications/castells-uclm03.pdf

[48] IRI, URI, URL, URN and their differences. (n.d.). Retrieved from http://fusion.cs.uni-jena.de/fusion/blog/2016/11/18/iri-uri-url-urn-and-their-differences/

[49] Pollock, J. T. (2009). *Semantic Web for dummies*. Hoboken, NJ: Wiley Publishing.

[50] Schorlemmer, M. (n.d.). Diez Años construyendo una Web Semántica. From http://www.fgcsic.es/lychnos/es_es/articulos/construyendo_una_web_semantica

[51] Schema.org. (n.d.). Retrieved from https://schema.org/Person

[52] English, M., Auer, S., & Domingue, J. (n.d.). *Blockchain Technologies & The Semantic Web: A Framework for Symbiotic Development* (Unpublished master's thesis). University of Bonn. Retrieved from http://cscubs.cs.uni-bonn.de/2016/proceedings/paper-10.pdf

[53] *A W3C Workshop on Distributed Ledgers on the Web* (Rep.). (n.d.). Retrieved https://www.w3.org/2016/04/blockchain-workshop/report.html

ANEXO A: TARIFICACIÓN DE OPERACIONES[GAS]

Como se ha descrito en la investigación sobre el paradigma de la Blockchain, cada vez que se incorpora un *Smart Contract* a la red, o se realizan llamadas/operaciones sobre uno ya implantado, el nodo creador (o el usuario del contrato) debe pagar una tasa. Ésta se calcula inicialmente en Gas, una unidad que más tarde puede convertirse a Ethereum, la criptomoneda. La siguiente tabla refleja la tarificación correspondiente a cada operación:

Value	Mnemonic	Gas Used	Subset	Removed from stack	Added to stack	Notes	Formula	Formula Notes
0x00	STOP	0	zero	0	0	Halts execution.		
0x01	ADD	3	verylow	2	1	Addition operation		
0x02	MUL	5	low	2	1	Multiplication operation.		
0x03	SUB	3	verylow	2	1	Subtraction operation.		
0x04	DIV	5	low	2	1	Integer division operation.		
0x05	SDIV	5	low	2	1	Signed integer division operation (truncated).		
0x06	MOD	5	low	2	1	Modulo remainder operation		
0x07	SMOD	5	low	2	1	Signed modulo remainder operation.		
0x08	ADDMOD	8	mid	3	1	Modulo addition operation.		
0x09	MULMOD	8	mid	3	1	Modulo multiplication operation.		
0x0a	EXP	FORMULA		2	1	Exponential operation.	(exp == 0) ? 10 :	If exponent is 0, gas used is 10. If exponent is greater than 0, gas used is 10 plus 10 times a factor related to how large the log of the exponent is.
0x0b	SIGNEXTEND	5	low	2	1	Extend length of two's complement signed integer.		
0x10	LT	3	verylow	2	1	Less-than comparison.		
0x11	GT	3	verylow	2	1	Greater-than comparison.		
0x12	SLT	3	verylow	2	1	Signed less-than comparison.		
0x13	SGT	3	verylow	2	1	Signed greater-than comparison.		
0x14	EQ	3	verylow	2	1	Equality comparison.		
0x15	ISZERO	3	verylow	1	1	Simple not operator.		
0x16	AND	3	verylow	2	1	Bitwise AND operation.		
0x17	OR	3	verylow	2	1	Bitwise OR operation		
0x18	XOR	3	verylow	2	1	Bitwise XOR operation.		
0x19	NOT	3	verylow	1	1	Bitwise NOT operation.		
0x1a	BYTE	3	verylow	2	1	Retrieve single byte from word		
0x20	SHA3	FORMULA		2	1	Compute Keccak-256 hash.	30 + 6 * (size of i	30 is the paid for the operation plus 6 paid for each word (rounded up) for the input data.
0x30	ADDRESS	2	base	0	1	Get address of currently executing account.		
0x31	BALANCE	400		1	1	Get balance of the given account.		
0x32	ORIGIN	2	base	0	1	Get execution origination address.		
0x33	CALLER	2	base	0	1	Get caller address.		
0x34	CALLVALUE	2	base	0	1	Get deposited value by the instruction/transaction responsible for this execution.		
0x35	CALLDATALOAD	3	verylow	1	1	Get input data of current environment.		
0x36	CALLDATASIZE	2	base	0	1	Get size of input data in current environment.		
0x37	CALLDATACOPY	FORMULA		3	0	Copy input data in current environment to me	2 + 3 * (number c	2 is paid for the operation plus 3 for each word copied (rounded up).
0x38	CODESIZE	2	base	0	1	Get size of code running in current environment.		
0x39	CODECOPY	FORMULA		3	0	Copy code running in current environment to	2 + 3 * (number c	2 is paid for the operation plus 3 for each word copied (rounded up).
0x3a	GASPRICE	2	base	0	1	Get price of gas in current environment.		
0x3b	EXTCODESIZE	700	extcode	1	1	Get size of an account's code.		
0x3c	EXTCODECOPY	FORMULA		4	0	Copy an account's code to memory.	700 + 3 * (numbe	700 is paid for the operation plus 3 for each word copied (rounded up).
0x40	BLOCKHASH	20		1	1	Get the hash of one of the 256 most recent	complete blocks.	
0x41	COINBASE	2	base	0	1	Get the block's beneficiary address.		
0x42	TIMESTAMP	2	base	0	1	Get the block's timestamp.		
0x43	NUMBER	2	base	0	1	Get the block's number.		
0x44	DIFFICULTY	2	base	0	1	Get the block's difficulty.		
0x45	GASLIMIT	2	base	0	1	Get the block's gas limit.		
0x50	POP	2	base	1	0	Remove item from stack.		
0x51	MLOAD	3	verylow	1	1	Load word from memory.		
0x52	MSTORE	3	verylow	2	0	Save word to memory		
0x53	MSTORE8	3	verylow	2	0	Save byte to memory.		
0x54	SLOAD	200		1	1	Load word from storage		
0x55	SSTORE	FORMULA		1	1	Save word to storage.	((value != 0) && (	20000 is paid when storage value is set to non-zero from zero. 5000 is paid when the storage value's zeroness remains unchanged or is set to zero.
0x56	JUMP	8	mid	1	0	Alter the program counter		
0x57	JUMPI	10	high	2	0	Conditionally alter the program counter.		
0x58	PC	2	base	0	1	Get the value of the program counter prior to the increment corresponding to this instruction.		
0x59	MSIZE	2	base	0	1	Get the size of active memory in bytes.		
0x5a	GAS	2	base	0	1	Get the amount of available gas, including the corresponding reduction for the cost of this instruction.		
0x5b	JUMPDEST	1		0	0	Mark a valid destination for jumps		
0x60 .. 0x7f	PUSH*	3	verylow	0	1	Place * byte item on stack. 0 < * <= 32		
0x80 .. 0x8f	DUP*	3	verylow	*	* + 1	Duplicate *th stack item. 0 < * <= 16		
0x90 .. 0x9f	SWAP*	3	verylow	* + 1	* + 1	Exchange 1st and (* + 1)th stack items.		
0xa0	LOG0	FORMULA		2	0	Append log record with no topics.	375 + 8 * (numbe	375 is paid for operation plus 8 for each byte in data to be logged.
0xa1	LOG1	FORMULA		3	0	Append log record with one topic.	375 + 8 * (numbe	375 is paid for operation plus 8 for each byte in data to be logged plus 375 for the 1 topic to be logged.
0xa2	LOG2	FORMULA		4	0	Append log record with two topics.	375 + 8 * (numbe	375 is paid for operation plus 8 for each byte in data to be logged plus 2 * 375 for the 2 topics to be logged.
0xa3	LOG3	FORMULA		5	0	Append log record with three topics.	375 + 8 * (numbe	375 is paid for operation plus 8 for each byte in data to be logged plus 3 * 375 for the 3 topics to be logged.
0xa4	LOG4	FORMULA		6	0	Append log record with four topics.	375 + 8 * (numbe	375 is paid for operation plus 8 for each byte in data to be logged plus 4 * 375 for the 4 topics to be logged.
0xf0	CREATE	32000		3	1	Create a new account with associated code.		
0xf1	CALL	FORMULA		7	1	Message-call into an account.	Complex -- see yellow paper	Appendix H
0xf2	CALLCODE	FORMULA		7	1	Message-call into this account with an alterna	Complex -- see yellow paper	Appendix H
0xf3	RETURN	0	zero	2	0	Halt execution returning output data.		
0xf4	DELEGATECALL	FORMULA		6	1	Message-call into this account with an alterna	Complex -- see yellow paper	Appendix H
0xfe	INVALID	NA		NA	NA	Designated invalid instruction.		
0xff	SELFDESTRUCT	FORMULA		1	0	Halt execution and register account for later d	5000 + ((create_	5000 for the operation plus 25000 if a new account is also created. A refund of 24000 gas is also added to the refund counter for self-destructing the account.

Figura 40. Tarificación de Operaciones. Red Ethereum

ANEXO B: *SMART CONTRACT. SISTEMA ELECTORAL*

A continuación se proporciona el código correspondiente al *Smart Contract* que gestiona un proceso electoral, contrato con el que se trabajó en la sección 5.1.1.1. de esta memoria. Al visualizar [41], analizar y modificar ligeramente un contrato descrito en Solidity, se observó que este lenguaje comparte propiedades y principios con otros ya manejados durante los últimos seis años: Java, JavaScript, C, etc. No obstante, se han proporcionado una serie de comentarios junto con el código para facilitar la comprensión de éste.

```
pragma solidity ^0.4.16;

contract Voting_v1 { //"constructor"
    // Esta estructura representa a cada uno de los votantes
    struct Votante { //estructura de datos, creación de tipos de datos que solo funciona dentro del contrato
        bool hecho;  // un booleano que indica si la persona ha votado ya o no
        uint voto;   // índice del candidato elegido
        uint peso; // peso que tiene cada votante en las elecciones (si no se quiere que vote: peso 0)
        address delegado; // persona a la que el votante delega su derecho a voto

    }

    // El tipo que se usará para definir a cada uno de los candidatos.
    struct Candidato {
        bytes32 nombre;   //nombre del candidato
        uint recuento; //numero de votos recibidos
    }

    address public moderador;

    // Variable de estado que asigna a cada dirección de la cadena un tipo Votante
    //Cada dirección tiene derecho a votar a priori
    mapping(address => Votante) public votantes;

    // Vector de candidatos
    Candidato[] public candidatos;

    /// Crea la "papeleta" virtual con los nombres de los distintos candidatos (almacenados en un vector)
    function Papeleta(bytes32[] proposalNames) public {
        moderador = msg.sender;
        votantes[moderador].peso = 1;

        // Se añaden los distintos candidatos al vector de propuestas, creando un objecto Candidato
        //temporal y haciendo un push que inserta el candidato con su recuento a 0 en el vector.
        for (uint i = 0; i < proposalNames.length; i++) {
            candidatos.push(Candidato({
                nombre: proposalNames[i],
                recuento: 0
            }));
        }
    }
```

Figura 41. Smart Contract Proceso Electoral. Parte 1/3

```
    // Acción del moderador de dar derecho a voto a una dirección en concreto en estas "elecciones"
    function proporcionarDchoVoto(address votante) public {
        // si el moderador está llamando a esta función, el votante en cuestión aún no ha votado y
        //su peso está a cero, ponerlo a 1 para que al votar el recuento del candidato x se sume 1
        require(
            (msg.sender == moderador) &&
            !votantes[votante].hecho &&
            (votantes[votante].peso == 0)
        );
        votantes[votante].peso = 1;
    }

    /// Delegar el voto a otra dirección "to"
    function delegar(address to) public {
        // apunta al votante que quiere delegar
        Votante storage sender = votantes[msg.sender];
        require(!sender.hecho); //no se puede delegar si ya has votado

        // No te puedes delegar a ti mismo
        require(to != msg.sender);


        while (votantes[to].delegado != address(0)) {
            to = votantes[to].delegado;

            require(to != msg.sender);
        }

        sender.hecho = true;
        sender.delegado = to;
        Votante storage delegado_ = votantes[to];
        if (delegado_.hecho) {
            //si el delegado ya ha votado, directamente el mismo voto
            //se añade al recuento
            candidatos[delegado_.voto].recuento += sender.peso;
        } else {
            //si no ha votado aún, transmitimos el peso del que delega al delegado
            delegado_.peso += sender.peso;
        }
    }
```

Figura 42. Smart Contract Proceso Electoral. Parte 2/3

```
    /// función que permite computar tanto tu voto como el de los que hayan delegado en ti

    function votar(uint candidato) public { //como parámetro de entrada el índice del candidato
        Votante storage sender = votantes[msg.sender];
        require(!sender.hecho); //la dirección debe no haber votado aún
        sender.hecho = true;
        sender.voto = candidato;

        candidatos[candidato].recuento += sender.peso; //el peso en elecciones "justas" será 1 siempre
    }

    // El ganador se descubre checkeando el recuento de votos
    function candidatoGanador() public view
            returns (uint candidatoGanador_)
    {
        uint conteoGanador = 0;
        //recorre el array de candidatos y va actualizando el index del ganador
        for (uint p = 0; p < candidatos.length; p++) {
            if (candidatos[p].recuento > conteoGanador) {
                conteoGanador = candidatos[p].recuento;
                candidatoGanador_ = p;
            }
        }
    }

    // para saber el nombre del ganador, se llama a la función anterior y
    //se accede a su atributo nombre
    function nombreGanador() public view
            returns (bytes32 nombreGanador_)
    {
        nombreGanador_ = candidatos[candidatoGanador()].nombre;
    }
}
```

Figura 43. Smart Contract Proceso Electoral. Parte 3/3

ANEXO C: *PRUEBAS ICAICOIN. METAMASK Y ETHERSCAN*

Como ya se ha comentado en la memoria del proyecto, para verificar el funcionamiento del *token/moneda* creado, se realizaron transferencias del mismo entre dos carteras Metamask. Los pasos a seguir para la realización de este *test* fueron:

1. El otro usuario descarga Metamask, y agrega el *token* ICC

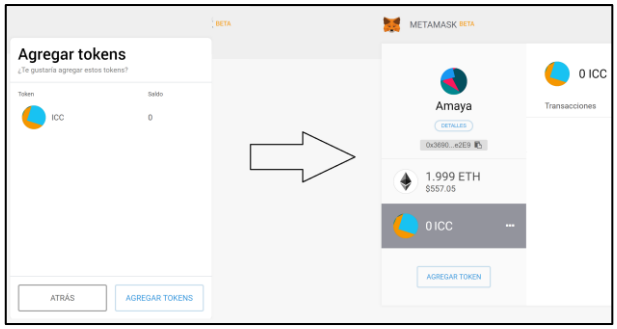


Figura 44. Usuario 2 añade el token ICC

2. El usuario (creador del contrato, y autor del TFM) envía a Amaya 500 ICC:

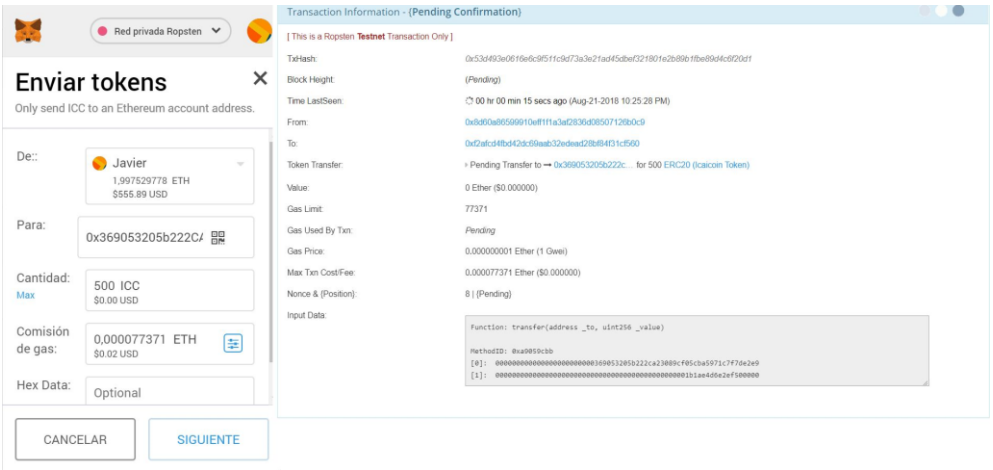


Figura 45. Usuario 1 envía 500 ICC a Usuario 2

3. Compruébese que la dirección a la que se están enviando los 500 ICC coincide con la dirección de Amaya (disponible en la pantalla principal de su perfil de Metamask). Cuando el estado de la transacción pasa de "*pending"* a "*success"* se comprueba que el balance de Amaya ha sido actualizado:

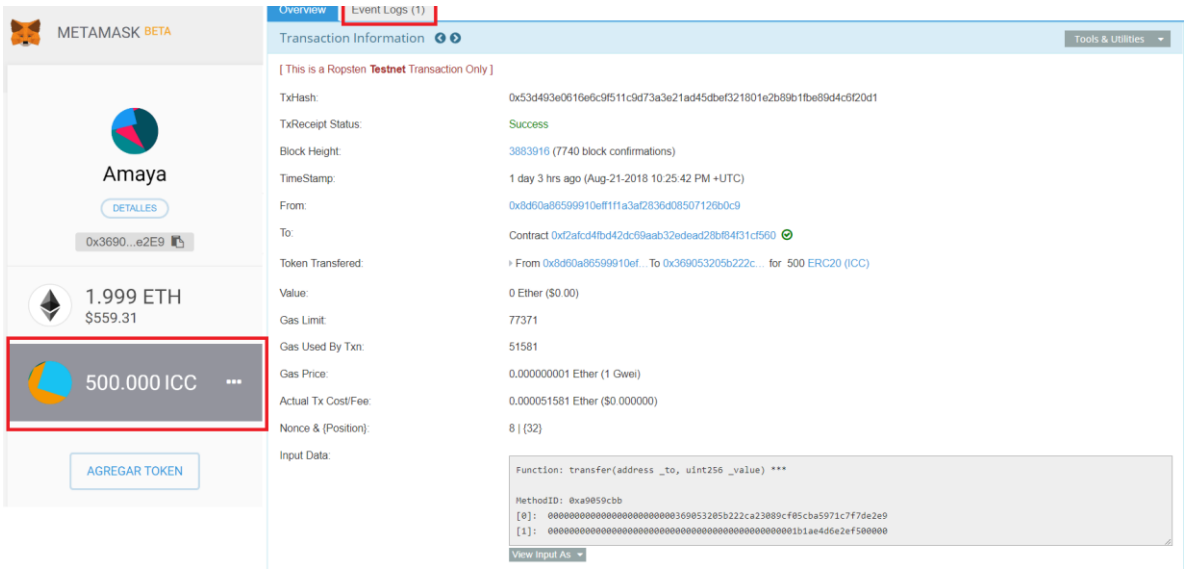


Figura 46. Comprobación del éxito de la transacción

Además, como se aprecia en la captura de la herramienta Etherscan, al realizar esta transacción se ha creado un *log*, como consecuencia de que ha saltado un evento en la ejecución del *Smart Contract* (Transfer, en este caso).

Además de permitir verificar el correcto funcionamiento de la criptomoneda, Etherscan permite rastrear multitud de elementos dentro de la red Ethereum. Por ejemplo, la siguiente captura refleja la información relativa al *Smart Contract* creador de Icaicoin, en la que se reflejan aspectos importantes como:

- El **código del contrato**, traducido a *opcodes* entendidas por la EVM (*Ethereum Virtual Machine*)

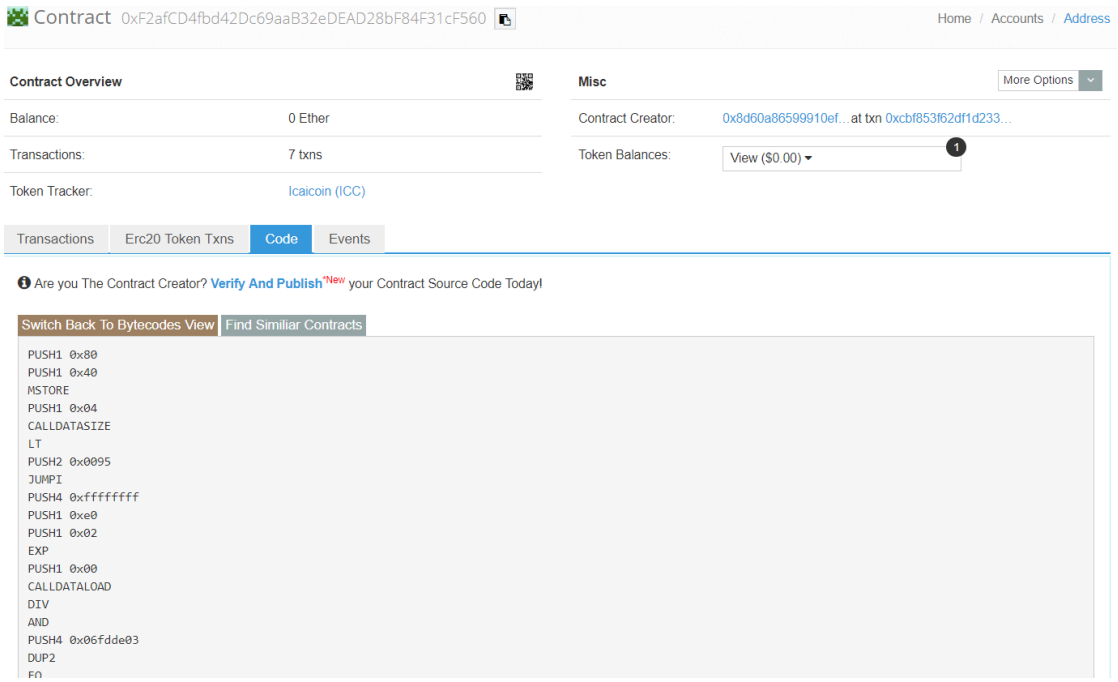


Figura 47. Información Smart Contract. Opcodes

- Los **eventos** que se han disparado desde la creación del contrato, todos de tipo Transfer

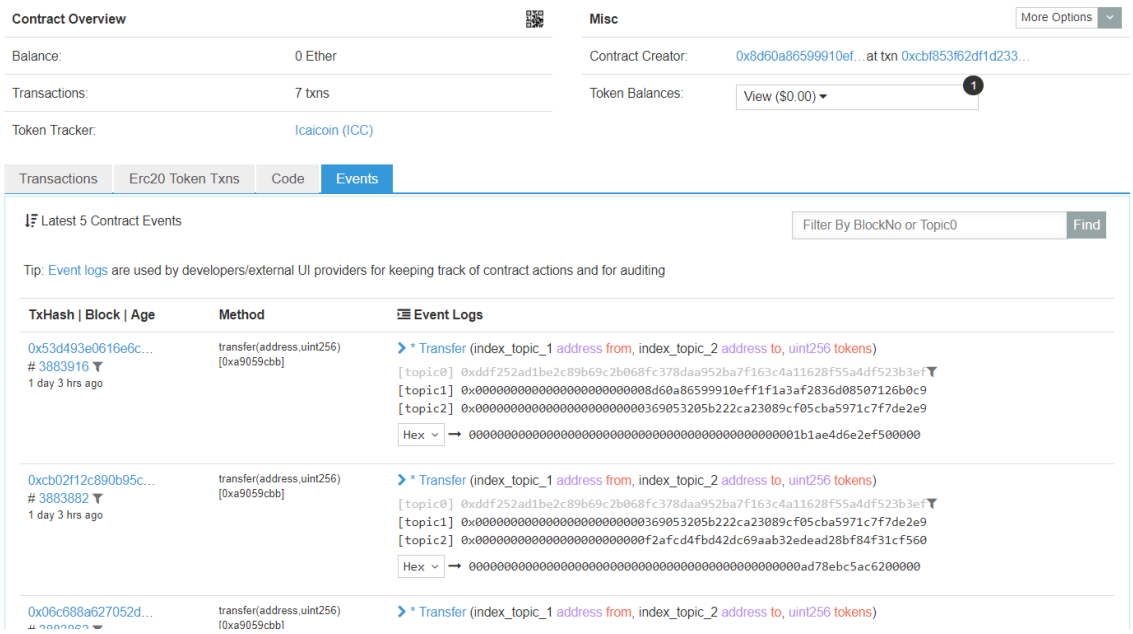


Figura 48. Información Smart Contracts. Historial de Eventos