**UNIVERSIDAD DE OVIEDO**
MÁSTER UNIVERSITARIO EN INGENIERÍA INFORMÁTICA
ÁREA DE ARQUITECTURA DE COMPUTADORES

# Blockchain: Estado del arte, tendencias y retos

TRABAJO FIN DE MÁSTER

D. Luque Lodeiro, Rubén

tutor: Dr. Díaz de Arriba, José Luis
cotutor: Dr. García Martínez, Daniel F.

FECHA: SEPTIEMBRE 2020

TRIBUNAL ASIGNADO:

| | |
|---|---|
| Presidente: | Dr. García Fanjul, José |
| Secretario: | Dr. Entrialgo Castaño, Joaquín |
| Vocal: | Dr. González Bulnes, Francisco |

Este trabajo fin de Máster se realizó en la Escuela Politécnica de Ingeniería de Gijón, Universidad de Oviedo.

TUTOR DEL TRABAJO FIN DE MÁSTER:

Dr. Díaz de Arriba, José Luis

_____

# Resumen

Desde la aparición del primer sistema que utilizaba la tecnología que hoy conocemos como blockchain en 2009 ha existido una rápida evolución hasta llegar al día de hoy. Las características que idealmente debe proporcionar el uso de la tecnología blockchain, como son la inmutabilidad, el no repudio, la preservación de la privacidad, la transparencia o la descentralización, se cumplen pero con matices. Los esfuerzos actuales de los investigadores en este campo se centran en diseñar nuevos mecanismos de preservación de la privacidad, mejorar la escalabilidad del sistema, tanto aumentando el throughput, como disminuyendo la latencia de la red, mejorar o proponer nuevos mecanismos de gobernanza para las redes blockchain que permitan una gestión más segura y justa para todas las partes interesadas o explorar nuevos mecanismos y técnicas para el almacenamiento de datos seguro y eficiente.

Este trabajo fin de Máster aporta una visión general de las bases de la tecnología blockchain al lector, además de realizar una revisión sistemática del estado del arte actual, tanto académico, como industrial. Finalmente se exponen las conclusiones alcanzadas y el trabajo futuro propuesto.

# Abstract

Since the appearance of the first system that used the technology we know today as blockchain in 2009, there has been a rapid evolution until today. The ideal characteristics provided by the use of blockchain technology such as immutability, non-repudiation, preservation of privacy, transparency or decentralization are met, but with nuances. The current efforts of researchers in this field are focused on designing new privacy preservation mechanisms, improving the scalability of the system, both increasing throughput and reducing network latency, improving or proposing new governance mechanisms for blockchain networks that allows a more secure and fair management for all interested parties or to explore new mechanisms and techniques for safe and efficient data storage.

This Master's thesis provides an overview of the foundations of blockchain technology to the reader, in addition to conducting a systematic review of the current state of the art, both academic and industrial. Finally, the reached conclusions and the proposed future work are presented.

# Índice general

**Índice de figuras** v

**Índice de tablas** vi

**1. Introducción y Objetivos** 1
- 1.1. Presentación y justificación del trabajo . . . 1
- 1.2. Definición del objeto de estudio . . . 2
- 1.3. Objetivos . . . 3
- 1.4. Contribuciones . . . 3
- 1.5. Estructura del documento . . . 4

**2. Tecnología Blockchain** 6
- 2.1. Historia . . . 7
- 2.2. Arquitectura de la BC . . . 8
  - 2.2.1. Uso de criptografía . . . 10
  - 2.2.2. Transacciones y bloques . . . 18
  - 2.2.3. Algoritmos de consenso . . . 21
  - 2.2.4. Contratos inteligentes . . . 26
- 2.3. Tipos de redes . . . 28
  - 2.3.1. Públicas . . . 28
  - 2.3.2. Privadas . . . 29
  - 2.3.3. Híbridas . . . 30
- 2.4. Ventajas y desventajas de la Tecnología Blockchain . . . 30
  - 2.4.1. Descentralización . . . 31
  - 2.4.2. Inmutabilidad . . . 31
  - 2.4.3. Escalabilidad . . . 32
  - 2.4.4. Riesgos de seguridad . . . 34
- 2.5. Posibles casos de uso . . . 40
  - 2.5.1. Sanidad . . . 40
  - 2.5.2. Gobierno . . . 43
  - 2.5.3. Vehículos inteligentes . . . 44
  - 2.5.4. Sector financiero . . . 44
  - 2.5.5. Agricultura . . . 45

2.5.6. Otros casos de uso relevantes . . . 46
2.6. Otras tecnologías de registro distribuido . . . 47
2.6.1. Grafos acíclicos dirigidos (DAGs) . . . 48

**3. Estado del arte sistemático** **51**
3.1. Revisión de estados del arte previos . . . 51
3.1.1. Objetivo y visión general . . . 51
3.1.2. Resúmenes de los estudios consultados . . . 54
3.1.3. Conclusiones . . . 70
3.2. Metodología . . . 71
3.3. Análisis de los resultados obtenidos . . . 79
3.3.1. Literatura blanca . . . 80
3.3.2. Literatura gris . . . 83
3.4. Situación actual de los tópicos analizados . . . 85
3.4.1. Privacidad de los datos . . . 85
3.4.2. Almacenamiento de datos . . . 88
3.4.3. Latencia . . . 93
3.4.4. Tasa de transferencia efectiva (throughput) . . . 95
3.4.5. Gobierno . . . 99
3.5. Respondiendo a las preguntas de investigación . . . 103

**4. Conclusiones y trabajo futuro** **107**

**A. Glosario** **110**
Acrónimos . . . 110
Términos y conceptos . . . 114

**Bibliografía** **123**
Literatura blanca . . . 123
Literatura gris . . . 129
Otras fuentes citadas . . . 133
Otras fuentes en línea consultadas . . . 142

# Índice de figuras


2.1. Diagrama de la cronología de las ideas clave de Bitcoin (adaptado de Narayanan y Clark [NC17]) . . . 9
2.2. Ejemplo de función *hash* . . . 11
2.3. Ejemplo de construcción de un árbol Merkle y de la verificación de pertenencia de una transacción al mismo . . . 14
2.4. Usos de la criptografía en la TBC . . . 16
2.5. Proceso de transferencia de fondos simplificado en Bitcoin . . . 18
2.6. Estructura de un bloque de la cadena . . . 19
2.7. Relación entre los bloques de la cadena y sus contenidos . . . 20
2.8. Bifurcación (*fork*) de la cadena . . . 21
2.9. Trilema de la escalabilidad en redes Blockchain . . . 33
2.10. Diagrama de flujo para la decisión de cuándo blockchain es necesario (adaptado de Wust y Gervais [WG18]) . . . 41
2.11. Representación de la red Nano con un *ledger* de 5 cuentas. . . . 49

3.1. Diagrama de flujo del proceso de revisión sistemática de la literatura científica (blanca). . . . 77
3.2. Diagrama de flujo del proceso de revisión sistemática de la literatura gris. 78
3.3. Total de resultados por año tras la eliminación de duplicados y resultados irrelevantes . . . 79
3.4. Literatura blanca: número de resultados por tópico y por año . . . 80
3.5. Literatura gris: número de resultados por tópico . . . 84
3.6. Literatura gris: número de resultados por tópico y por año . . . 84

# Índice de tablas

2.1. Diferentes combinaciones de redes BC en función del ámbito y los permisos 29
2.2. Comparación entre redes de tipo públicas, privadas e híbridas. . . . . 31

3.1. Matriz de análisis de estados del arte previos . . . . . 53
3.2. Tipos de literatura según Garousi *et al.* [GFM19] . . . . . 72
3.3. Tonos de la literatura gris según Garousi *et al.* [GFM19] . . . . . 72
3.4. Criterios de inclusión/exclusión para la literatura blanca. . . . . 74
3.5. Criterios de calidad para la literatura gris. . . . . 76
3.6. Matriz de análisis de la literatura blanca para los términos relacionados con la privacidad de los datos . . . . . 81
3.7. Matriz de análisis de la literatura blanca para los términos relacionados con la escalabilidad . . . . . 82
3.8. Matriz de análisis de la literatura blanca para los términos relacionados con la gobernanza . . . . . 83

Capítulo 1

# Introducción y Objetivos

## 1.1. Presentación y justificación del trabajo

En junio de 2019 arrancó el diseño de este Trabajo Fin de Máster, planteándose dos objetivos fundamentales:

- El primero de ellos era ofrecer una visión general pero rigurosa y detallada de la Tecnología Blockchain (TBC), orientada a un lector que tenga una base sólida en ciencias de la computación y probablemente haya oído hablar de Bitcoin y todas las teorías —más o menos acertadas— que le rodean, pero nunca se haya inmerso en los detalles técnicos de las *Distributed Ledger Technologies*, o tecnologías de registro distribuidos (DLTs).

  Para alcanzar este objetivo se han consultado muchas fuentes generalistas sobre la TBC, como [Wat16; Raj19; Ant17; VS18; Thi19; Vos19a; Pre17; Dob18; BP18; MF18; UKGov16; Ethos18; ZXL19; Bon15; Mik17; Vos19b; MPJ18], entre otras (véanse las referencias completas en el capítulo 2).

  También se ha sacado partido de la experiencia adquirida al formar parte durante casi 2 años de un grupo de investigación sobre TBC de un centro tecnológico. Este grupo acumula un total de 5 años de experiencia en el campo y lidera numerosos proyectos tanto internos como con empresas que desean adentrarse en la TBC y realizar un piloto para posteriormente ponerlo en producción [véanse por ejemplo CTIC18; CTIC17; Mei19a]. También cabe destacar las numerosas publicaciones del grupo en forma de opinión critica sobre aspectos que atañen a la TBC, [como por ejemplo ML20; MAB17; Mei18; Mei19b]. Por último, destacar también la experiencia con diferentes redes Blockchain, como Ethereum o EOS, formando parte de los *block producers* de una red hermana de esta última, [Teloscope19].

  El resultado es el Capítulo 2, que presenta una panorámica de todas las tecnologías involucradas en el Blockchain, sus campos de aplicación, sus puntos débiles, etc.

- El segundo objetivo, más importante y que da título al trabajo, era realizar una revisión sistemática de la literatura científica ("estado del arte") sobre las TBCs, con objeto de encontrar líneas de investigación prometedoras para una futura Tesis doctoral. Este segundo objetivo se puede descomponer en dos sub-objetivos:

- Ya que, naturalmente, ésta no iba a ser la primera revisión sistemática en este tópico, es necesario estudiar primero hasta dónde llegaron las revisiones o estados del arte previos.

  Por tanto se comenzó por buscar las revisiones o *surveys* más recientes en el área de TBC, tratando de encontrar las que cubrieran un mayor número de tópicos y áreas. Esta revisión de estados del arte previos dio lugar al apartado 3.1 (pág. 51) del presente trabajo, donde se pueden consultar más detalles y referencias de todos los estudios consultados.

  Como resultado de este estudio se identifican una serie de tópicos (véase sección 3.1.3) que según las revisiones consultadas eran retos a resolver, y por tanto líneas de investigación prometedoras, en 2019.

- El siguiente objetivo sería actualizar el estado del arte en los tópicos resultantes del estudio previo, es decir, encontrar y revisar los artículos sobre esos tópicos que hubieran ido apareciendo desde que dichas revisiones fueron publicadas.

  Para esta fase se utilizó la metodología de revisión sistemática explicada en Butijn *et al.* [BTH19], adaptada a nuestro caso (véase sección 3.2). Se realizaron las búsquedas, recopilación de información, selección y lectura de los resultados más relevantes y se clasificaron según diferentes criterios, dando lugar al resto del capítulo 3 (a partir de la pág. 79).

Una vez realizada la revisión y analizados los resultados obtenidos, hemos extraído varias conclusiones (ver capítulo 4) que podrán servir de punto de partida para una futura tesis doctoral.

## 1.2. Definición del objeto de estudio

A pesar de la popularidad de la tecnología blockchain (de aquí en adelante, TBC) no existe un consenso en la definición técnica de lo que es (y no es) esta tecnología. En numerosas ocasiones se vincula únicamente a la red Bitcoin, pero es necesario resaltar que Bitcoin solo es el primer y más conocido caso de uso de la TBC, pero ésta no se restringe a este único caso. La mayoría de elementos que componen lo que hoy en día llamamos blockchain (de aquí en adelante, BC) ya existían años antes de construirse la red Bitcoin.

Una definición acertada de la TBC podría ser la que sugiere Werbach [Wer18]:

> *"Un sistema de base de datos distribuido manejado por una red P2P de computadores que proporcionan un registro compartido pero preciso".*

Después de Bitcoin que fue bautizada posteriormente como la "primera generación" blockchain o Blockchain 1.0, la siguiente red blockchain que supuso un avance en la forma y las posibilidades de esta nueva tecnología fue Ethereum, también conocida como blockchain de "segunda generación" o Blockchain 2.0. El aporte de esta red fueron

los contratos inteligentes que básicamente se definen como un programa codificado en función de diversas restricciones para plasmar un acuerdo o contrato entre diferentes partes interesadas que no poseen confianza entre ellas. El contrato inteligente podrá almacenar datos, recibir entradas y generar salidas en función de las reglas predefinidas en el momento de su creación y que no pueden ser modificadas a posteriori. Además, el programador debe ser muy cuidadoso con su código y tener en cuenta diferentes aspectos, ya que el mínimo fallo en un contrato inteligente podría derivar en pérdidas millonarias.

En el capitulo 2 explicaremos en profundidad todos los detalles de la TBC, describiendo su arquitectura, cuál es la unidad mínima de información almacenada en una red BC, cómo se transmiten los datos, o cómo se garantizan las características que aportan este tipo de redes como son la inmutabilidad (sección 2.4.2) o la descentralización(sección 2.4.1). También se describirán los diferentes tipos de redes (sección 2.3), así como los casos de uso (sección 2.5) más comunes o que mayores ventajas aportan.

## 1.3. Objetivos

Como ya se ha avanzado en la introducción (sección 1.1) los dos objetivos clave de este trabajo son:

- Ofrecer una visión técnica de la TBC comenzando por como surgen este tipo de sistemas distribuidos, cuáles son sus principales características y qué tipos de redes blockchain existen. Como parte de este objetivo se revisarán también los ataques más comunes que sufre este tipo de sistemas y los algoritmos criptográficos y de consenso que utiliza, lo que convierte a la TBC en lo que es hoy en día.

- Realizar una revisión sistemática de la literatura científica con objeto de capturar la tendencia actual de la investigación en este campo, y los retos a los que se enfrenta.

Como conclusión se presentarán algunas de las direcciones de investigación más prometedoras que se pueden tomar en esta área de estudio, y las posibles líneas de trabajo futuro.

## 1.4. Contribuciones

Las contribuciones más importantes de este trabajo se enumeran a continuación:

- Se aporta una **revisión sistemática del estado de la literatura**, en la que incluimos tanto la literatura científica o blanca como la literatura gris, analizando todo lo publicado hasta la fecha y centrado en las brechas en la investigación (*research gaps*) y retos que se han identificado después de realizar una revisión de los estados del arte más recientes en este campo (sección 3.1.2).

- Nuestro trabajo estudia la tecnología *blockchain* desde un punto de vista genérico, y no centrado exclusivamente en una red concreta como Bitcoin, como es el caso otras revisiones previas, como la de Yli-Huumo *et al.* [Yli+16], o la revisión de Tschorsch y Scheuermann [TS16] en la cual se revisa la TBC desde el punto de vista de las criptomonedas, pero no desde una visión más global, o la de Casino *et al.* [CDP19] en la que se proporciona una revisión de la literatura con el objetivo de constituir una base para el diseño de aplicaciones multipropósito. Pero ninguna de ellas se centra en actualizar el estado de los tópicos más importantes que presentan un reto a la evolución de la TBC de forma genérica, sin centrarse en un campo de aplicación o red BC concreta. Este TFM aporta **una visión lo más genérica posible de la TBC**, recopilando, identificando y exponiendo los **retos más importantes** en el momento de realizar este trabajo, junto con su estado actual y las posibles líneas futuras a seguir.

- Se han revisado numerosos estudios y estados del arte previos con el objetivo de definir cuál era el estado del arte y las tendencias más prometedoras a finales de 2018, coincidiendo con el inicio conceptual de este trabajo fin de Máster. Como resultado, **se identifican y revisan todos los estudios y estados del arte previos (sección 3.1.2)** que encajan dentro de nuestro objetivo, las tendencias y los retos de la TBC de forma general.

- En la sección 3.5 **damos respuesta a las preguntas de investigación planteadas al inicio de este TFM** aportando los últimos avances en la investigación centrados en los tópicos seleccionados, los retos a los que se enfrentan tanto la academia, como la industria, y por último, las tendencias que extraemos después de realizar nuestra revisión.

- Se identifican y exponen las **líneas de investigación que pueden resultar prometedoras e interesantes** dentro de los tópicos seleccionados para realizar nuestra revisión sistemática, habilitando un camino a seguir por los investigadores interesados en cualquiera de estos tópicos.

- Se proporciona en este documento una referencia completa de las tecnologías *blockchain*, explicando sus aspectos más importantes y completando el documento con un **glosario de términos con referencias cruzadas al texto** que puede ser útil al lector para navegar entre la gran cantidad de acrónimos y conceptos que rodean a esta tecnología.

## 1.5. Estructura del documento

Este documento está dividido en 4 capítulos, un glosario y la bibliografía.

El primer capítulo consiste en una introducción a la temática de este trabajo fin de Máster, además de presentar los objetivos planteados, las contribuciones y la estructura del documento.

En el segundo capítulo se ofrece una visión general de las tecnologías de registro distribuido (DLTs) en las cuales se incluye la tecnología blockchain (TBC) y se detalla en profundidad aspectos claves de dicha tecnología.

El tercer capítulo es el estado del arte propiamente dicho. En él se comienza con un estudio de los estados del arte previos, se describe la metodología aplicada para realizar la revisión sistemática acometida en este trabajo, así como el enfoque, y la estrategia de búsqueda utilizados, se analizan los datos obtenidos mediante las búsquedas realizadas, se presenta la situación actual de cada uno de los tópicos estudiados y se responde a las preguntas de investigación planteadas previamente en el mismo capítulo.

Finalmente en el cuarto capítulo encontramos las conclusiones y las posibles líneas futuras de trabajo.

Además, en los anexos se proporciona un glosario de acrónimos y términos y la bibliografía citada desde los restantes capítulos.

Capítulo 2

# Tecnología Blockchain

El principal objetivo de la tecnología blockchain es eliminar la necesidad de confianza entre dos partes que desean realizar una transacción, además de eliminar a las terceras partes centralizadas que verifiquen dicha transacción. A la vez es necesario mantener un registro inmutable de todas las transacciones realizadas con el fin de poder verificar si las transacciones son válidas (por ejemplo, si el emisor tiene fondos cuando se trata de transacciones monetarias).

La Tecnología Blockchain (TBC) es una tecnología de registro distribuido en la cual se almacena la secuencia de transacciones en una cadena de bloques, y cada nodo del sistema tiene una copia completa de dicha cadena. Cada bloque de la cadena contiene transacciones válidas y verificadas por los nodos mineros. Para crear una nueva transacción, el usuario debe tener una *wallet* (cartera) y conocer tanto su propia dirección pública (creada a partir de su clave pública) como la dirección pública del destinatario. Todos los bloques se encuentran encadenados entre sí a través del cálculo de su *hash*, lo que asegura la ausencia de modificaciones o inmutabilidad y el no-repudio. Cada red BC puede utilizar un algoritmo de consenso (sección 2.2.3) diferente, el cual define determinadas reglas que deben cumplirse para que se llegue a un acuerdo entre todos los nodos validadores o mineros. Los algoritmos de consenso más conocidos son PoW, PoS o dPoS, que detallaremos a lo largo del documento.

Por simplicidad, explicaremos cómo funciona el proceso de consenso basándonos en el algoritmo utilizado por Bitcoin, PoW. En otros algoritmos como PoS, el proceso cambia y los mineros ya no necesitan poseer una alta capacidad computacional, sino que la probabilidad de crear un nuevo bloque varía en función del número de *tokens* de la red que posea el nodo. A mayor número, mayor probabilidad de minar el siguiente bloque. Los usuarios de la red generan transacciones que deben añadirse a un bloque, el cual a su vez debe añadirse a la cadena y ser encadenado por al menos 6 bloques posteriores para considerarse esa transacción válida e "inmutable". Cuando un nodo minero o validador recibe una nueva transacción, deberá comprobar que el emisor y el receptor poseen direcciones públicas válidas y que el emisor tiene fondos suficientes para realizar la transacción. En ese caso añadirá la transacción a un bloque que necesita "llenar" y una vez completado el bloque será sometido a la aprobación del resto de mineros de la red. La aprobación dependerá de que cumpla ciertas condiciones dependientes de la red. En general, el contenido del bloque deberá coincidir con el contenido de la copia

que del mismo bloque posee cada minero, y la prueba de trabajo debe ser válida (en redes que utilicen PoW).

El nodo que quiera añadir un nuevo bloque a la red, también conocido como minero, deberá resolver ciertos cálculos computacionales complejos que, gracias a ciertas características de las funciones *hash* utilizadas, sólo pueden resolverse por fuerza bruta. En concreto, en el algoritmo PoW se debe encontrar una secuencia de bits llamada *nonce* tal que al ser concatenada al bloque y calcular el *hash* de todo el bloque (incluyendo el *nonce*), produzca como resultado un *hash* con un determinado número de ceros al inicio. A mayor número de ceros requeridos, mayor dificultad para calcular dicho *nonce*, y viceversa. Esto permite regular la dificultad y por tanto el tiempo necesario para añadir nuevos bloques a la cadena.

Internamente, como veremos más adelante, cada bloque almacena sus transacciones haciendo uso de árboles Merkle (subsección 2.2.1.2), cuya ventaja es que se puede verificar la pertenencia o no de una transacción al bloque, sin requerir acceso al bloque completo.

## 2.1. Historia

Aunque el término blockchain se popularizó gracias a la publicación del *white paper* *"Bitcoin: A Peer to Peer Electronic Cash System"* [Nak09] —origen del Bitcoin, que es una de las aplicaciones más extendidas de las que tiene la blockchain— dicho artículo no es más que una recopilación de ideas publicadas por otros autores en el pasado, pero dándole un sentido, una forma, y lo que es aún más importante, proponiendo una aplicación relevante y consiguiendo a su vez una adopción masiva. Antes de Bitcoin ya se habían publicado modelos de dinero electrónico, como por ejemplo, *B-money* de Dai [Dai98], el cual nunca llegó a implementarse, pero que guarda gran relación con Bitcoin. Al parecer, Nakamoto leyó el artículo de Wei Dai después de haber planteado su *white paper*, y al ser ideas muy similares, se lo envió a Dai antes de publicarlo para que lo revisara y lo añadió como bibliografía en su *white paper*. Algo parecido ocurrió con *Bit Gold*, una propuesta de Szabo [Sza05] en 2005.

El problema que intenta resolver la tecnología BC ya era bien conocido desde las décadas de los años 80 y 90. En el artículo *"The Part-Time Parliament"*, publicado por Lamport [Lam98] se describe el problema de cómo lograr el consenso en una red de computadores donde uno o varios computadores o incluso la propia red pueden ser no confiables (comportarse de forma errática o maliciosa, comportamientos englobados en el término *fallos bizantinos*), y se propone un algoritmo para resolverlo, pero éste requiere de un gran número de mensajes intercambiados, que crece exponencialmente conforme aumenta el número de nodos no confiables. En otro artículo titulado *"How to Time-Stamp a Digital Document"*, publicado en 1991 por Haber y Stornetta [HS91], se planteaba un mecanismo para asegurar la no alteración de documentos digitales, el cual se basaba en bloques encadenados por el *hash* unidireccional de su estampa de tiempo (*timestamp*) calculada en el momento de crear el documento.

Más recientemente, se publica *HashCash*, un sistema propuesto por Back [Bac02] que consiste en un mecanismo para minimizar la recepción de grandes cantidades de correos electrónicos no deseados, haciendo uso de funciones hash. Este sistema se podría considerar una primera versión del algoritmo de prueba de trabajo (*Proof-Of-Work*) utilizado y popularizado por Nakamoto [Nak09] años después. Bitcoin se considera a día de hoy como una blockchain de primera generación o Blockchain 1.0.

En Narayanan y Clark [NC17] los autores muestran un diagrama (Figura 2.1) de la cronología de las ideas clave de Bitcoin (también aplicable de forma genérica a BC) como por ejemplo, sellados de tiempo enlazados, *Proof of Work* (PoW), tolerancia a fallos bizantinos, etc.

Más adelante, en 2013 Buterin *et al.* [But+] publicaron el *white paper* de la red Ethereum que se lanzaría al público en 2015, que inaugura la nueva generación de *blockchain*, las Blockchain 2.0 caracterizada por la posibilidad de ejecutar contratos inteligentes (sección 2.2.4) aumentando así los posibles casos de uso de la tecnología *blockchain* y permitiendo la creación de *dApps*. Pero esta generación presentaba problemas heredados de las *blockchain* como son la limitada escalabilidad (sección 2.4.3) o la escasa velocidad de transacción.

Es por ello, que nacen más adelante las Blockchain 3.0 con el principal objetivo de mejorar esos dos aspectos. Algunos ejemplos de este tipo de redes son EOS, Nano o Cardano (Hoskinson [Hos17]), que además de implementar mecanismos para mejorar la escalabilidad y los problemas de velocidad de transacciones de la generación anterior, permiten un "auto-gobierno" de la red, interoperabilidad entre redes Blockchain (BC) o posibilidad de utilizar varias redes BC en paralelo.

En la actualidad, las tendencias en la tecnología BC se centran en su integración con redes *Internet of Things*, o Internet de las cosas (IoT) o *Industrial Internet of Things*, o Internet Industrial de las cosas (IIoT), con sistemas de inteligencia artificial, en la creación de estándares que asienten las bases de procesos comunes a muchas redes BC o en la mejora de la interoperabilidad entre redes. Según se desprende de las conclusiones de este Trabajo fin de Máster, en la actualidad las tendencias incluyen la mejora de la escalabilidad (ya sea mejorando la latencia o el *throughput*), el almacenamiento y la privacidad de los datos o el gobierno de las redes BC.

## 2.2. Arquitectura de la BC

La cadena de bloques, o BC, debe almacenar de forma distribuida un conjunto de transacciones, que son la información mínima que se almacena en la misma. Por eficiencia, las transacciones se agrupan en bloques y son estos bloques los que se añaden a la cadena. La forma de añadirlos implica primitivas criptográficas como *hashes*, árboles Merkle, etc, que veremos en las secciónes siguientes. Ya que la cadena está distribuida, existen múltiples copias de la misma, por lo que son necesarios algoritmos de consenso (sección 2.2.3) que aseguren que todas las copias contienen la misma información, además de orquestar el proceso de añadir o no un nuevo bloque a la cadena. El primer objetivo de la blockchain fue almacenar transacciones económicas con criptomonedas,

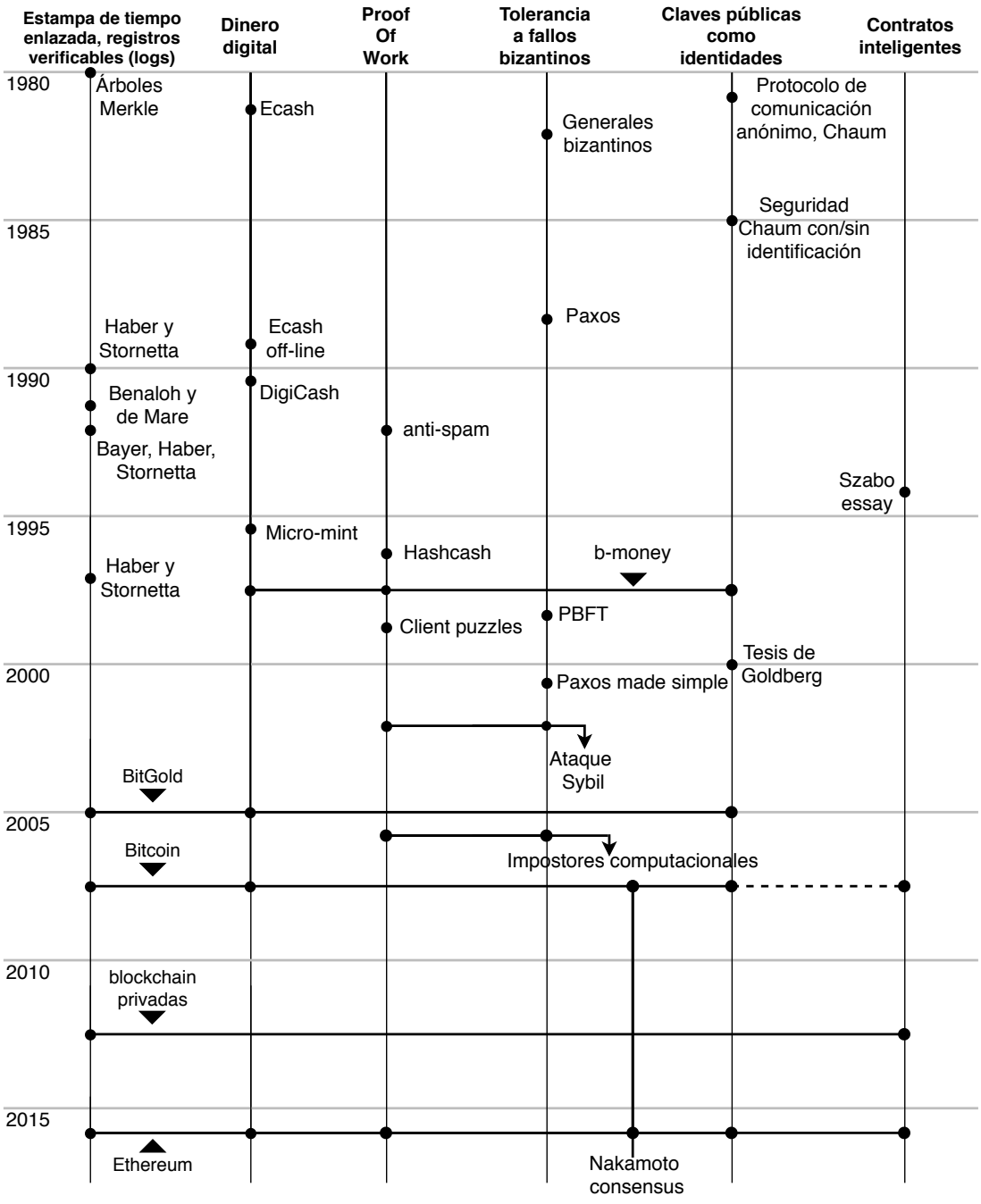


**Figura 2.1.-** Diagrama de la cronología de las ideas clave de Bitcoin (adaptado de Narayanan y Clark [NC17])

lo que hace necesario identificar las "cuentas" o direcciones de los usuarios, que deben ser únicas (dos usuarios no pueden tener la misma dirección), pero sin que sea necesario identificar a los usuarios en sí, lo cual se logra de nuevo mediante primitivas criptográficas. Posteriormente la utilidad de la blockchain se amplía al poder almacenar no sólo transacciones económicas, sino código que pueda ser ejecutado de forma distribuida, dando lugar a contratos inteligentes (Blockchain 2.0) y a *dApps* (Blockchain 3.0). Desde su creación, uno de los apartados más importantes de las redes BC es el gobierno. El gobierno de una red BC se refiere todos los procesos de consenso y toma de decisiones que afectan de manera directa o indirecta a todos los usuarios de la red, y pueden tener lugar dentro de la cadena (*on-chain*) o fuera de ella (*off-chain*). En esta sección estudiaremos los componentes y las tecnologías que hacen todo esto posible.

### 2.2.1. Uso de criptografía

La TBC hace uso de mecanismos y primitivas criptográficas junto con conceptos de registros distribuidos. Los componentes principales serían funciones *hash*, transacciones, criptografía de clave asimétrica, direcciones, registros, bloques y el mecanismo que crea los enlaces entre bloques (la cadena). Varias de estas primitivas criptográficas se describen en esta sección. Su uso dentro de la tecnología BC, sería el siguiente:

- **Creación de la pareja de claves público/privada** mediante el algoritmo de firma digital ECDSA o similar.
- **Generación de la dirección pública** a partir de la aplicación de una función *hash* a la clave pública generada anteriormente. A su vez dicha dirección pública se codifica en base58 (en Bitcoin).
- **Generación de *hash* únicos** para identificar transacciones y bloques (en Bitcoin por ejemplo, se utiliza *SHA-256*). Dichos *hashes* sirven además para encadenar los bloques.
- Las transacciones se **firman** por el emisor utilizando algoritmos de firma digital como ECDSA o edDSA.

#### 2.2.1.1. Funciones Hash

Una función *hash* es un algoritmo que se aplica a un bloque de bytes de longitud arbitraria y produce como resultado una nueva cadena de bytes de longitud fija. Dada una entrada $E$ siempre se obtendrá la misma salida $S$, y además la función no es reversible (es decir, no es posible reconstruir la entrada $E$ a partir de la salida $S$). Esto es útil, por ejemplo, para verificar que los datos no han sufrido ninguna modificación, ya que cualquier cambio en los datos, por mínimo que sea, derivará en una cadena de salida radicalmente distinta.

Existen dos tipos de funciones resumen o *hash*: si utilizan una clave adicional para generar el resultado, son de tipo MAC (*Message Authentication Codes*) y una de las

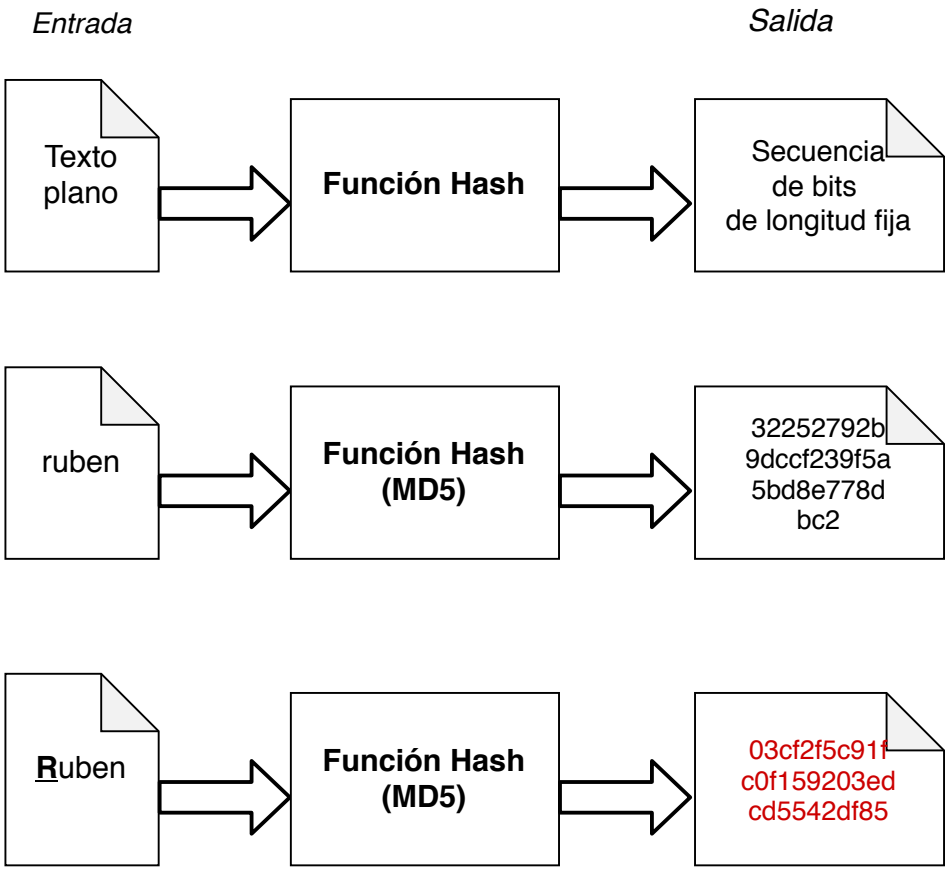


**Figura 2.2.-** Ejemplo de función *hash*

principales ventajas es que garantiza el origen del mensaje, ya que solo quien esté en posesión de la clave podrá haber computado el *hash* resultante. El segundo tipo son las llamadas MDC (*Modification Detection Codes*) que se calculan directamente sobre el mensaje de entrada y su único objetivo es garantizar que el mensaje no ha sido alterado.

La figura 2.2 presenta el funcionamiento de una *función hash MD5*. En la primera fila se muestra de forma genérica, en la segunda fila se aplica como entrada la cadena `ruben`, y da como resultado un *hash*, y por último, en la última fila le aplicamos como entrada la cadena `Ruben`, donde solo hemos modificado un carácter y el *hash* resultante es completamente distinto.

Las funciones *hash* que se usan en criptografía deben tener las siguientes propiedades:

1. **Resistencia a la preimagen:** Esta propiedad garantiza que las funciones *hash* son unidireccionales, es decir, no es computacionalmente factible obtener el valor de entrada haciendo cambios en el valor de salida. Es decir, conocida la salida $S$ no podríamos encontrar, salvo por fuerza bruta, una entrada $E$ tal que $\text{Hash}(E) = S$.

2. **Resistencia a colisiones:** No es computacionalmente factible encontrar dos entradas diferentes que como resultado de aplicarles la función *hash* se obtenga la misma salida. Es decir, no es posible encontrar dos valores de entrada $E$ y $E'$ distintos tales que, $\text{Hash}(E) = \text{Hash}(E')$, salvo por fuerza bruta.

3. **Resistencia a la segunda preimagen:** Debido al diseño de las funciones *hash*, no es computacionalmente factible encontrar un valor de entrada con el cual se obtenga el mismo resultado que con otro valor de entrada conocido. Es decir, dado un valor de entrada $e$, no es posible encontrar otro valor de entrada diferente $e'$, tal que, $\text{Hash}(e) = \text{Hash}(e')$, salvo por fuerza bruta. Se podría decir que es un tipo de resistencia a colisiones, con la diferencia de que en este caso conocemos un valor de entrada y pretendemos encontrar otro con el mismo *hash*.

4. **Resistencia a la casi colisión:** Debe ser difícil encontrar dos entradas diferentes $E$ y $E'$, tales que, el resultado de aplicarles la función *hash* a cada entrada se diferencie únicamente en unos pocos bits.

Además de cumplir las propiedades anteriores, las funciones *hash* utilizadas en la TBC deben cumplir una propiedad adicional:

5. ***Puzzle friendly*** [véase StackExchange20]. Una función *hash* $H$ es *puzzle friendly* si para cada posible valor de salida $S$ de $N$ bits, y dado un valor de entrada $k$ obtenido de una distribución con alta entropía, entonces no es posible encontrar un valor $x$ tal que $H(k||x) = y$, en un tiempo significativamente menor que $2^N$. Esto básicamente significa que si se busca una entrada que produzca un *hash* dado, por el método de concatenar a dicha entrada un grupo de bits $k$, no será posible encontrar el valor apropiado de $k$ salvo por fuerza bruta. Esta propiedad es la clave del algoritmo de consenso PoW (que se verá con más detalle en la sección 2.2.3.1 en la página 22).

Los cometidos de las funciones *hash* en la tecnología BC son:

- **Creación de identificadores únicos:** Gracias a las propiedades de las funciones *hash* es posible crear identificadores únicos.
- **Generación de direcciones:** Muchas redes BC utilizan direcciones (cadenas alfanuméricas) que se generan a partir de la clave pública del usuario y algún dato adicional que varía el función de la red. Esto se combina con una función de *hash*, dando como resultado la dirección. La forma de generar las direcciones varía en cada red BC.
- **Protección de la integridad de cada bloque** gracias a que la propia cabecera del bloque contiene un *hash* que resume los datos del bloque.
- **Protección de la integridad de la cadena:** El *hash* de la cabecera del bloque actual se incluirá en la cabecera del siguiente bloque, donde asegurará los datos de la cabecera del bloque actual. Un cambio en el bloque actual será detectado en el bloque siguiente al no coincidir el *hash*.

Las primitivas criptográficas más utilizadas en las 30 redes BC con mayor volumen de transacciones diario, según el artículo publicado por Wang *et al.* [Wan+18], son:

- **Funciones resumen (*hash*):** SHA256 y RIPEMD160, respectivamente.
- **Firma digital:** ECDSA y edDSA, respectivamente.

#### 2.2.1.2. Árboles Merkle

Un árbol Merkle es una estructura de datos propuesta por Merkle [Mer88] que combina árboles binarios con funciones resumen o *hash* y se usa en la BC para almacenar de forma eficiente las transacciones que componen cada bloque (véase figura 2.6). Al ser binario, los nodos hoja siempre serán pares. Si se da el caso en el que hay un número impar de transacciones, se duplicará el último *hash* para conseguir un número par de hojas *(árbol equilibrado)*. La complejidad de comprobar si un elemento pertenece al árbol es $O(log(n))$ de media.

Su utilidad en BC consiste en verificar de manera eficiente la integridad de un conjunto extenso de transacciones así como comprobar si una transacción pertenece a un conjunto dado de transacciones (bloque). El árbol Merkle se construye de manera ascendente y recursiva. Consideremos por ejemplo un escenario en el que tengamos 4 transacciones, $A, B, C$ y $D$, que constituyen las hojas del árbol (véase fig. 2.3). El árbol no almacena todos los datos contenidos en cada transacción, sino que sólo almacena el resumen (*hash*) de sus datos: $H_A, H_B, H_C$ y $H_D$. Se calcula entonces de forma recursiva el *hash* de la suma (concatenación) de los *hashes* de cada par de nodos, lo que dará como resultado el valor del nodo padre de ese par de nodos, y este proceso se repite hasta que solo quede un nodo, que sería la raíz del árbol, y en este caso, la raíz merkle (*Merkle root*).

La primera iteración daría para el nodo padre de las hojas $A$ y $B$ el valor $H(H_A + H_B)$ y para el nodo padre de las hojas $C$ y $D$ el valor $H(H_C + H_D)$. En la siguiente iteración (y última en este caso), se realizaría el *hash* de la concatenación de estos nodos padre para obtener el resultado $H(\text{Nodo}_{A-B} + \text{Nodo}_{C-D})$, que sería la raíz Merkle, ya que no tenemos más pares de nodos.

Para la comprobación de pertenencia nos basaremos en el mismo ejemplo de la figura 2.3. Supongamos que queremos comprobar si la transacción $C$ pertenece a este árbol Merkle. Para ello es necesario conocer información adicional, que nos debe ser suministrada (pero como veremos esta información puede darla el propietario del árbol sin necesidad de revelar los contenidos de las transacciones almacenadas en el mismo). La información necesaria será: el *hash* de su nodo *hermano* (el que comparte padre), así como los *hashes* de todos los nodos hermanos de los nodos ancestrales (es decir, el hermano del padre, el hermano del abuelo, etc.) hasta llegar a la raíz del árbol. En nuestro ejemplo, para validar si $C$ está en el bloque, debemos conocer $H_D$ y el valor del $\text{Nodo}_{A-B}$ (ambos rodeados de una línea discontinua roja en la figura). La validación se realizaría como sigue: a partir del valor $C$ que queremos validar, calculamos $H_C$. A partir de $H_C$ y $H_D$ (que sería un dato), obtenemos $H(H_C + H_D)$, que será el valor del $\text{Nodo}_{C-D}$. Con este valor más el valor de $\text{Nodo}_{A-B}$ (que sería otro dato) se puede

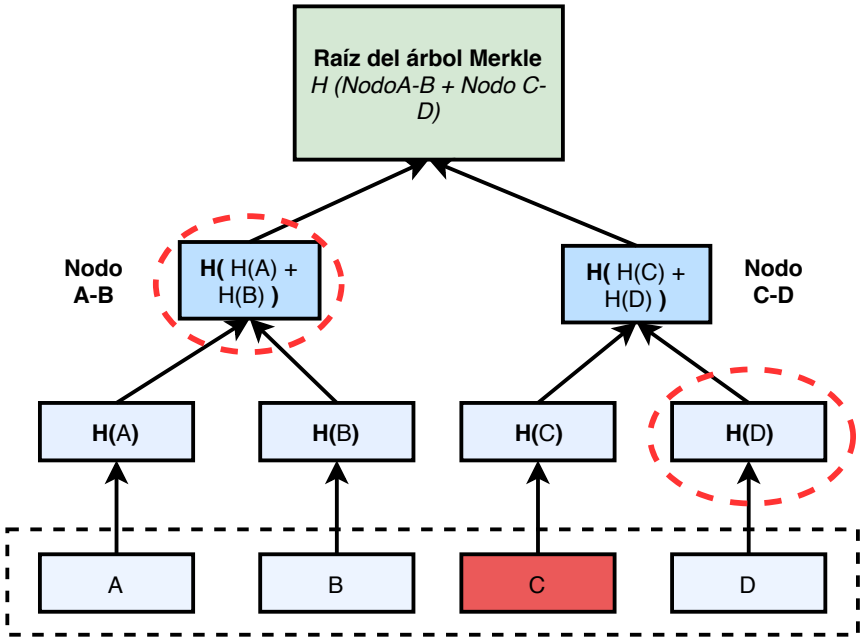


**Figura 2.3.-** Ejemplo de construcción de un árbol Merkle y de la verificación de pertenencia de una transacción al mismo

calcular el *hash* de dicha suma, lo que nos daría el valor del nodo raíz Merkle. Por último, solo debemos comprobar si la raíz Merkle obtenida es igual a la raíz Merkle contenida en el bloque.

Obsérvese que aunque la validación requiere que ciertos *hashes* del árbol nos sean dados, esto no revela los valores de los nodos hoja, que siguen siendo confidenciales.

#### 2.2.1.3. Criptografía de clave pública

El uso de *criptografía de clave pública*, también conocida como *criptografía asimétrica*, es esencial en la TBC. Básicamente en este esquema tenemos dos claves, la clave pública y la clave privada. Cada usuario $A$ dispone de una pareja de claves que denotaremos por $KP_A$ (clave pública de $A$) y $KS_A$ (clave privada, o secreta, de $A$). Estas claves son generadas con un algoritmo que garantiza que lo que una de las claves cifra, sólo la otra clave puede descifrarlo. Aunque desde un punto de vista matemático estas dos claves son intercambiables, en la práctica una de ellas, $KS_A$, debe mantenerse en secreto (siendo de vital importancia su almacenamiento seguro), mientras que la otra, $KP_A$, puede ser distribuida libremente sin que ello suponga ningún riesgo como veremos seguidamente.

Este esquema puede usarse con dos propósitos:

- Para **garantizar el secreto de las comunicaciones.** Cuando otro usuario $B$ quiera enviar un mensaje $M$ de forma confidencial al usuario $A$, cifrará el mensaje

con la clave pública de $A$. Ya que sólo la clave secreta de $A$ puede descifrarlo, se garantiza el secreto de la comunicación.

Obsérvese que para que $B$ pueda realizar esta operación necesita conocer el valor de $KP_A$, pero ya que se trata de una clave pública, no hay peligro en que $A$ la difunda. Cualquiera que tenga el valor de $KP_A$ puede usarlo para cifrar mensajes, pero no para descifrar un mensaje cifrado con $KP_A$, que sólo puede descifrarse con $KS_A$, la cual no es conocida salvo por $A$.

- **Verificar la autenticidad y origen de un mensaje.** En este caso, $A$ puede cifrar un mensaje $M$ usando su clave privada $KS_A$, y difundir este mensaje cifrado. El mensaje no será secreto, ya que cualquiera que esté en posesión de $KP_A$ podrá descifrarlo, y en principio $KP_A$ puede ser conocida por cualquiera. Lo que sí se consigue es la garantía de autenticidad. El mensaje difundido por $A$ puede ser *verificado* por cualquiera, descifrándolo con $KP_A$. El éxito de esta operación garantiza que $A$ es el verdadero autor del mensaje, ya que sólo $A$ tiene acceso al valor de $KS_A$ que fue usado en el cifrado.

  Esta es la base de la firma digital, sólo que habitualmente en lugar de cifrar el mensaje $M$ se cifra tan solo un *hash* del mensaje, es decir, se cifra $H(M)$ por motivos de eficiencia. Entonces $A$ puede comunicar el mensaje $M$ junto con el cifrado de $H(M)$ usando $KS_A$, siendo este último la firma digital del mensaje. Quien recibe el mensaje puede intentar descifrar la firma usando $KP_A$ y comparar el resultado obtenido con lo que sale de aplicar la función *hash* al mensaje, es decir $H(M)$. Si el resultado es el mismo se tiene la garantía de autenticidad del mensaje.

Cabe señalar que en la TBC la criptografía de clave pública se utiliza normalmente sólo con el segundo objetivo, ya que no es importante el secreto de las comunicaciones. La criptografía de clave pública es utilizada para proporcionar un mecanismo de verificación de transacciones, de modo que sea posible verificar que el usuario que genera una transacción está en posesión de su clave privada y por tanto está legitimado para realizar la transacción. Con este objetivo, las claves privadas se utilizan para firmar las transacciones generadas, y las claves públicas se utilizan para generar las direcciones dentro de la red. Otro uso de las claves públicas es la verificación de las firmas generadas con la clave privada.

Como ya hemos visto en la sección 2.2.1, Bitcoin utiliza ECDSA para firmar transacciones y crear la pareja de clave público/privada, y el algoritmo de *hash SHA-256* para generar *hash* únicos para identificar transacciones y bloques o para generar la dirección pública a partir de la clave pública (que finalmente será codificada en base58).

En la figura 2.4 se muestra gráficamente los usos de las primitivas criptográficas utilizadas en la TBC, como son las funciones resumen o *hash* y las firmas digitales.

**Importancia del almacenamiento de la clave privada** En vista a la explicación anterior es deducible que todo el peso de las garantías que un usuario tiene en una red BC recae sobre la clave privada de dicho usuario. Por tanto, es de vital importancia un

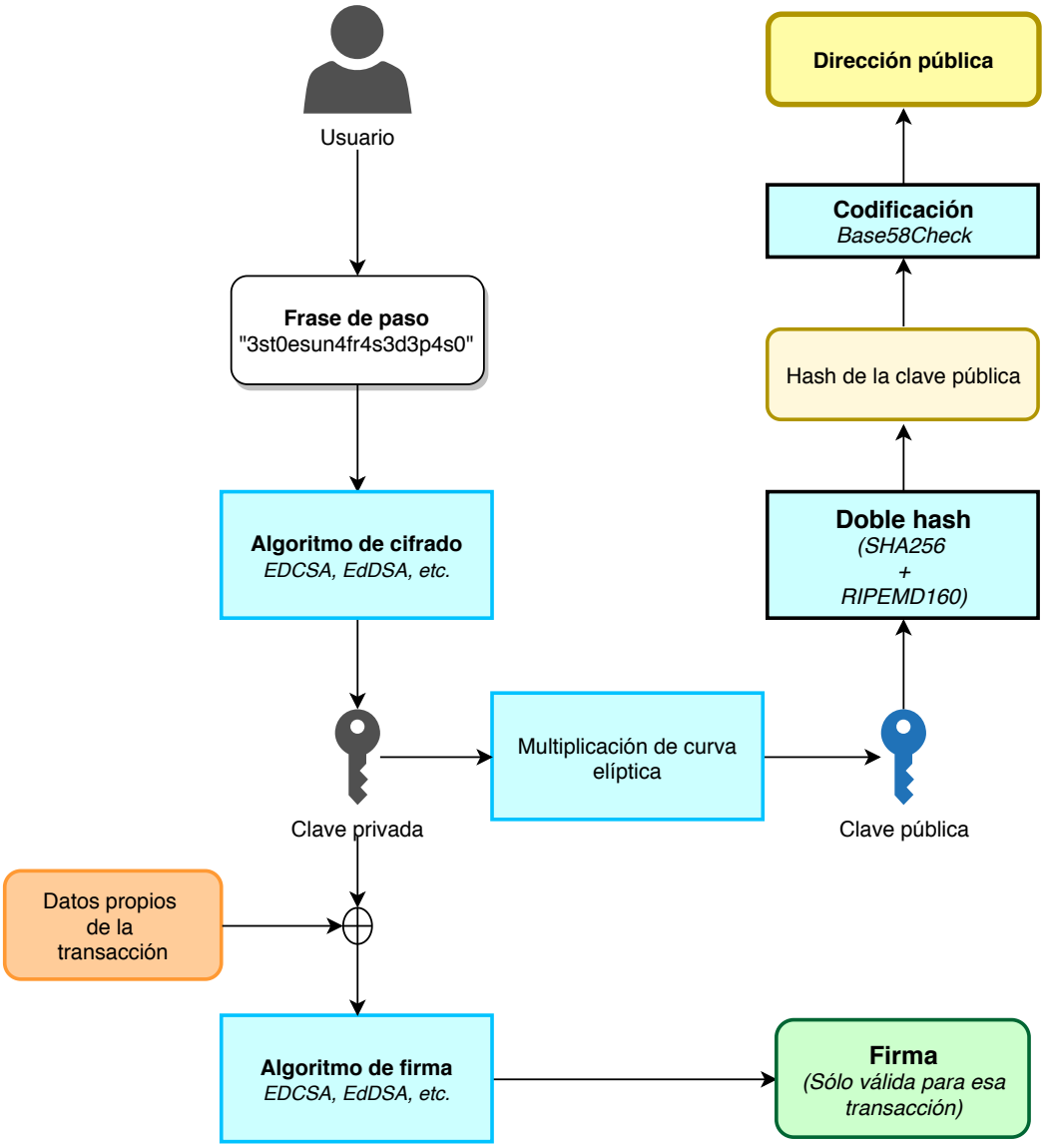


**Figura 2.4.-** Usos de la criptografía en la TBC

almacenamiento seguro que puede hacerse mediante dispositivos *hardware* o *software* llamados billeteras o *wallets*. Las billeteras permiten al usuario almacenar sus claves de forma segura y además ofrecen una serie de funcionalidades útiles para gestionar los activos digitales que posee en la red.

Si un usuario pierde sus claves privadas, pierde todos los activos digitales asociados a dicha clave ya que es inviable computacionalmente generar la misma clave privada a partir de la pública. Si un usuario malicioso obtiene las claves privadas de otro usuario y se transfiere todos los fondos a otra cuenta, no hay manera de revertir dicha transacción

(en condiciones normales[1]).

#### 2.2.1.4. Proceso de transferencia de fondos en una red BC

En este subapartado se mostrará de forma gráfica y práctica cual es el proceso que se sigue de forma general en una red BC para realizar una transacción La explicación y los pasos que en ella se mencionan se refieren a la figura 2.5 en la página siguiente.

Supongamos que un usuario $A$ quiere enviar $x$ bitcoins a un usuario $B$. En primer lugar ambos deben poseer una dirección pública en la red Bitcoin, y el usuario $A$ deberá conocer la dirección pública del usuario $B$ (paso ❶). Además generalmente ambos utilizarán un *wallet* o monedero, que se encargará de administrar sus claves pública y privada, y realizará otras operaciones de interés como consultar la cadena para obtener el saldo actual. El usuario $A$, deberá crear una transacción que incluya su dirección pública (la cual es generada a partir de su clave pública, como hemos visto en la figura 2.4), la dirección pública de B y la cantidad a transferir (paso ❷). El usuario $A$ deberá utilizar la clave privada con la cual generó su dirección pública para firmar la transacción (lo que actúa como medio de autenticación) y además, deberá tener fondos suficientes (paso ❸). Cabe destacar que si introduce la dirección del usuario $B$ de forma errónea, no tendrá opción a recuperar la cantidad transferida, pues la transacción tendrá lugar igualmente y la cuenta destino recibirá los fondos, aún si esa cuenta no pertenece a nadie. Además, dado que es imposible obtener la clave privada asociada con una clave pública y por tanto con una dirección pública, aunque la transacción y la cuenta que recibe los fondos es visible para cualquiera, nadie podrá acceder a esos fondos para gastarlos o transferirlos de nuevo, ya que nadie conoce la clave secreta asociada a la dirección errónea.

La *wallet* enviará la nueva transacción mediante *broadcast* a todos los nodos mineros de la red (paso ❹), y cada nodo la incluirá (si es válida) en el bloque que esté elaborando junto con otras transacciones (paso ❺). Cuando el bloque esté "completo" el minero tratará de encontrar el *nonce* que cumpla la condición impuesta por la red (determinado número de ceros al inicio del *hash* del bloque). El primer nodo minero que consiga resolver el cálculo añadirá el bloque a la cadena (paso ❻) y enviará el bloque al resto de mineros que deberán comprobar que todas las transacciones del bloque son válidas, así como el *hash* resultante . Si todo va bien, añadirán también el bloque a su cadena de bloques (local) "encadenandolo" con el bloque anterior mediante el campo "*hash* del bloque previo" (paso ❻, que ocurre también en el resto de mineros).

Por prudencia, si es una transacción comercial, el usuario $B$ deberá esperar a que se añadan al menos 6 bloques posteriores al bloque que contiene la transacción para entregar el bien o servicio al usuario $A$ (paso ❼). Esta espera se debe a que, como explicaremos en la sección 2.4.2, antes de llegar a ese estado los bloques pueden ser reversibles (pueden ser adelantados por otra cadena más larga, que pasaría a ser la considerada "oficial"). Después de la adición de 6 bloques posteriores a la cadena, el usuario $B$ ya tiene la transferencia realizada de forma completamente inmutable

[1] *Véase "Hard fork" de* Ethereum Classic*, en la página 32*

(paso 8).

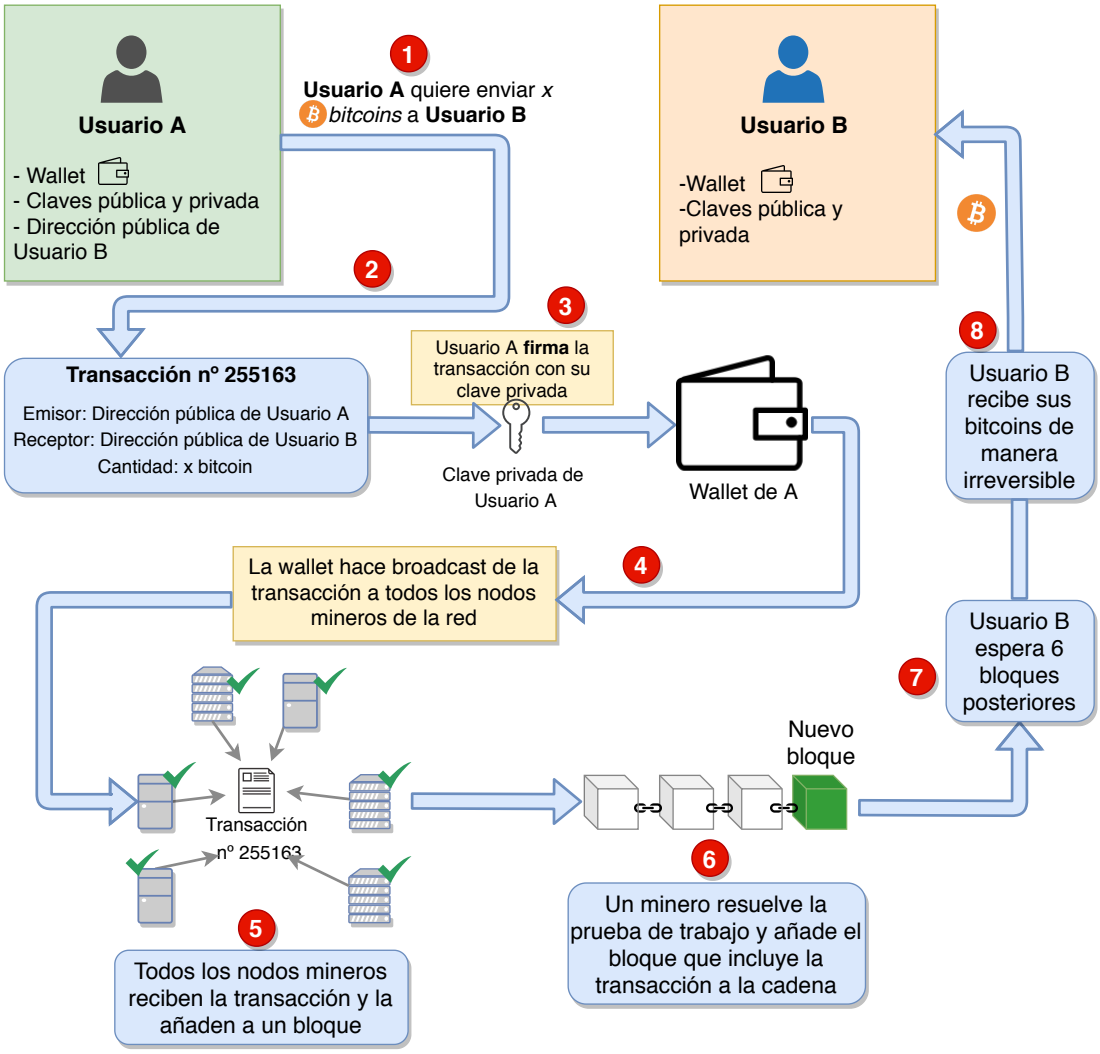


**Figura 2.5.-** Proceso de transferencia de fondos simplificado en Bitcoin

### 2.2.2. Transacciones y bloques

En una cadena de bloques, una transacción es la unidad mínima para almacenar información. Las transacciones se identifican de manera única, comúnmente, mediante un *hash* o una cadena de caracteres alfanuméricos dependiendo de la red BC. Por eficiencia, las transacciones deben ser agrupadas en estructuras especiales llamadas *bloques* para ser añadidas a la cadena.

De forma general, una transacción se estructura de la siguiente forma: posee un identificador único, tamaño de la transacción, estampa de tiempo, número de confir-

maciones, etc. En función de la red (es decir, de si se trata de Bitcoin, o Ethereum, etc.) variará en mayor o menor medida dicha estructura "básica".

No sería una opción inteligente intentar validar cada transacción que se produce en una red de forma individual, pues el número de ellas por unidad de tiempo puede ser muy elevado. Una opción más eficiente es agrupar un conjunto de transacciones en una estructura llamada bloque. Un bloque contiene ciertos datos que lo enlazan con el bloque anterior de la cadena, además de datos propios del bloque. En concreto, el dato que permite el enlazado de un bloque con el bloque anterior es el *hash* o resumen. En los datos propios del bloque encontramos una estampa de tiempo, que indica cuándo se creó dicho bloque, la raíz del árbol Merkle (ver sección 2.2.1.2) que contiene todas las transacciones del bloque, e información adicional que depende del algoritmo de consenso utilizado (p. ej. *nonce* y versión).

En función de la red BC en la que nos encontremos, los bloques tienen unas limitaciones estructurales. Por ejemplo, se establece el tamaño máximo de cada bloque o cada cuánto tiempo se puede añadir un nuevo bloque a la cadena.

En la figura 2.6 se muestra la estructura general de un bloque, y en la figura 2.7 se muestra cómo cada bloque se relaciona con el anterior, y cómo los contenidos del bloque (transacciones) son referenciados desde el árbol Merkle para posibilitar la verificación segura de la pertenencia de una transacción a un bloque (como se explica en la página 13).

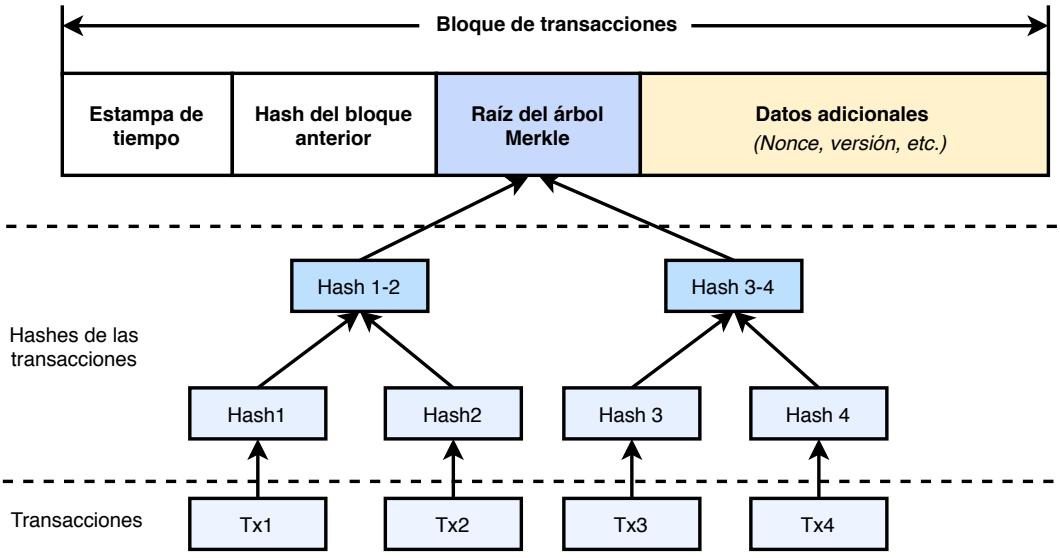


**Figura 2.6.-** Estructura de un bloque de la cadena

La cadena está estructurada de tal manera que cuando se modifica un bloque *"padre"* (o *"abuelo"*, etc.) esto tiene consecuencias en los resúmenes o *hashes* de sus bloques descendientes a modo de cascada. Para mantener la integridad de esos *hashes*, una vez un bloque se ha modificado, es necesario actualizar el *hash* en su hijo, lo que también le modifica, haciendo necesario modificar también el siguiente, etc. Por tanto, cuántos más

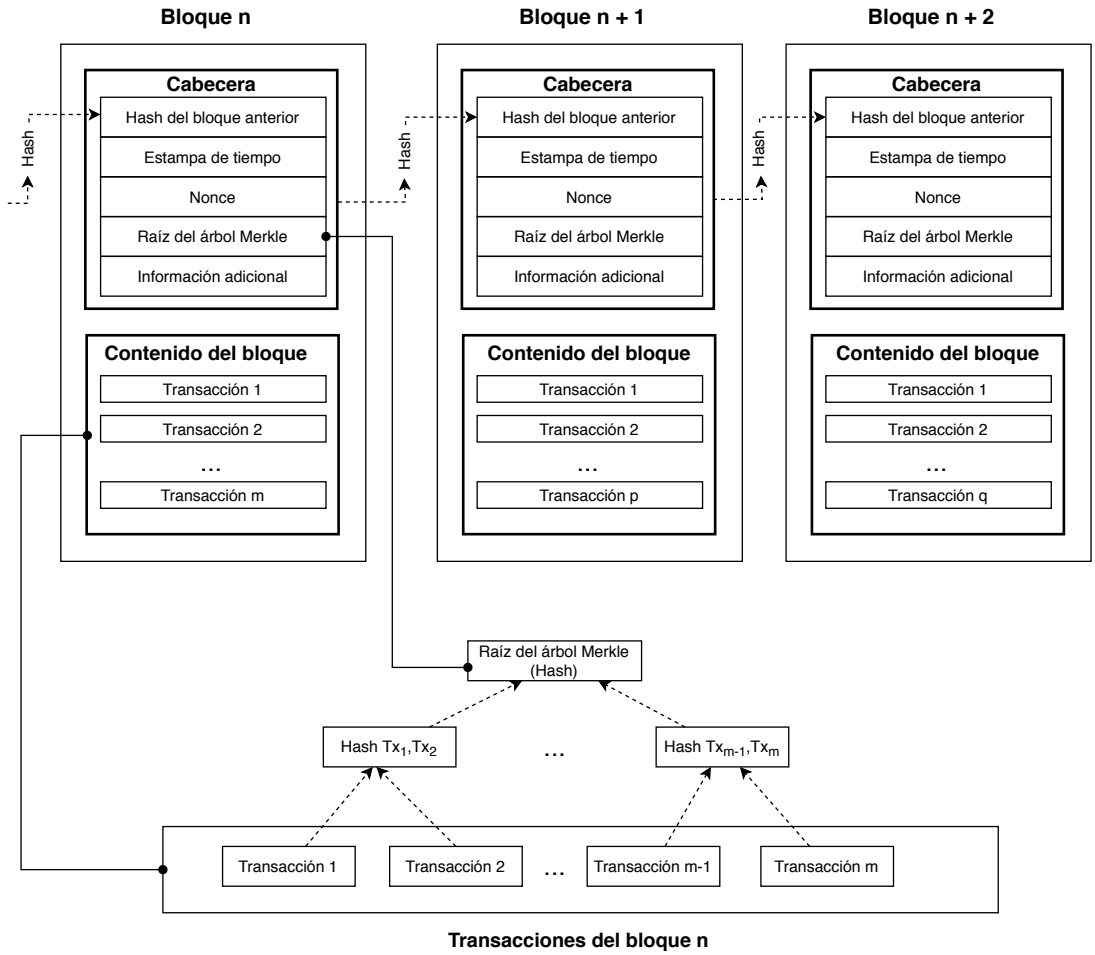


**Figura 2.7.-** Relación entre los bloques de la cadena y sus contenidos

descendientes tenga un bloque, más difícil y costoso se hace modificar algún bloque de la cadena, ya que es necesario modificar todos los siguientes. El grado de inmutabilidad de la misma va aumentando a medida que se añaden nuevos bloques.

Un bloque solo puede tener un bloque padre o bloque anterior, pero se puede dar el caso en el que dos o más bloques consideren que tienen el mismo padre, esto ocurre cuando de manera casi simultánea se añaden dichos bloques. Esto se llama bifurcación o "*fork*" (véase fig. 2.8 en la página siguiente). Este problema se resolverá cuando uno de los hijos sea descartado y se mantenga sólo el otro.

De lo anterior se deduce un punto importante a destacar, y es que, aunque un bloque haya sido añadido a la cadena, las transacciones que contiene no son definitivas. Debe esperarse a que se añadan al menos 5 bloques posteriores a la cadena. Antes de esos 5 bloques sucesivos se podrían producir bifurcaciones (fork) o invalidar por completo la transacción o el bloque (véase fig. 2.8). En Bitcoin el número de bloques está consensuado en la comunidad en 6 bloques (véase Ethos [Ethos18], Zhang *et al.* [ZXL19]

y Bonneau [Bon15]) y dado que los bloques se añaden a un ritmo bajo, esto supondría una espera de unos 60 minutos aproximadamente. En Ethereum deben esperarse 12 transacciones (véase Doloto [Dol16]), lo que supone aproximadamente unos 3 minutos.

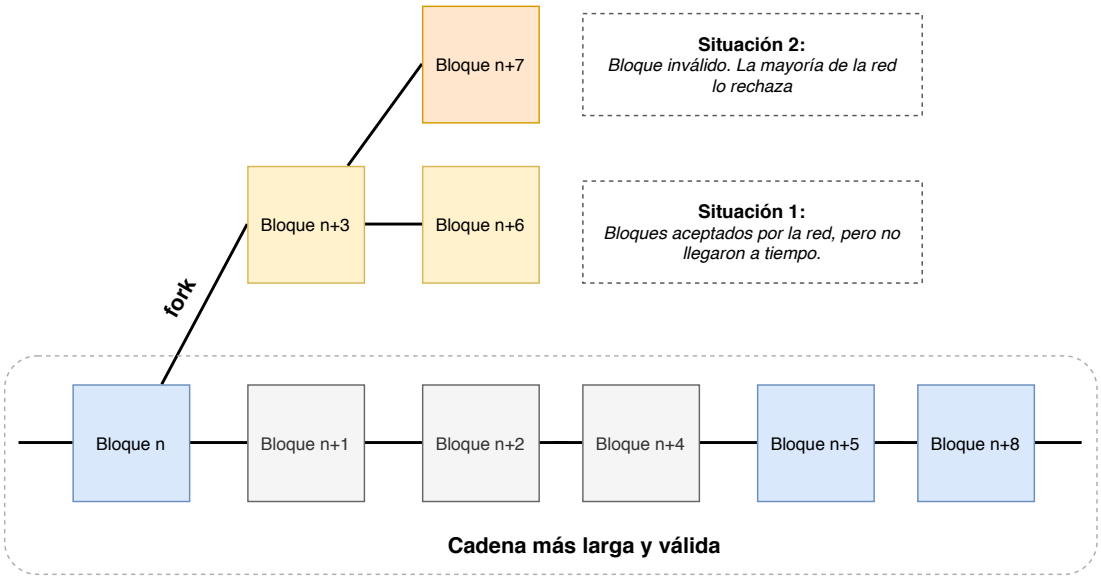


**Figura 2.8.-** Bifurcación (*fork*) de la cadena

### 2.2.3. Algoritmos de consenso

Uno de las aspectos más importantes de una red BC es la decisión de quien tiene la capacidad de añadir nuevos bloques a la cadena. Para ello se utilizan mecanismos de consenso distribuido. Si nos encontramos en una red pública, los modelos de consenso utilizados recompensarán con bienes digitales (p.ej. criptomonedas) a los nodos que agreguen nuevos bloques a la cadena, y por tanto habrá una competencia entre los diferentes nodos candidatos que tienen la capacidad de publicar dichos bloques. Los algoritmos de consenso arbitrarán dicha creación de bloques estableciendo la dificultad de generar un nuevo bloque, la protección de la red frente a usuarios maliciosos y permitiendo el trabajo en equipo de los diferentes nodos productores de bloques sin que sea necesario un alto grado de confianza entre ellos.

En las redes privadas al encontrarnos con un alto grado de confianza entre los nodos, no es necesario ofrecer incentivos económicos para los nodos productores de bloques. A mayor grado de confianza entre nodos, menor coste computacional de creación de bloques. En los siguientes subapartados se presentan los algoritmos de consenso más utilizados en la actualidad en las principales redes BC.

#### 2.2.3.1. Proof Of Work (PoW)

El algoritmo de consenso *Proof of Work* (PoW), o en castellano, *prueba de trabajo*, es el más conocido principalmente por ser el tipo de consenso seguido por las principales criptomonedas, como Bitcoin o Ethereum. En resumen, este tipo de modelos de consenso se basan en exigir a los nodos participantes la resolución de problemas con un coste computacional elevado para poder añadir un nuevo bloque a la cadena. Dicho problema debe ser muy difícil de resolver, pero muy fácil de verificar que ha sido resuelto correctamente, ya que cuando un nodo resuelve el problema, el resto de nodos deberán comprobar si la solución es correcta y aceptar o no ese nuevo bloque en la cadena. Un nodo que quiere publicar un nuevo bloque deberá realizar pequeños cambios en la cabecera del bloque para cumplir con el requisito exigido por el problema (p. ej. que el *hash* comience con un número determinado de ceros), y para cada intento, deberá recalcular el *hash* de la cabecera, lo que se convierte en un problema de alta complejidad computacional, gracias a la propiedad *puzzle friendly* de las funciones *hash* utilizadas. Para ajustar la dificultad del problema se modifica el valor objetivo, lo que repercutirá en la frecuencia a la que se publican nuevos bloques, debido a que se modifica el tiempo necesario para producir un bloque válido. En el caso de Bitcoin, por ejemplo, esa frecuencia de publicación se ajusta modificando el número de ceros iniciales requeridos; a mayor número de ceros, mayor dificultad y viceversa. Los mineros de la red Bitcoin consiguen ese número de ceros generando valores aleatorios, conocidos como *nonce*, que son concatenados a la cabecera del bloque antes de calcular su *hash*. Cuando el minero obtenga el resultado esperado (determinado número de ceros iniciales), el *nonce* que ha utilizado le será útil al resto de nodos mineros de la red para comprobar que en efecto, ha conseguido superar la prueba de trabajo y minar el bloque. Si se da el caso en que dos mineros añaden un bloque a la vez, creándose dos cadenas diferentes de cara al resto de mineros de la red, se aplicaría la regla de la "cadena más larga" y el resto de mineros seguirían añadiendo bloques a la cadena que mayor prueba de trabajo acumule, o sea más "larga".

**Principales ventajas**

- **Defensa de ataques Sybil:** Un ataque Sybil (véase sección 2.4.4.2 en la página 35) sería posible pero bastante improbable debido a la gran capacidad de cómputo requerida para que sea exitoso.
- **Democracia:** Cualquiera con un poder computacional suficiente puede añadir y verificar nuevos bloques en la red, no hace falta poseer un gran número de bienes digitales. Además se obtiene una recompensa bastante interesante. (Ver minar)

**Principales desventajas**

- **Cálculos inservibles:** Los amplios esfuerzos empleados para resolver el puzzle exigido al intentar añadir un nuevo bloque a la red, son en vano, más allá de la red. Dichos cálculos tan complejos y costosos no pueden ser aprovechados para realizar otra tarea.
- **Alto consumo:** El coste computacional y por tanto, eléctrico de la minería asciende de forma casi exponencial conforme pasa el tiempo. Actualmente en la red Bitcoin, la minería únicamente esta disponible para grupos reducidos y especializados de minería que poseen una gran infraestructura y se encuentran situados en localizaciones geográficas especiales, donde el coste de la energía eléctrica sea menor.

#### 2.2.3.2. Proof Of Stake (PoS)

El algoritmo de consenso "*Proof of Stake*" (PoS), o en castellano, "prueba de participación" se fundamenta en la cantidad de monedas o *tokens* que posee cada usuario en la red. La probabilidad de añadir el siguiente bloque a la cadena es directamente proporcional a la cantidad de monedas que posee el usuario, en lugar de la cantidad de capacidad de cómputo como en PoW. La idea subyacente a este protocolo es que supone que los usuarios con más *tokens* están más interesados en la persistencia de la red. Pero no vale solo con tener monedas, además hay que hacer *stake* de ellas, lo que significa que no podrán ser utilizadas por el usuario mientras estén en este estado.

En función de la red se podrán seguir diferentes estrategias con las monedas en *stake* para complementar con el hecho de mantenerlas en *stake*. Algunas de esas estrategias serían: selección aleatoria entre los usuarios que hacen *stake*, sistemas de delegación de votos (algunos usuarios podrían delegar sus votos a otros usuarios para que voten por ellos, no perdiendo nunca la cantidad de monedas que mantienen en *stake*, pero sí la capacidad de voto temporal), sistemas de envejecimiento de monedas (p.ej., aumenta la probabilidad de ser elegido nodo validador en función del tiempo que lleva con sus monedas en *stake*) o varias rondas de votación. Independientemente de la estrategia utilizada los usuarios con más monedas en *stake* tendrán mayor probabilidad de publicar nuevos bloques.

**Principales ventajas**

- **Eficiencia energética:** Es la mayor ventaja de este tipo de algoritmos de consenso con respecto a su predecesor (PoW), al no ser necesario un gasto computacional para producir un nuevo bloque, se elimina ese consumo eléctrico desmesurado.
- **Seguridad frente a ataques del 51 %:** Para hacerse con el control del 50 % + 1 de la red es necesario hacer *stake* de los *tokens*, por tanto, hay un riesgo de pérdida para el atacante. Este riesgo no existe en un protocolos de tipo PoW donde un atacante no puede "perder" su *hardware*.

**Principales desventajas**

- **Whales (Ballenas):** En este tipo de redes se forman las llamadas ballenas que se pueden hacer con el control de la red y tener una gran influencia en las decisiones importantes. Esto derivaría en centralización. El lado positivo es que la existencia de ballenas implica un aumento de la estabilidad de la red.
- **Ataque del "nada en juego" (*nothing-at-stake* ó *stake-grinding*):** Este posible ataque se da cuando los productores o validadores de bloques no tienen nada que perder e intentan realizar bifurcaciones (*forks*) de la cadena, entorpeciendo así el alcance del consenso. El posible beneficio del atacante está en que si su bifurcación termina por ser aceptada como cadena principal por el resto de nodos mineros, habrá obtenido una gran cantidad de recompensas, en caso contrario, no perderá nada. En PoW se evita este problema mediante el alto gasto computacional necesario para añadir un nuevo bloque.

#### 2.2.3.3. Delegated Proof Of Stake (dPoS)

Un paso más allá de PoS tenemos dPoS, que nace como la solución al principal problema de PoS: la centralización de la red en los usuarios que más *tokens* poseen (*whales*). Este algoritmo de consenso planteado por Dantheman [Dan17] en 2013 propone la utilización de un mecanismo de delegación del voto. Los candidatos a ser productores de bloques tendrán que competir entre ellos para conseguir los votos de los usuarios de la red. Dicha competición consiste en realizar acciones que beneficien a la comunidad, como por ejemplo la creación de guías para nuevos usuarios, el desarrollo de herramientas para la red, etc. Los productores de bloques gobernarán el sistema y podrán proponer cambios en el núcleo de la red. El poder de voto será proporcional al número de *tokens* que los usuarios hayan delegado en ellos sumado al número de *tokens* que tengan en *stake*. Prometen mayor escalabilidad que la obtenida con otros protocolos como PoW o PoS.

**Principales ventajas**

- **Mayor escalabilidad:** Dado que no requieren una alta capacidad de cómputo, es accesible para la mayoría de usuarios con un equipo no muy avanzado.
- **Mayor democratización:** Comparados con los algoritmos PoS los usuarios que poseen cantidades pequeñas de monedas tienen mayor poder participativo. Y comparado con PoW al requerir un menor poder computacional cualquier usuario medio puede ser candidato a productor de bloques.
- **Incentivos económicos:** Los productores de bloques son recompensados económicamente por su trabajo, esto crea un interés en ser productor de bloques y en mantener la red segura.

**Principales desventajas**

- **Alto compromiso de participación:** Para que este tipo de redes funcione correctamente debe existir un alto nivel de participación entre los usuarios de dichas redes. Esto hace que la competición entre los candidatos a producir bloques sea más exigente.
- **Centralización:** Existe cierto nivel de centralización que recae en los productores de bloques, aunque es una centralización "asumida" ya que dichos productores han sido elegidos por los usuarios de la red.
- **Problemas inherentes a las votaciones:** Un usuario podría perder el interés en las votaciones porque tiene la sensación de que su voto no vale nada en comparación con los usuarios que poseen grandes fortunas (*whales*).

#### 2.2.3.4. Otros algoritmos relevantes

En un intento de mejorar la eficiencia del algoritmo PoW, surge PoET *(Proof-of-Elapsed Time)*. El algoritmo, estocásticamente, selecciona los nodos que van a procesar las solicitudes, a los que les asigna un número aleatorio de espera. El primer nodo que termine de esperar, será el que procese el siguiente bloque. Para evitar el fraude en este tipo de algoritmos, se utiliza un entorno de ejecución seguro, se verifica la identidad, se hace uso de listas negras basadas en criptografía de clave pública y, por último, se utiliza un conjunto de políticas electorales.

En los algoritmos de consenso PoS, los nodos que poseen cantidades bajas de *tokens* tienen pocas probabilidades de llegar a producir bloques. De este problema surge un algoritmo alternativo conocido como LPoS *(Leased Proof-of-Stake)*. La principal diferencia con PoS es que se incentiva a los nodos pequeños a participar mediante el alquiler de sus *tokens*. La recompensa obtenida por el nodo productor de bloques será repartida proporcionalmente entre los nodos arrendatarios.

El siguiente algoritmo es el *Proof of Burn* (PoB) que nace con el objetivo de mejorar el algoritmo PoW y ser más sostenible con el medio ambiente. En lugar de requerir un elevado gasto computacional como "prueba de trabajo", dicha prueba se consigue mediante el quemado de unidades de *token*, enviándolas a determinadas cuentas llamadas *"eater addresses"*. Este quemado implica que el *token* se convierte en inutilizable, lo que demuestra cierto interés en la continuidad de la red. Además, se recompensa económicamente a los mineros de forma que después de cierto tiempo, el total de las recompensas supere a la inversión realizada por el minero. Entre sus ventajas se encuentra una supuesta reducción energética, ahorro en costes de infraestructura exclusiva de minado (ASIC), se fomenta un compromiso duradero en el tiempo entre los mineros y el quemado de *tokens* implica una escasez en el mercado de la red. Entre las desventajas actuales destacan la lentitud en la verificación de la prueba de trabajo en comparación con PoW. Un usuario promedio podría tener dificultades para verificar la quema de *tokens* y tampoco se ha comprobado que funcione en una red lo suficientemente grande.

Otro protocolo de consenso que se podría considerar respetuoso con el medio ambiente es el *Proof of Authority* (PoA) el cual se basa en la identidad de cada usuario acompañado de un sistema de reputación. Cada nodo validador puede firmar un número máximo de bloques consecutivos durante su turno. Para conseguir ser un validador debe revelar su identidad de forma voluntaria, y mediante el sistema de reputación se garantizará el buen funcionamiento de la red y el interés de los nodos validadores por el mismo. En la práctica es complicado llevar a cabo un algoritmo de este tipo, ya que debemos establecer una manera de comprobar las identidades de forma fiable y eliminar a los posibles nodos maliciosos. Una de las desventajas de este protocolo es que solo tiene sentido en una red privada donde ya existe un cierto grado de confianza entre nodos.

Por último, nos encontramos con otro algoritmo ideal para redes híbridas o privadas donde existe cierta confianza entre nodos, el *Practical Byzantine Fault Tolerance* (PBFT) propuesto por Castro y Liskov [CL99] ideado para trabajar de manera eficiente en sistemas asíncronos (esto incluye a los sistemas distribuidos generales y las TBC). Su principal objetivo es resolver los problemas asociados a las soluciones propuestas hasta el momento para solucionar los fallos bizantinos [LSP82]. Los nodos de un sistema distribuido que utiliza PBFT pueden ser de dos tipos: *leader node* (nodo líder), del existirá un único nodo, y *backup nodes* (nodos de respaldo), que serán todos los demás. En caso de fallo del nodo líder, se asignará como líder a uno de los nodos de respaldo. El objetivo principal es alcanzar el consenso entre todos los nodos, guiándose por la regla de la mayoría. Debemos tener en cuenta que para un correcto funcionamiento del protocolo, los nodos maliciosos (o defectuosos) no deben ser superiores a un tercio de los nodos totales del sistema. A mayor número de nodos totales, mayor seguridad.

Entre sus desventajas encontramos la escalabilidad, ya que en cada comunicación deben intervenir todos los nodos de la red, creando así una sobrecarga. A medida que aumenta el número de usuarios, aumenta el tiempo necesario para responder a la petición, siendo éste del orden $O(m^n)$, donde $m$ es el número de mensajes y $n$ el número de nodos. Esto implica que en la práctica el protocolo PBFT necesita ser implementado en conjunto con otros protocolos, como por ejemplo, PoW [véase por ejemplo Zil17] o dPoS [véase por ejemplo Tender18]. Relacionado directamente con la escalabilidad tenemos la posibilidad de ataques Sybil, que es mayor cuanto menor sea el número de nodos que componen la red.

### 2.2.4. Contratos inteligentes

La primera definición en la literatura de *"contrato inteligente"* fue propuesta por Szabo [Sza97], quien definió los contratos inteligentes como un protocolo de transacciones que ejecuta los términos de un contrato. Los objetivos generales de diseño son satisfacer las condiciones del contrato, minimizar todo tipo de excepciones y la necesidad de intermediarios de confianza.

En una red BC, un contrato inteligente o *smart contract* es un conjunto de datos y código desplegado en la red mediante el uso de transacciones. El código del contrato inteligente es ejecutado por los nodos y siempre es determinista: para una entrada $E$

siempre producirá la misma salida $S$. El resultado de dicha ejecución se almacenará en la cadena de bloques. Permite una automatización de procesos de manera descentralizada, verificando y ejecutando las condiciones que se encuentran en el acuerdo. Se puede compartir cualquier bien (dinero, propiedades, etc.) entre dos partes de manera transparente (el código del contrato debe ser público) sin la necesidad de un tercero de confianza, eliminando los posibles conflictos de intereses.

Obviamente, el hecho de eliminar la autoridad central o un tercero de confianza en un proceso y delegar esa tarea en un contrato inteligente hace que su programación sea muy ardua, ya que no debe contener errores de ningún tipo que puedan ser explotados de alguna manera.

A continuación, se detallan las características de los contratos inteligentes de forma general:

- **Eficiencia temporal:** Cualquier proceso burocrático puede llevar días, semanas o incluso meses que pueden ser debidos, por ejemplo, a la cantidad de intermediarios que deben revisar o aprobar los documentos. Con los contratos inteligentes estos procesos se acelerarían ya que según esté establecido en el contrato inteligente, una vez se cumplan las condiciones necesarias se obtendrá una salida en función a dichas entradas.

- **Seguridad:** Los contratos inteligentes heredan la seguridad inherente en la BC, lo que nos asegura la no modificación de los contratos a posteriori (puesto que están almacenados en la BC). Los riesgos de seguridad están asociados a errores de programación, siendo necesario una extenso abanico de pruebas de todo tipo, y a ser posible una auditoría antes de desplegar un contrato inteligente en producción.

- **Acuerdos colectivos:** Se pueden diseñar clausulas multi-firma que hagan que el contrato ejecute una determinada acción una vez se alcanza un porcentaje determinado de acuerdo. Por ejemplo, se podría orquestar una apuesta entre un grupo de personas, donde cuando más del 60 % envíen al contrato el nombre de la persona ganadora, el contrato automáticamente ejecuta la acción (p.ej. envío de fondos).

En las redes públicas como Ethereum es común que la ejecución de un contrato inteligente conlleve un gasto económico, proporcional al número de instrucciones del contrato ejecutadas, limitando así el tiempo de ejecución consumido por una llamada al contrato. Esto nos asegura un interés por parte de los productores de bloques de ejecutar contratos inteligentes en sus nodos (pues reciben una contraprestación económica por ello) y además protege a la red de usuarios maliciosos que desarrollen código que al ser ejecutado pueda causar un mal funcionamiento del nodo o de la red (pues han de pagar por ejecutar código, y éste dejará de ejecutarse cuando se agote la provisión de fondos asignada a su ejecución).

La primera red BC en implementar los contratos inteligentes fue Ethereum, creando un lenguaje de programación para dichos contratos llamado Solidity, el cual es Turing

completo.[1]

## 2.3. Tipos de redes

A continuación se presentan los diferentes tipos de redes existen en BC. En función de quién pueda participar y cómo se obtiene esa posibilidad de participación, las redes se dividen en públicas o privadas. Además se puede contemplar un tipo de red *"híbrida"* que es una combinación de las características de los otros dos tipos.

Se puede establecer otra categorización en función de quién puede leer o escribir en la cadena: redes con permisos (*Permissioned*) o sin permisos (*Permissionless*). Una red puede ser pública con o sin permiso y privada con o sin permisos. En la tabla 2.1 se muestran estas combinaciones.

En una red BC pública sin permisos todos los usuarios están en igualdad de condiciones, mientras que en una con permisos algunos usuarios tienen más valor que otros. En las redes privadas ocurre algo similar, una red privada y sin permisos puede ser interesante para colaboraciones entre empresas sin la necesidad de establecer un consorcio, aunque puede no ser el tipo de red más adecuado para la mayoría de casos, siendo el tipo de red privada más común las que incluyen permisos, las cuales están controladas por una entidad.

### 2.3.1. Públicas

La principal característica de este tipo de redes es que cualquiera puede convertirse en usuario de la red, y arrancar un nodo que verifique y añada bloques a la red. Aunque en la práctica, debemos tener en cuenta distintas variables que dependen de la red en la que nos encontramos. El ejemplo más claro sería el de una red que utiliza un algoritmo de consenso como *Proof Of Work* (PoW) en la cual para añadir nuevos bloques debemos resolver computacionalmente un problema matemático que implica un gasto excesivo tanto a nivel computacional como de consumo eléctrico. Por tanto, en la práctica no cualquier usuario podría añadir nuevos bloques a este tipo de redes. Esta restricción es necesaria porque de no existir, cualquier usuario de la red podría tener intenciones maliciosas para la misma, e intentar añadir bloques incorrectos con el fin de obtener algún beneficio. Las medidas para evitar intenciones maliciosas varían en función del algoritmo de consenso utilizado como hemos visto en la sección 2.2.3. Además comúnmente para tratar de evitar posibles usuarios maliciosos en la red e incentivar que algunos usuarios se planteen ser "creadores/verificadores de bloques" (mineros), éstos obtienen una recompensa cada vez que añaden un nuevo bloque a la red, normalmente en forma de la criptomoneda nativa de la red.

Algunos ejemplos de redes públicas, independientemente del algoritmo de consenso utilizado, serían Bitcoin, Ethereum o EOS.

[1] Cualquier lenguaje de programación que pueda simular una máquina de Turing Universal, o dicho con otras palabras, que sea capaz de ejecutar cualquier cálculo computacional si dispone de los recursos necesarios.

| | **Públicas** | |
|---|---|---|
| | **Con permisos** | **Sin permisos** |
| Definición | Algunos tienen más permisos que otros | Igualdad de condiciones |
| Objetivo | Centralización | Escalabilidad |
| Debilidad | Privacidad | Privacidad |
| Ejemplos | EOS, Ripple | Bitcoin, Ethereum |

| | **Privadas** | |
|---|---|---|
| | **Con permisos** | **Sin permisos** |
| Definición | Similar a una red privada corporativa pero en Blockchain | Colaboración entre empresas sin consorcio |
| Objetivo | Centralización | Escalabilidad |
| Debilidad | Consenso | Consenso |
| Ejemplos | Hyperledger, CORDA | Holochain |

**Tabla 2.1.-** Diferentes combinaciones de redes BC en función del ámbito y los permisos

### 2.3.2. Privadas

Se denomina **blockchain de tipo privada** a aquella en la que solo pueden añadir y verificar nuevos bloques determinados usuarios autorizados por la entidad que controla la red (autoridad), que puede ser centralizada o descentralizada. Como es de esperar, en este tipo de redes los usuarios no autorizados no podrán añadir información a la cadena de bloques, o incluso pueden tener prohibido el acceso para lectura. La autoridad podrá personalizar los permisos de lectura o escritura que proporciona a cada participante de la red de forma individualizada.

La **trazabilidad** de la información que se almacena en la red también puede ser distribuida, al igual que en las redes públicas, pues así pueden ser más resilientes, y los datos también se mantienen replicados en los nodos.

Además, aunque también utilizan algoritmos de consenso, en este caso no es necesario que tengan asociado un coste computacional o de recursos de tipo PoW, ya que todos los usuarios de la red están autorizados por la entidad autoridad que controla la red, y existe una confianza total o parcial ellos. Teniendo en cuenta este hecho, este tipo de redes implican un consumo de recursos mínimo y una alta velocidad de transacción.

Este tipo de redes son utilizadas principalmente en tres escenarios:

1. Varias organizaciones que desean trabajar juntas en una red BC pero sin existir confianza entre alguna de ellas. En este caso, la red proporciona esa confianza en mayor o menor medida en función del tipo de algoritmo de consenso utilizado.

2. Una organización desea tener un control absoluto de la red BC a utilizar y de los datos que se almacenan en ella, además de restringir el acceso a dichos datos. En este escenario, todos los usuarios de la BC deben confiar en la organización.

3. Cualquier otro escenario en el que no exista confianza total entre las partes, la probabilidad de censura sea inexistente o casi nula y no sea importante que la red y los datos estén abiertos al público.

Algunos ejemplos de este tipo de redes serían Corda [Corda15] o cualquier red BC creada con el *framework* Multichain [Multichain15].

### 2.3.3. Híbridas

Por último, el tercer tipo de redes serían las híbridas o consorciadas. Se definen como una combinación de las dos anteriores. La red no está controlada por la comunidad como en las redes públicas ni por una única entidad como en las redes privadas, sino que en este caso, está controlada por un conjunto de nodos determinados (consorcio).

El acceso a la información de la red podrá ser restringido a determinados usuarios o ser público. Este tipo de redes se considera que están parcialmente centralizadas, ya que al establecer un conjunto de nodos productores de bloques, se podría alcanzar un cierto grado de descentralización, o por el contrario si el conjunto de productores es reducido, sería bastante centralizada. Al ser una combinación de redes públicas y privadas, pueden aprovechar las virtudes de cada tipo y evitar los defectos. Esto brinda una gran flexibilidad para las empresas ya que pueden configurar un espacio de trabajo a medida para interactuar con sus partes interesadas de la manera más eficiente posible.

Entre los beneficios que encuentran las empresas en este tipo de redes, podemos destacar la protección contra el ataque del 51 % ya que la incorporación a la red está limitada. Además protege ciertos datos almacenados en la cadena, a la vez que se pueden comunicar con el exterior. Por último, pueden cambiar las "reglas del juego" cuando sea preciso, aunque para que sea útil este tipo de redes se deben establecer restricciones en función del uso que se le dé a la red. Un ejemplo de este tipo de redes sería cualquier red creada con el *framework Hyperledger Fabric*.

La tabla 2.2 en la página siguiente muestra una comparativa esquemática entre los tres tipos de redes vistos.

## 2.4. Ventajas y desventajas de la Tecnología Blockchain

En esta sección expondremos las principales ventajas que proporciona el uso de la TBC, como son la descentralización, la inmutabilidad y la escalabilidad, pero también explicaremos la parte negativa, exponiendo los principales riesgos de seguridad a los que se enfrenta esta tecnología en la actualidad.

| | Tipos de redes blockchain | | |
|---|---|---|---|
| **Propiedad** | Pública | Privada | Híbrida |
| Productores | Cualquier usuario | Una entidad | Conjunto de nodos seleccionados |
| Permisos | Públicos | Públicos, parcialmente públicos o restringidos | Públicos, parcialmente públicos o restringidos |
| Inmutabilidad | Prácticamente garantizada | No asegurada | No asegurada |
| Centralización | No | Si | Parcial |

**Tabla 2.2.-** Comparación entre redes de tipo públicas, privadas e híbridas.

### 2.4.1. Descentralización

Uno de los aspectos fundamentales de las redes BC es el grado o nivel de centralización. Teóricamente, el nivel de centralización dependerá del tipo de red. Por ejemplo, las redes públicas son totalmente descentralizadas, las redes privadas son totalmente centralizadas en una entidad única y las redes híbridas una mezcla de las dos anteriores, estando parcialmente centralizadas en un conjunto determinado de nodos.

En la práctica, en las redes privadas o híbridas lo antes dicho se cumple en la mayor parte de los casos, pero en las redes públicas no es necesariamente así. En función del tipo de consenso utilizado la red puede estar, de forma no intencionada, centralizada en un conjunto de nodos que poseen ciertas características. A menor número de nodos en dicho conjunto, mayor grado de centralización. Ocurre en redes que utilizan *Proof of Work* (PoW), donde la red está controlada por un pequeño conjunto de mineros o en *Proof of Stake* (PoS), donde se entrega una cantidad de *tokens* a los productores de bloques, enriqueciéndolos con el paso del tiempo y haciendo cada vez más difícil que existan nuevos productores de bloques. También ocurre en el tercer algoritmo de consenso más común, el *Delegated Proof of Stake* (dPoS) que, aunque es el más descentralizado de los tres, cae en una cierta centralización "temporal" en el conjunto de productores de bloques elegidos mediante votación.

### 2.4.2. Inmutabilidad

Otra de las supuestas ventajas más habitualmente mencionadas para promocionar la tecnología BC es la inmutabilidad, de la que se dice que está asegurada. Pero realmente la cadena de bloques no es 100 % inmutable, ya que hay determinados escenarios en los que se puede alterar la cadena.

El primer escenario en el que no se cumple la inmutabilidad de la cadena se da en

las redes en las que, cuando hay varias cadenas compitiendo por ser la "buena", los últimos bloques añadidos a la cadena corren el riesgo de ser eliminados de la misma porque exista una cadena más larga que sea aceptada por la mayoría como la versión válida. Es por ello que los usuarios de las redes BC deben esperar a la publicación de varios bloques sucesivos para considerar que su transacción se ha realizado con éxito.

Uno de los ejemplos más icónicos de la mutabilidad de la cadena de bloques es el conocido como "ataque DAO". En 2016 se desplegó en la red pública Ethereum el contrato inteligente del proyecto DAO[1] *(Decentralized Autonomous Organization)* que pretendía ser un fondo de inversión descentralizado que, entre otras cosas, sirve para financiar mediante la moneda nativa nuevos proyectos para desarrollar en la red. Debido a la naturaleza del contrato cabía la posibilidad de extraer fondos de forma no autorizada, aunque para el creador del contrato este escenario era altamente improbable. Pero ocurrió, un atacante logró substraer 3.6 millones de Ether, que en ese momento equivaldrían a $50M. Como contramedida se creó una bifurcación (*hard-fork*) de la red con el resultado de dos nuevas redes: *Ethereum*, red que revirtió las transacciones del ataque DAO, y *Ethereum Classic*, una red que mantiene un compromiso de inmutabilidad, y en la que por tanto, no se revirtieron las transacciones del ataque.

En las redes privadas es aún más sencillo que no se cumpla dicha inmutabilidad ya que la entidad propietaria de la red puede añadir o eliminar a su antojo nodos productores de bloques. Además al tener un control total sobre la cadena puede eliminar bloques cuando considere necesario. Hay que tener en cuenta que dicha eliminación de bloques conlleva un coste asociado, ya que es necesario rehacer los hashes de la cadena, pero debido a los algoritmos de consenso utilizados en las redes privadas, este coste es prácticamente nulo.

### 2.4.3. Escalabilidad

Al igual que en los sistemas distribuidos tradicionales existe una limitación expresada en el teorema de Brewer[2][Bro20; FB99], en la TBC se tiene también un famoso trilema. En la figura 2.9 se muestra el trilema de la escalabilidad en las redes BC. Esto significa que cualquier modificación en una de las tres variables principales, *seguridad*, *escalabilidad* y *descentralización*, repercute inversamente en las otras dos. Por tanto, es imposible garantizar un cumplimiento total de las tres variables al mismo tiempo.

#### 2.4.3.1. Factores que afectan a la escalabilidad

- **Protocolo de consenso:** Cuando existen diferentes incentivos en función de que transacciones se añadan a la cadena, los productores de bloques siempre intentarán procesar las transacciones que les aporten un mayor beneficio. Esto deriva en un cuello de botella para las transacciones con menor tasa de ganancia.

[1] No confundir el "Project DAO" con Organización Autónoma Descentralizada (DAO)

[2] Teorema de Brewer (también conocido como teorema CAP): Un sistema distribuido solo puede garantizar simultáneamente, como máximo, dos de las siguientes características: consistencia, disponibilidad y tolerancia al particionado.

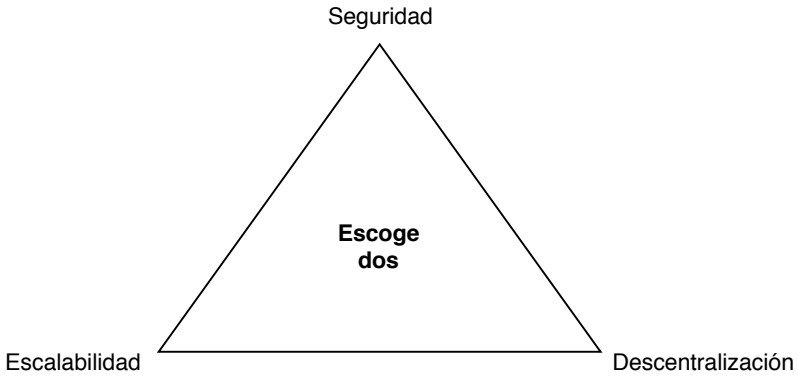


**Figura 2.9.-** Trilema de la escalabilidad en redes Blockchain

- **Tiempo de confirmación:** Hay situaciones en las que crear un bloque válido no garantiza que las transacciones contenidas en él estén aseguradas (como se explicó en la sección 2.2.2).
- **Tamaño del bloque:** Este tamaño afecta al tiempo entre cada publicación de un nuevo bloque y esto limita el número de transacciones por segundo de la red, lo que afecta significativamente a la escalabilidad.

#### 2.4.3.2. Limitaciones a la escalabilidad

- **Almacenamiento en disco:** Con el paso del tiempo, aumenta el número de transacciones en el histórico, y todos los nodos deben almacenarlas de forma persistente. Este problema se acentúa en las redes con contratos inteligentes y *dApps* ya que aumentan significativamente el número de transacciones. Actualmente tener un nodo completo en redes como Bitcoin o Ethereum implica que como mínimo debes tener más de 237 Gb[1] o más de 3TB[2] de disco duro SSD, respectivamente.
- **Coste de mantenimiento:** Como consecuencia del tráfico elevado de la red los nodos necesitan procesar un mayor número de transacciones, lo que conlleva un gasto elevado de recursos computacionales (y consumo eléctrico).
- **Tiempo de respuesta:** Otro problema que limita la capacidad de escalar de una red BC es el tiempo de respuesta. En función del tipo de red, se crearán bloques cada cierto periodo de tiempo. Este tiempo de espera se puede considerar tiempo de respuesta de la red, ya que si por ejemplo, nos encontramos en una red en la que se crean bloques cada 10 minutos, y nuestra transacción se añadirá en el próximo bloque y se acaba de añadir uno hace 2 minutos, como consecuencia, existirá un tiempo de espera de 8 minutos, aproximadamente.

[1]Fuente: https://www.blockchain.com/es/charts/blocks-size (consultado el 01/09/19)
[2]Fuente: https://etherscan.io/chartsync/chainarchive (consultado el 01/09/19)

#### 2.4.3.3. Posibles soluciones

Entre las posibles soluciones a los problemas de escalabilidad nos encontramos la posibilidad de aumentar el tamaño del bloque para aumentar ese número de transacciones por segundo. Otro posible enfoque sería eliminar cierta información de las transacciones, con el fin de hacerlas más "ligeras", conservando únicamente la información estrictamente necesaria, lo que conllevaría un aumento del número de transacciones por bloque. Además, se podría modificar el algoritmo de consenso de la red e implementar un algoritmo de consenso diferente que mejore el rendimiento de la red, como por ejemplo *Proof of Stake* (PoS) o *Proof of Authority* (PoA). Por último, se puede plantear una solución *off-chain* que realice ciertas tareas en un sistema tradicional y así minimizar el gasto computacional a realizar por la red.

### 2.4.4. Riesgos de seguridad

Como toda tecnología o sistema informático, las cadenas de bloques también presentan riesgos conocidos que pueden afectar a su estabilidad. En esta sección se presentan los ataques más importantes que puede sufrir este tipo de sistemas. Los más conocidos serían los ataques de doble gasto (p. ej. Finney o 51 %) o el ataque Sybil. Algunas de estas amenazas no son exclusivas de la TBC, como por ejemplo, los ataques DDoS o algunos ataques de tipo *hijack*.

#### 2.4.4.1. Ataques de doble gasto

El doble gasto engloba a todo tipo de estrategias o ataques cuya intención es utilizar dos veces y simultáneamente el mismo *token* único dentro de una red BC. Un usuario $A$ envía un *token* a un usuario $B$ y a la vez podría enviar el mismo *token* a un usuario $C$, siempre que posea el poder computacional necesario (para reescribir la cadena de modo que el bloque que registró la primera transacción ya no forme parte de ella). Para evitar este tipo de estrategias, el usuario receptor de una transacción deberá a que se confirmen entre 3 y 6 bloques posteriores al bloque en el que se incluye su transacción. Esto hace más difícil eliminar el bloque de la cadena, ya que la capacidad computacional necesaria para reemplazar no sólo ese bloque sino también los que le siguen, ha de ser mayor que el del resto de la red BC, que seguirá añadiendo bloques a la cadena más larga.

Dentro de esta categoría se engloban los ataques de carrera, ataque Finney, Vector 76, fuerza bruta (sección 2.4.4.3 en la página 36) o del 51 % (a continuación).

**Ataque del 51 %** El ataque del 51 % es uno de los más conocidos en el ámbito de las redes BC. Se da cuando un usuario de la red posee más poder computacional que el resto de usuarios juntos. El atacante que consiga esto podría, teóricamente, añadir nuevos bloques más rápido que los demás y que su cadena siempre sea la más larga y por tanto, la válida. Además también podría en principio bloquear transacciones válidas simplemente no añadiéndolas a los bloques, o duplicar alguna de esas transacciones.

No obstante, el atacante no puede hacer cualquier cosa. Por ejemplo, no podría enviar transacciones desde una cuenta de la cual no posea su clave privada, ni podría crear nuevos *tokens* nativos de la red *"de la nada"*.

Una acción que el atacante sí podría intentar sería un "*doble gasto*", es decir, utilizar el mismo *token*, dos veces. Por ejemplo imaginemos un escenario en el que tenemos al atacante y a un *exchange*, que recibe *tokens* de la red y hace el cambio a una divisa tradicional para enviárselo al usuario por medios tradicionales. El atacante enviaría ese *token* al *exchange* en la red pública, el *exchange* vería reflejada en la BC dicha transacción y procedería a enviar a la cuenta bancaria tradicional del atacante la cantidad correspondiente. A su vez, el atacante estaría creando una nueva cadena de forma privada en la que esa transacción no se produjo nunca. Una vez que el atacante reciba la transacción en su cuenta bancaria, si tiene el suficiente poder computacional, su cadena, hasta entonces privada, será la más larga y será dada por válida por el resto de la red. Si esto ocurre, el atacante seguiría teniendo el *token* "enviado" al *exchange*, y también la cantidad transferida en una divisa tradicional *off-chain*.

En redes como Bitcoin, este tipo de ataques no es viable ni rentable, debido al enorme gasto económico y los dudosos beneficios que se obtendrían. De hecho, si un conjunto de nodos consiguen un poder computacional equivalente o superior al 51 % de la red podrían realizar este tipo de ataque. El único beneficio real que podrían obtener sería el poder llegar a realizar un doble gasto en en algún momento, pero presumiblemente éste sería invalidado por el resto de nodos y el bloque con la transacción fraudulenta sería invalidado mediante una bifurcación o *fork*.

#### 2.4.4.2. Ataque Sybil

Se trata de un término acuñado por Douceur [Dou02], que describe este tipo de ataques que afecta de forma general a redes P2P. El ataque Sybil consiste en crear múltiples identidades falsas para conseguir cierta influencia en al red y así poder llevar a cabo acciones contrarias a las normas de la red. Un ejemplo sencillo de este tipo de ataque lo proporcionan las primeras redes P2P usadas para compartir ficheros con derechos de autor, como por ejemplo, *Napster*. Una discográfica detectaba en dicha red un archivo que contenía una canción que acababa de salir al mercado y, para dificultar su propagación, llevaba a cabo un ataque Sybil. La discográfica creaba múltiples identidades en la red que añadían a la red ficheros idénticos a la canción pero con diferente contenido. Al usuario de la red que busque este fichero le será muy complicado distinguir entre todos los ficheros, cuál es el real. En una red BC, este tipo de ataques consiste en crear múltiples identidades con el fin de controlar dicha red, ya sea total o parcialmente.

Se podría evitar este tipo de ataques añadiendo un coste asociado a la creación de una nueva identidad en la red, añadiendo diferentes roles a los usuarios que conlleven un mayor o menor peso en la red o validando las identidades antes de unirse a la red. En las redes BC se intenta evitar este tipo de ataques mediante los algoritmos de consenso utilizados, en PoW se combate mediante la minería de bloques, en PoS mediante el sistema de "*stake*", etc.

#### 2.4.4.3. Otros ataques relevantes

Además de los tipos de ataque anteriores Muhammad *et al.* [Muh+19] muestran un total de 22 ataques posibles en una red BC organizados según su clase de ataque. Definen tres tipos de ataques en función del componente afectado: ataques a la estructura de la cadena, al sistema *Peer-to-Peer* (P2P) o a la aplicación de la BC. Muestran los elementos afectados por cada tipo de ataque, las consecuencias que podrían tener y proponen contramedidas. Algunas de esas contramedidas son comunes a varios ataques, lo que podría derivar en una orientación hacia una futura solución general a estos problemas.

**Ataques a la estructura de la cadena**

**Bloques huérfanos:** Ocurre cuando el proceso de consenso no incluye un bloque en la cadena. Afecta a la red, mineros y grupos de minería.

- *Implicaciones:* Puede causar pérdida de ingresos.
- *Contramedidas:* Incrementar el tiempo entre bloques.

**Bifurcaciones:** Cuando se produce una bifurcación los nodos de la red se encuentran en diferentes estados de la cadena que puede persistir durante largos periodos de tiempo o de forma indefinida. Afecta a la red.

- *Implicaciones:* Puede causar división de la cadena y pérdida de ingresos.
- *Contramedidas:* Consenso conjunto.

**Ataques al sistema P2P**

**DNS hijacks:** Consiste en la modificación de las direcciones IP a las que el servidor DNS debería dirigir al usuario para enviarle en cambio a una página distinta y comúnmente maliciosa. Afecta a los mineros, grupos de minería, exchanges y usuarios.

- *Implicaciones:* Puede causar pérdida de ingresos, particionado de la cadena y robos.
- *Contramedidas: Routing-awareness*.

**BGP hijacks:** Consiste en la modificación de forma no autorizada y con fines maliciosos de las tablas de enrutamiento del protocolo BGP. Afecta a los mineros, grupos de minería y usuarios.

- *Implicaciones:* Puede causar pérdida de ingresos, particionado de la cadena y robos.
- *Contramedidas: Routing-awareness*

**Ataque de eclipse:** Este tipo de ataque ocurre cuando un grupo de nodos maliciosos aísla a uno o varios nodos de un clúster (p. ej. en Bitcoin) modificando así su vista de la "cadena". Afecta a los mineros y usuarios.

- *Implicaciones:* Puede causar particionado de la cadena.
- *Contramedidas:* Monitorización de los nodos.

**Ataque 51 %:** Descripción en la sección 2.4.4.1. Afecta a la red, mineros y aplicaciones.

- *Implicaciones:* Puede causar división de la cadena, pérdida de ingresos y minería malintencionada.
- *Contramedidas:* Utilizar el algoritmo de PoW con doble fase.

**Minería egoísta:** Este tipo de ataque ocurre cuando un nodo retiene bloques en lugar de añadirlos a la cadena y difundirlos al resto de la red, para obtener así un mayor beneficio económico. Afecta a la red, mineros y grupos de minería.

- *Implicaciones:* Puede causar pérdida de ingresos y minería malintencionada.
- *Contramedidas: Time-stamping blocks.*

**Ataques DDoS:** Este tipo de ataque es la evolución del ataque DoS, donde ahora se utilizan diferentes dispositivos los cuales realizan peticiones de forma simultánea hacia una misma dirección destino, con el fin de inhabilitar a la víctima. Afecta a la red, mineros y grupos de minería.

- *Implicaciones:* Puede causar minería malintencionada.
- *Contramedidas:* Incrementar el tamaño del bloque.

**Retraso del consenso:** En este tipo de ataque un nodo podría generar bloques fraudulentos con el fin de aumentar la latencia de la red, retrasando así que la red alcance un consenso. Afecta a los mineros y grupos de minería.

- *Implicaciones:* Puede causar retrasos y pérdida de información.
- *Contramedidas:* Monitorización de los nodos.

**Retención del bloque:** Este ataque es muy parecido al anterior, pero en este caso el minero que retiene el bloque (que pertenece a un *pool* de minería) tiene como objetivo aliarse con otro *pool* de mineros para compartir con ellos el esfuerzo realizado y obtener así mayor recompensa. Otra opción es que el minero difunda el bloque desde otra dirección que no pertenece a ese *pool* de mineros. Afecta a los mineros y grupos de minería.

- *Implicaciones:* Puede causar pérdida de ingresos y minería malintencionada.
- *Contramedidas: Enforce PoW submission.*

**Ataque timejacking:** Como la TBC es de naturaleza distribuida, todos los nodos deben estar sincronizados en fecha y hora, es por ello que cada nodo posee un contador interno. Un atacante podría adelantar o atrasar dicho reloj de *"sistema"* realizando diferentes conexiones suplantando a varios nodos y enviando una estampa de tiempo diferente e incorrecta en cada conexión. Afecta a los mineros, grupos de minería y aplicaciones.

- *Implicaciones:* Puede causar división de la cadena, pérdida de ingresos, minería malintencionada y retrasos.
- *Contramedidas:* Sincronización del reloj. (*Synchronized clocking*)

**Ataque Finney:** Se considera un tipo de ataque de doble gasto (véase sección 2.4.4.1 en la página 34) que ocurre cuando una persona acepta una transacción no confirmada en la red realizada por un usuario, que anteriormente transfirió dichos fondos a una dirección cómplice. El atacante difunde el bloque que contiene la primera transacción e inmediatamente se invalidará la segunda. Afecta a los mineros, grupos de minería y a los usuarios.

- *Implicaciones:* Puede causar pérdida de ingresos.
- *Contramedidas:* Incrementar la recompensa por bloque.

### Ataques a la aplicación Blockchain

***Blockchain ingestion:*** Se denomina así al proceso de análisis de una red BC para obtener información pública que puede resultar útil para un atacante. Afecta a la red.

- *Implicaciones:* Puede causar pérdida de información.
- *Contramedidas:* Blockchain encriptadas.

**Robo de *wallets*:** Este tipo de ataque incluye a todas las billeteras digitales o físicas, y el robo se puede materializar de múltiples formas. Afecta a los exchanges, aplicaciones y usuarios.

- *Implicaciones:* Puede causar pérdida de ingresos y robo.
- *Contramedidas:* Copias de seguridad y seguro de *wallets*.

**Doble gasto:** Descripción en la sección 2.4.4.1. Afecta a la red y a los usuarios.

- *Implicaciones:* Puede causar pérdida de ingresos y robo.
- *Contramedidas:* Utilización de esquemas *One-Time Settlement*, o liquidación única (OTS).

**Minería maliciosa (*Cryptojacking*):** Consiste en la utilización de sitios webs para realizar la computación necesaria en algoritmos como el PoW de forma ilegal y no autorizada. El tipo más común es la minería en el navegador, que utiliza la potencia computacional de los usuarios para resolver los cálculos necesarios. Afecta a las aplicaciones y usuarios.

- *Implicaciones:* Puede causar división de la cadena, robo y minería malintencionada.
- *Contramedidas:* Uso de *Mineguard* [Tah+17]

**DoS a contratos inteligentes:** Ataques de denegación de servicio dirigidos únicamente a contratos inteligentes. Afecta a la red, aplicaciones y usuarios.

- *Implicaciones:* Puede causar pérdida de ingresos, retrasos y robo.
- *Contramedidas:* Parchear EVM.

**Ataque de reentrada:** Tipo de ataque utilizado en el caso DAO. Se produce cuando el usuario no actualiza el saldo antes de enviar Ether, en ese caso, un atacante podría robar todo el Ether que posee el contrato inteligente haciendo llamadas recursivas al método *call.value()* en un *token* . Afecta a las aplicaciones y usuarios.

- *Implicaciones:* Puede causar pérdida de ingresos y robo.
- *Contramedidas:* Parchear EVM.

**Ataque de repetición:** Cuando se produce un fork *temporal* en una red BC, los usuarios mantienen los mismos recursos en las dos cadenas. El atacante escanea las transacciones y cuando un usuario realiza una transacción en una de las bifurcaciones, genera una copia en la otra, por tanto, el usuario perdería los fondos en ambas bifurcaciones. Afecta a la red, grupos de minería, aplicaciones y usuarios.

- *Implicaciones:* Puede causar pérdida de ingresos y robo.
- *Contramedidas:* Seguir los principios del desarrollo seguro.

**Ataque de desbordamiento:** Ocurre en los contratos inteligentes cuando el valor máximo de una variable es sobrepasado ($2^{256}$). Este ataque es muy raro que ocurra. Afecta a las aplicaciones y usuarios.

- *Implicaciones:* Puede causar robo.
- *Contramedidas:* Parchear EVM.

**Ataque de direcciones cortas:** Tipo de ataque comúnmente aplicable a *tokens* . El atacante debe tener una dirección que termine en 0 y realizar una compra con esa dirección omitiendo el último 0 de la dirección. Si el contrato tiene saldo suficiente, de cada 1000 *tokens* comprados, la EVM de Ethereum devolverá 256000 *tokens*. Afecta a las aplicaciones.

- *Implicaciones:* Puede causar pérdida de ingresos y robo.
- *Contramedidas:* Parchear EVM.

**Ataque de balance:** Este ataque tiene como objetivo a nodos con poder de minería en redes que utilizan PoW. El atacante retrasa la comunicación de red entre un subgrupo de nodos obligando a que se produzca un doble gasto. Afecta a las aplicaciones y usuarios.

- *Implicaciones:* Puede causar pérdida de ingresos y robo.
- *Contramedidas:* Seguir los principios del desarrollo seguro.

## 2.5. Posibles casos de uso

Son numerosos los autores que tratan el asunto de si es necesario o no utilizar la TBC para ciertos casos concretos. De entre todas las variantes que hay, Wust y Gervais [WG18] presentan un diagrama de flujo muy claro y conciso donde en pocos segundos el lector puede saber si realmente necesita o no una red BC para su producto/servicio y en caso afirmativo, cuál sería el tipo idóneo a utilizar (véase Fig. 2.10 en la página siguiente). Si simplificamos al máximo la idea de la BC, se podría decir que es una alternativa a las bases de datos tradicionales, puesto que se pueden "almacenar y leer datos". Es por ello que la mayoría de prototipos o pilotos que se realizan en la industria e incluso en la academia carecen de razones para utilizar TBC y no les aporta mucho más que una base de datos tradicional pero a un coste mucho más elevado. A continuación, se presentan algunos casos de uso donde sí puede tener cabida utilizar la TBC.

### 2.5.1. Sanidad

Existen numerosos autores que se centran en utilizar la TBC en el ámbito sanitario para mejorar los sistemas actuales aprovechando las características intrínsecas de la TBC. Algunos de ellos son Rabah [Rab17], Radanović y Likić [RL18] y Zhang *et al.* [Zha+18], estos últimos presentan algunos de los múltiples casos de uso en los que incluir la TBC en el ámbito sanitario.

- **Seguimiento de la preinscripción de opioides:** En EEUU existe una crisis sanitaria debido al uso indebido de sustancias opiáceas. Actualmente se hace un abuso tanto en el lado de los profesionales de la salud, con prescripciones innecesarias, como en el lado del paciente que sufre una mejora real a corto plazo a costa de aumentar su adicción a estas sustancias. Una posible solución sería establecer una red de confianza formada por hospitales y farmacias que almacene transacciones relativas a los opioides. Para los hospitales y farmacias el incentivo que existiría es el poder formar un conjunto de datos sobre opioides y usuarios de los mismos. Así mismo, los pacientes consiguen un aumento de la calidad de sus cuidados médicos gracias a este gran conjunto de datos y sus posibles aplicaciones, y además podrían estar más informados de los riesgos de estas sustancias.

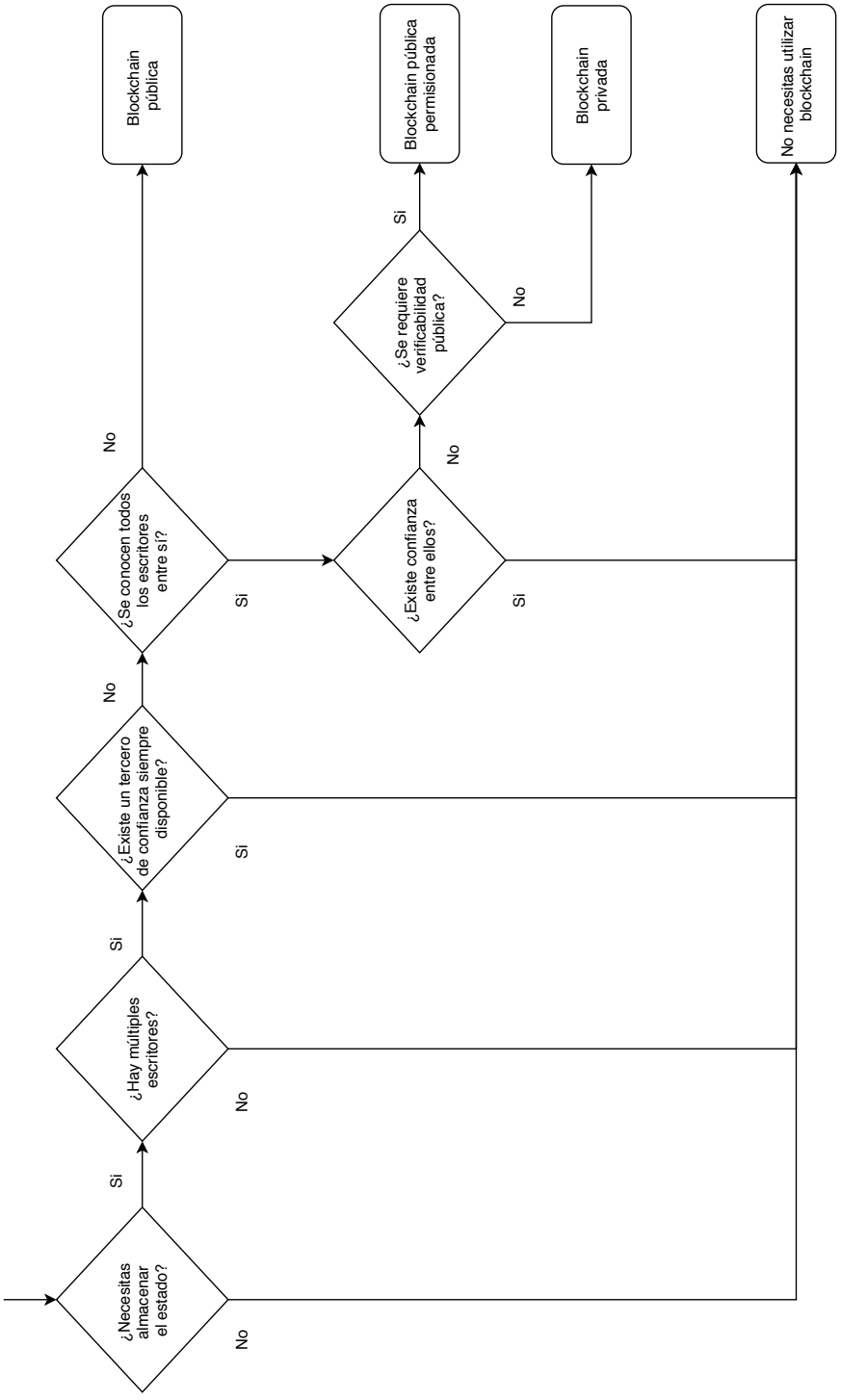


**Figura 2.10.-** Diagrama de flujo para la decisión de cuándo blockchain es necesario (adaptado de Wust y Gervais [WG18])

- **Intercambio de datos entre los cuidados tradicionales y la telemedicina:** La telemedicina está ganando popularidad entre los pacientes que necesitan un tratamiento menor pero urgente y por cualquier motivo, deben evitar las largas esperas en una consulta médica tradicional. Los datos recogidos y los tratamiento expedidos mediante este tipo de sistemas podrían ser inaccesibles para los médicos de atención primaria, lo que a su vez derivaría en un historial médico incompleto del paciente. Esto a su vez se traduciría, en general, en una disminución de la calidad de los futuros diagnósticos y tratamientos. Utilizando BC se consigue eliminar intermediarios que entorpezcan el intercambio de datos médicos y habilitar relaciones directas entre la telemedicina y los médicos de atención primaria. Un contrato inteligente se encargaría de orquestar los intercambios entre los diversos sistemas sanitarios.

- **Gestión de datos compartidos de los pacientes con cáncer:** La mayoría de pacientes que han sido diagnosticados con algún tipo de cáncer, desean tener una segunda opinión médica acerca del tratamiento o del diagnóstico. En la actualidad, este proceso involucra activamente al paciente (largos procesos burocráticos para obtener su historial médico completo y en muchos casos el de sus familiares cercanos), ya que debe obtener todos sus datos médicos y entregárselos al segundo facultativo. Este proceso no es nada recomendable teniendo en cuenta la situación en la que se encuentra el paciente. Una posible solución a este problema sería un sistema basado en BC que haga de intermediario entre los pacientes y los múltiples proveedores médicos de todo el mundo. Este sistema dotaría al paciente de total libertad para decidir qué datos médicos comparte y con quién.

- **Registro compartido del cáncer:** Continuando con el punto anterior, otro caso de uso interesante sería la creación de un registro distribuido y compartido sobre datos médicos de pacientes con algún tipo de cáncer. La compartición de datos sobre estas enfermedades es vital ya que los tratamientos son complejos y muy individualizados. La idea es construir un ecosistema de aprendizaje que utilizando BC permita compartir modelos predictivos, construidos utilizando los datos compartidos y que permitan a los profesionales médicos obtener pronósticos y mejorar el diagnostico y tratamiento final.

- **Registros sanitarios del paciente (PHRs):** En la actualidad existen numerosos registros médicos donde los pacientes pueden almacenar los resultados de diferentes pruebas. Algunos de estos ejemplos serían *Apple Health* o *Microsoft HealthVault*, que hacen uso de sistemas centralizados. Utilizando un sistema basado en BC se almacenarían los datos de forma distribuida y siendo el paciente el que controle qué datos compartir y con quién. A este sistema se podrían conectar el resto de sistemas sanitarios tradicionales, así como dispositivos, para realizar cualquier tipo de prueba y almacenar el resultado en la BC. Los pacientes tendrían un control total de sus datos, además de existir una traza inmutable que muestra cuándo se accedió a qué datos y quién accedió.

- **Adjudicación del reclamo del seguro médico:** Los clientes de aseguradoras médicas que tienen un accidente que deriva en una incapacidad permanente o incluso en la muerte, solicitan a su seguro una renta mensual, o una indemnización en el segundo caso. Aunque es cierto que hay un cierto porcentaje de fraudes, en la mayoría de los casos, se resuelve de forma automática. Con el fin de ampliar y asegurar dicha automatización, se propone el uso de contratos inteligentes que orquesten dicho proceso. Todo el proceso de adjudicación será transparente tanto para la aseguradora como para el cliente, mostrando los posibles errores y fraudes que podrían ser investigados de manera puntual. También se podrían utilizar estos acuerdos inteligentes para asegurar el conocimiento y actualización de las políticas actuales de la compañía.

### 2.5.2. Gobierno

La tecnología BC puede convertirse en un gran aliado para los gobiernos gracias a que les permitiría un mayor control y transparencia de las transacciones entre agencias gubernamentales y los ciudadanos, reducción del coste de operaciones, del número de fraudes y de los errores financieros. Dependiendo del Estado algunos de estos casos de uso podrían no ser factibles. Alketbi *et al.* [ANT18] presentan algunos de los casos de uso en este ámbito que pueden ser interesantes de cara a realizar algún proyecto piloto.

- **Gestión de la identidad y mantenimiento de registros:** Se propone un sistema BC que gestiona las identidades digitales de los ciudadanos que pueden ser utilizados en cualquier proceso burocrático gubernamental, documentos notariados o en cualquier servicio de un tercero. Entre las ventajas con respecto a un sistema tradicional se encuentran un control total de los datos por parte de los ciudadanos y la posibilidad de compartir determinados datos con terceras partes. Además estas terceras partes no tienen que almacenar ningún dato privado de los ciudadanos, reduciendo así los riesgos asociados y aumentando el cumplimiento.

- **Registro del valor:** En un sistema tradicional de registro de bienes debe existir una entidad central encargada de verificar todos los requisitos necesarios. Utilizando la TBC, se conseguiría eliminar dicha autoridad central, sustituyéndola por un modelo de verificación basada en usuarios que cumplan ciertos requisitos. Cualquier usuario identificado con su ID en el sistema BC podría añadir un bien con su correspondiente documentación asociada, firma y estampa de tiempo, y varios usuarios autorizados podrían verificar si dicha información es correcta.

- **Sistemas de votación:** Los gobiernos podrían hacer uso de la TBC para mejorar los sistemas de votación actuales. Esta tecnología proporcionaría mayor transparencia en todo el proceso, además de almacenar un registro inmutable del mismo. En un sistema de este tipo se podrían implementar diferentes soluciones como la delegación de votos, la votación de ideas propuestas por ciudadanos o diputados, modelos de elecciones aleatorias, etc.

- **Sistema sanitario:** Desde el punto de vista gubernamental, se podría utilizar la TBC para dotar de transparencia a todos los servicios sanitarios y sus costes asociados (véanse además de los casos de uso que se muestran en el subapartado anterior 2.5.1).

- **Ciudades inteligentes e IoT:** La sinergia de los ecosistemas IoT con la TBC da lugar a numerosos casos de uso. Entre ellos se encuentra la posibilidad de monitorizar y geolocalizar los activos IoT disponibles casi al instante, sin la intervención humana y sin posibilidad de modificación no deseada.

### 2.5.3. Vehículos inteligentes

En Singh y Kim [SK17] los autores proponen un *framework* para la compartición de datos entre vehículos inteligentes basado en BC, consiguiendo así un entorno de confianza entre los vehículos utilizando el algoritmo de consenso *Proof-Of-Driving* y en recompensas. Actualmente se utilizan diferentes redes ad-hoc para comunicar vehículos como pueden ser *WAVE* o *DSRC*, pero éstas no proporcionan la suficiente seguridad en la transmisión de datos. La seguridad puede aumentarse utilizando BC, gracias al uso que ésta hace de diferentes primitivas criptográficas como las vistas en la sección 2.2.1 en la página 10. El sistema propuesto ayuda a mejorar la privacidad de los vehículos inteligentes, proporciona comunicaciones rápidas y seguras entre los vehículos y permite detectar un histórico detallado de las conexiones entre vehículos. Si se produce un accidente en el que algún vehículo inteligente esté involucrado, todos los datos necesarios serán enviados a las autoridades competentes (servicios sanitarios, policía, etc.)

En Knirsch *et al.* [KUE17] los autores se centran en conseguir un sistema que garantice la privacidad de los usuarios de la red y proponen un sistema de subasta de energía en el que los usuarios publican su demanda y las estaciones de carga cercanas ofrecen una oferta para dicha demanda. En función de todas las ofertas recibidas, el usuario escogerá libremente su próxima estación de carga. Este sistema permite al usuario elegir la estación de carga más barata y/o cercana sin revelar su posición actual ni datos de su vehículo, solo se revelarán cuando haga uso de la estación de carga.

Y por último, cabe destacar el aporte de Dorri *et al.* [Dor+17] en el cual los autores proponen un sistema BC seguro y distribuido en un entorno con vehículos inteligentes. Aseguran la privacidad de los usuarios mediante claves públicas que deben ser renovadas y la seguridad con las propiedades inherentes a la TBC.

### 2.5.4. Sector financiero

Del trabajo de revisión en este campo realizado por Konstantinidis *et al.* [Kon+18] se pueden extraer los dos siguientes casos de uso más comunes:

- **Criptomonedas:** Las criptomonedas constituyen el primer caso de uso de la TBC y el que más peso tiene, hasta el momento. A día de hoy son casi innumerables las criptomonedas existentes en el mercado, cifra que aumenta con el paso del tiempo. Su principal objetivo es ser utilizadas como medio de pago electrónico

para cualquier tipo de bien o servicio. Por ejemplo, Kshetri [Ksh17] presenta un plataforma de comercio electrónico basado en Bitcoin que pone en contacto a compradores y vendedores sin cobrar una comisión por realizar compras o ventas en dichas plataformas. Además, la plataforma no recoge ningún dato del usuario, y en caso de sufrir un ataque que derive en una filtración de datos, no tendría consecuencias en los usuarios ya que no se almacenan su claves privadas.

- **Sistemas bancarios:** Algunos bancos han introducido la TBC en algunos de sus procesos consiguiendo una reducción del tiempo necesario para completar ciertas transacciones de los 2 días a apenas 10 segundos. Aunque únicamente se ha utilizado la TBC en una primera capa de transacciones bancarias, también se podría utilizar para facilitar intercambios y pagos en tiempo real mediante una red BC, en lugar de tener una entidad centralizada (banco).

En Sidhu [Sid17] se presenta una red BC llamada *Syscoin* basada en una criptomoneda propia que pretende ser una "segunda versión" del sistema Bitcoin planteado por Nakamoto [Nak09]. La plataforma se compone de un conjunto de contratos inteligentes que se utilizan para proporcionar soluciones de comercio electrónico basadas en BC para todo tipo de empresas. El principal objetivo que tienen es facilitar la adopción de la TBC a nivel empresarial mediante la minimización de las barreras de entrada de nuevas comunidades a la plataforma, baja inflación de su criptomoneda, alta seguridad, etc.

### 2.5.5. Agricultura

Bermeo-Almeida *et al.* [Ber+18] realizan una revisión sistemática de la TBC orientada a mejorar la agricultura actual y proponen dos casos de uso principales para la TBC en el sector agrícola.

- **Trazabilidad:** Se podría utilizar la TBC para conseguir un mercado de productos derivados de la agricultura más sólido y fiable, trazando los productos en todas sus etapas garantizando así una total transparencia de cara al consumidor. Si almacenamos todos los detalles de cada producto en todas sus etapas, aumenta la eficiencia de la cadena de suministros, mejorando tiempos de respuesta frente a imprevistos, como productos falsificados, contaminados o en mal estado. Además, sería posible obtener una traza completa y encontrar más fácilmente el origen del problema.

- **Datos de monitorización:** Otro posible caso de uso sería el almacenamiento de datos recogidos de los sensores situados en las plantaciones agrícolas. Con estos datos almacenados en la BC, aumenta la probabilidad de expansión a mercados internacionales cumpliendo con sus estándares, gracias a la transparencia y seguridad obtenida con este tipo de solución.

### 2.5.6. Otros casos de uso relevantes

Además de los casos de uso vistos anteriormente existen infinidad de aplicaciones de la TBC en numerosos sectores, algunos de los cuales se muestran a continuación.

En el área de estudio de la **gestión y análisis de datos**, Vo *et al.* [VKM18] proponen utilizar la TBC aprovechando las capacidades de los sistemas actuales, mejorando la protección de la información actual e integrando datos internos y externos a la BC. Esto lo combinan con un análisis de datos en un sistema de procesado de datos paralelo (como *MapReduce* o *Spark*) que posibilita el manejo de datos almacenados en una BC de manera eficiente.

Novotny *et al.* [Nov+18] proponen diferentes casos de uso en el ámbito de las **publicaciones académicas** utilizando la TBC, las más importantes de las cuales son:

- Gestión de confianza de material publicado.
- Citas y referencias de confianza.
- Proporcionar reputación, rendimiento e información sobre investigadores, instituciones, laboratorios, departamentos y equipos.
- Proveer reputación y confianza a nivel mundial de las fuentes de información y datos.
- Detectar y eliminar revistar depredadoras, conferencias y similares.
- Proceso de revisión por pares transparente.
- Gestión de la propiedad intelectual.

En el campo de la **logística y la cadena de suministros** la TBC tiene un gran potencial de cara a mejorar los sistemas actuales gracias a las ventajas propias de la tecnología como pueden ser la inmutabilidad, el no repudio o la transparencia de los datos y transacciones. Perboli *et al.* [PMR18], después de revisar numerosos casos de uso existentes, presentan la combinación entre la venta *online* de comida fresca y la TBC la cual proporciona beneficios tales como la mejora de la eficiencia en términos de capacidad de planificación y carga de trabajo, reducción de alimentos en mal estado causado por malas condiciones de almacenaje y por último, permite una mejora en la precisión del seguimiento de cada ingrediente para asegurar el cumplimiento de las regulaciones sanitarias.

Dentro del marco del **sector educativo**, Gräther *et al.* [Grä+18] presentan una plataforma educativa para gestionar certificados académicos (crear, revocar, verificar, etc.) haciendo uso de dos contratos inteligentes desplegados en la red pública Ethereum y almacenando cierta información en la red IPFS. El uso de la TBC posibilita la detección de una modificación no autorizada en cualquiera de los certificados de la plataforma.

Según la revisión sistemática del uso de la TBC en el **sector energético** realizada por Andoni *et al.* [And+19], se propone la utilización de la BC como una forma de

innovar permitiendo el intercambio de energía entre pares y la generación de energía descentralizada. Además, la utilización de esta tecnología puede suponer una gran mejora en los procesos habituales de este sector, mejorando procesos internos, servicios de cara al consumidor y reduciendo costes. Ofrece también un amplio abanico de posibilidades como son el rastreo del origen de la energía, carga de vehículos mediante *tokens* o la compra y venta de energía producida por usuarios de la red eléctrica que tienen infraestructura para generar energía sobrante, añadirla a la red y recibir algún tipo de remuneración.

También existen numerosas aplicaciones de la TBC usada junto con tecnologías existentes como son el *cloud computing* (Gai *et al.* [Gai+20; GCZ18]), las redes IoT (ver sección 3.1) o la inteligencia artificial (Nassar *et al.* [Nas+20] y Salah *et al.* [Sal+19]).

## 2.6. Otras tecnologías de registro distribuido

Hasta este punto nos hemos centrado en la tecnología Blockchain como mecanismo para almacenar de forma distribuida una secuencia de transacciones. En realidad Blockchain es una solución particular que implementa el concepto más general de *Distributed Ledger Technologies*, o tecnologías de registro distribuido (DLT). En ocasiones los términos DLT y TBC se utilizan casi como sinónimos, pero en rigor no son lo mismo, sino que las DLTs engloban a las TBCs.

En esta sección se explica someramente el concepto más genérico de DLT y se verá otra posible implementación conocida como "grafos acíclicos dirigidos" (DAG).

Se denomina *Distributed Ledger Technologies*, o tecnologías de registro distribuido (DLT) a toda aquella tecnología que almacene registros o información de forma distribuida entre todos sus usuarios, ya sea de forma pública o privada. Se diferencian de las bases de datos tradicionales en que los datos no se almacenan de forma centralizada, sino que todos los participantes de la red tienen una copia de la información, y además todas las transacciones son verificadas y confirmadas por toda la red.

La característica principal de las DLT radica en los algoritmos de consenso, utilizados para que todos los miembros de una red (maliciosos o no) se pongan de acuerdo en almacenar la misma información, además de garantizar su inmutabilidad ya que si un nodo realiza un cambio no autorizado, el resto de la red lo detectaría y descartaría. Su principal motivación es resolver el problema del *consenso ante fallos bizantinos* propuesto por Lamport *et al.* [LSP82].

Según Burkhardt *et al.* [BWL18] existen dos tipos de arquitecturas mayoritarias que se pueden considerar DLT. El primer tipo es Blockchain que ha sido ya suficientemente explicado en las secciones previas de este capítulo, y el segundo tipo es conocido como Grafos Acíclicos Dirigidos o DAGs, cuyas características básicas veremos en el siguiente apartado.

### 2.6.1. Grafos acíclicos dirigidos (DAGs)

Un DAG es un grafo que tiene las siguientes características: es **dirigido** (porque los arcos tienen dirección, es decir no es el mismo arco el que va del nodo $u$ al $v$ que el que va de $v$ a $u$) y **acíclico** (no existen ciclos, es decir, dado un vértice $v$, no existe ningún camino $c$ que empiece en $v$ y termine en $v$). Aunque en realidad la cadena de bloques de BC encaja en esta definición y por tanto es un tipo de DAG, la inversa no es cierta, es decir, no todo DAG es una cadena de bloques.

La red más importante que utiliza este tipo de DLT, es Nano, y en ella se definen 4 componentes básicos:

- **Cuenta:** o también llamada dirección (*address*), es la clave pública de una pareja de claves público-privada. Un usuario puede tener varias cuentas.
- **Bloque o transacción:** Se usan indistintamente ya que (a diferencia de BC) un bloque solo contiene una transacción. Las transacciones están firmadas con la clave pública de la cuenta que emite dicha transacción.
- **Registro (ledger):** Es el conjunto de todas las cuentas que existen en la red. Cada cuenta a su vez tiene asociada una cadena de bloques propia y "privada".
- **Nodo:** (no confundir con el concepto de nodo en un grafo) Es el software que gestiona el registro y las cuentas que tenga el usuario. Un nodo puede almacenar todo el registro completo o solo un los últimos bloques de la cadena de cada cuenta. Se ejecuta en una computadora.

Nano sigue una arquitectura de tipo "enrejado de bloques" (*block-lattice*), en la cual cada cuenta tiene su propia cadena de bloques que contiene su balance histórico y además puede actualizarse de forma asíncrona con el resto de la red.

La transferencia de fondos entre cuentas requiere 2 transacciones. La primera de ellas será el envío de la cantidad deseada, que se restará automáticamente de la cuenta del emisor y la segunda transacción deberá ser una confirmación del receptor. El nodo del receptor deberá estar activo para poder recibir la transacción.

En la figura 2.11 está representada la estructura de la red Nano con un *ledger* de 5 cuentas, las cuales han sido creadas a partir de la cuenta génesis de la red con el envío de una transacción abierta (*open transaction*). A partir de ese momento cada cuenta tiene asociada una cadena de bloques que está representada por todos los bloques encadenados dentro del rectángulo gris punteado. Los bloques azules representan el último bloque de dicha cuenta, a partir de los cuales se muestra una línea temporal con las transacciones enviadas y recibidas por cada cuenta. Esta parte de la figura sería el *Directed Acyclic Graph*, o grafo acíclico dirigido (DAG) propiamente dicho, en el que los nodos son las transacciones.

Esta red utiliza *Proof of Work* (PoW) para prevenir que un usuario con malas intenciones pueda generar excesivas transacciones. A diferencia de la TBC, en este tipo de DLT no hay necesidad de elegir un nodo que mine la transacción, ya que cada cuenta tiene la obligación de mantener las transacciones ordenadas en su propia cadena de

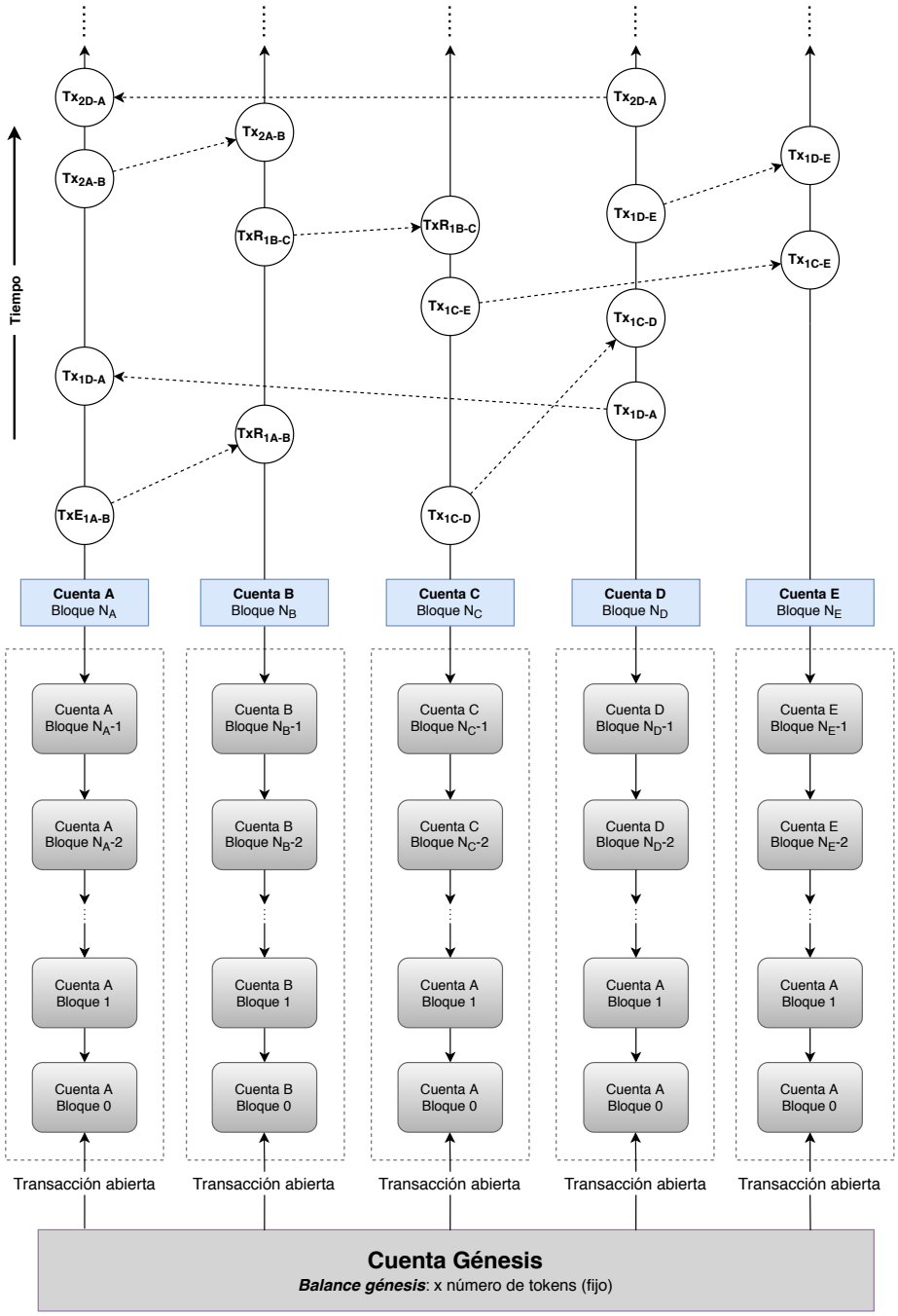


**Figura 2.11.-** Representación de la red Nano con un *ledger* de 5 cuentas.

bloques. Si hay conflicto, se soluciona utilizando lo que denominan *"sistema de representantes"*. Cuando se crea una cuenta nueva en la red se debe elegir un representante que se puede modificar más adelante si lo desea. Los representantes votan para resolver

los conflictos y el voto de cada representante tendrá el peso en función de la suma de todos los saldos de las cuentas a las que "representa". Además, si un representante ve una transacción nueva, si es válida la reenvía con una firma de voto adjunta, para que siga circulando por la red y se confirme.

Capítulo 3

# Estado del arte sistemático

El estado del arte (del inglés, *State Of Art*) se define como el estado máximo alcanzado en la investigación sobre un tema en concreto. Según Dochy [Doc06], un estado del arte considera principalmente la investigación más reciente en un cierto área o tópico. A menudo resume las tendencias actuales o emergentes. Puede ofrecer nuevas perspectivas sobre un tema o señalar un área que requiera de mayor investigación.

Una revisión de la literatura científica puede ser de dos tipos: narrativa o sistemática. Sanchez-Meca [San10] define la revisión sistemática (RSL) como: *"un tipo de investigación científica mediante la cual se revisa la literatura científica sobre un tópico partiendo de una pregunta de forma clara y objetiva, utilizando métodos sistemáticos y explícitos para localizar, seleccionar y valorar críticamente las investigaciones relevantes a dicha pregunta y aplicando protocolos sistemáticos para la recogida de datos e información de dichas investigaciones, con el objetivo de alcanzar conclusiones válidas y objetivas sobre qué es lo que dicen las evidencias sobre dicho tópico."* Este es el tipo de revisión realizada en el presente trabajo.

Además, como sugieren numerosos autores en diversas publicaciones, como por ejemplo, Neto *et al.* [Net+19] o Garousi *et al.* [GFM16], en el contexto de las ciencias de la computación es interesante realizar un tipo de RSL llamada *Revisión Multivocal de la Literatura*, en el cual a la literatura publicada en revistas científicas (literatura *blanca*) se añaden otros recursos en linea como blogs, vídeos, *white papers*, etc. (literatura *gris*). Este tipo de revisiones son muy útiles sobre todo en campos como la TBC, que avanza muy rápido, pues nos ofrece una visión de dicha tecnología en la industria, aportando nuevos procedimientos, casos de uso o teorías que tienen una aplicación práctica y no suelen materializarse en un artículo científico revisado por pares.

## 3.1. Revisión de estados del arte previos

### 3.1.1. Objetivo y visión general

Antes de comenzar la revisión sistemática multivocal se ha realizado un estudio de las revisiones o *surveys* recientemente publicados con el fin de obtener una visión previa (a nuestra revisión) aportada por diferentes investigadores con experiencia en el campo. Este estudio previo nos permitirá seleccionar los tópicos más interesantes sobre los que

centrar nuestra propia revisión sistemática.

Para ello se ha hecho una búsqueda a través de Google Scholar con palabras clave como "blockchain", "review" y "survey". Se han descartado las que hubieran sido realizadas años atrás, quedándose por tanto obsoletas y no aportando nada nuevo en comparación con el resto de revisiones publicadas pocos meses después de ser realizadas. Las que han pasado el filtro se han estudiado para determinar su campo de aplicación, detectándose que la mayoría de ellas son bastante generalistas, aunque algunas se centran con más énfasis en aspectos de seguridad, y unas cuantas estudian el caso particular de la aplicación de las TBC a la *Internet of Things*, o Internet de las cosas (IoT) (*Blockchain-Based Internet of Things* (BIoT)), que parece ser un campo de aplicación que está surgiendo con gran fuerza. La mayoría de estos artículos incluyen lo que los respectivos autores consideran los "retos" más importantes de esta tecnología. La tabla 3.1 en la página siguiente resume visualmente toda esta información, mostrando los retos detectados por estas publicaciones, mientras que la subsección siguiente explicará con más detalle el contenido de cada artículo revisado.

En la tabla se observan unos cuantos hechos relevantes:

- Los aspectos de privacidad y seguridad son muy importantes, y la mayoría de los estudios coinciden en ello.
- La escalabilidad de la BC es también un tema muy importante, especialmente en conjunción con IoT dado que en esta aplicación el número de datos a guardar en la cadena puede ser muy grande, así como el número de nodos que participan en la misma.
- La escalabilidad es en el fondo un concepto de "alto nivel", que está constreñido por otros aspectos de más bajo nivel, como ya vimos en la sección 2.4.3. Varios de estos aspectos también son señalados como retos importantes, en concreto:
  - El almacenamiento de grandes volúmenes de información en la cadena
  - El rendimiento (mejora de la latencia y aumento del *throughput*)
- Varios de los estudios concuerdan en que es importante modificar la estructura de la BC o sus algoritmos de consenso, ya sea para mejorar su eficiencia energética y sostenibilidad, para una mejor integración de las diferentes redes y aplicaciones o para evitar la centralización en la que fácilmente puede caerse.

  La modificación de los protocolos de la BC lleva parejo el problema de la gobernanza, ya que estos cambios afectan a muchas partes interesadas y no está claro el mecanismo por el cual debe llegarse a un acuerdo sobre cómo llevar a cabo estas modificaciones de forma consensuada. En particular el problema de la gobernanza es identificado como tal por varios de los artículos (excepto los que se centran en IoT donde por alguna razón este problema no parece ser tan relevante).
- Finalmente otro reto que es mencionado en varios de los artículos es el del tratamiento de los grandes volúmenes de información almacenados en la BC, ya sea con

| | Campo | | | Retos | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Artículo | Generalista | Seguridad | IoT | Privacidad | Seguridad | Autenticación | Vulnerabilidades | Escalabilidad | Almacenamiento | Rendimiento (Latencia y Throughput) | Eficiencia energética/sostenibilidad | Infraestructura blockchain | Gobernanza | Mejorar compatibilidad/interoperabilidad | Plataforma de test estandarizada | Dificultad Análisis Big Data | Uso de IA |
| Fernández-Caramés y Fraga-Lamas [FF18] | ○ | ○ | ● | ● | ● | ○ | ○ | ○ | ○ | ● | ● | ○ | ○ | ○ | ○ | ○ | ○ |
| Panarello *et al.* [Pan+18] | ○ | ○ | ● | ○ | ● | ● | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| Zheng *et al.* [Zhe+18] | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ● | ○ | ● | ● | ● |
| Wang *et al.* [Wan+19] | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ● | ○ |
| Hasanova *et al.* [Has+19a] | ○ | ● | ○ | ○ | ● | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● |
| Feng *et al.* [Fen+19] | ● | ○ | ○ | ● | ○ | ○ | ○ | ● | ○ | ○ | ● | ○ | ○ | ● | ○ | ○ | ○ |
| Hassan *et al.* [Has+19b] | ● | ○ | ○ | ○ | ○ | ○ | ○ | ● | ● | ● | ● | ○ | ● | ○ | ○ | ○ | ● |
| Casino *et al.* [CDP19] | ● | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ● | ○ | ○ | ○ |
| Dai *et al.* [DZZ19] | ○ | ○ | ● | ● | ● | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ |
| Butijn *et al.* [BTH19] | ● | ○ | ○ | ● | ○ | ○ | ○ | ○ | ● | ● | ○ | ○ | ● | ○ | ○ | ○ | ○ |
| Sengupta *et al.* [SRD20] | ○ | ● | ● | ● | ● | ● | ○ | ● | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| Di Francesco Maesa y Mori [DM20] | ● | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ |
| Islam *et al.* [Isl+20] | ● | ● | ● | ○ | ● | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| Belchior *et al.* [Bel+20] | ● | ○ | ○ | ● | ● | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ |
| Wang *et al.* [Wan+20c] | ○ | ○ | ● | ● | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ● | ○ | ○ |
| Huang *et al.* [Hua+20] | ● | ○ | ○ | ● | ● | ○ | ○ | ● | ○ | ● | ○ | ○ | ○ | ● | ○ | ● | ○ |
| Bhushan *et al.* [Bhu+20] | ○ | ○ | ● | ● | ● | ○ | ○ | ○ | ○ | ● | ● | ● | ○ | ○ | ○ | ○ | ○ |
| Hunhevicz y Hall [HH20] | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ |

**Tabla 3.1.-** Matriz de análisis de estados del arte previos

técnicas de *Big Data* o de inteligencia artificial (esta última más bien relacionada con aspectos de seguridad y detección de vulnerabilidades)

En resumen, la conclusión que he obtenido de la lectura de estos estudios es que los tópicos de investigación más importantes señalados en los mismos serían: almacenamiento de grandes volúmenes de datos, y la privacidad de los mismos, mejora de la eficiencia de la red, en latencia y rendimiento, y mejora de los mecanismos de gobernanza para posibilitar modificaciones en los protocolos. En la sección 3.2 en la página 71 partiré de estas conclusiones para realizar mi propia revisión sistemática que actualice el estado del arte en estos tópicos.

### 3.1.2. Resúmenes de los estudios consultados

En lo que sigue se enumeran y resumen brevemente los estudios consultados, que son los ya mostrados en la tabla 3.1 en la página anterior, en orden cronológico de publicación. Cabe señalar que la mayoría de estos estudios no especifican qué metodología han seguido en su realización, salvo unos pocos que sí han seguido una metodología sistemática como la descrita en la introducción de este capítulo.

- En mayo de 2018 se publica la revisión de Fernández-Caramés y Fraga-Lamas [FF18] en la cual se revisa la literatura correspondiente a la integración de la TBC en redes IoT para dar como resultado las aplicaciones que definen como "*Blockchain-Based Internet of Things* (BIoT) *applications*". En este estudio los autores elaboran una lista de condiciones que debería cumplir una aplicación IoT existente para decidir si es apropiado integrar en ella la TBC. Las condiciones son: descentralización, intercambios P2P, sistema de pagos, registro público y secuencial de transacciones, sistema distribuido robusto y recopilación de *micro-transacciones*. También incluyen un diagrama de flujo para obtener el tipo de red BC que más se adecua a nuestra aplicación IoT.

  Entre las posibles aplicaciones utilizando esta sinergia están: sector agrícola, transacciones financieras, detección de multitudes, ciudades inteligentes, salud, logística, monitorización y gestión de una flota de vehículos, objetos inteligentes, transporte, energía, gobierno y democracia, telecomunicaciones y defensa y seguridad pública.

  Analizan en el estudio las arquitecturas típicas en IoT:

  - *"Cloud-based"*: Varias vulnerabilidades: punto único de fallo (la nube), posibilidad de espionaje de los datos privados o la alteración de los datos recogidos por el dispositivo. No es la arquitectura más adecuada para añadir TBC.
  - *"Fog y Edge computing-based"*: Estos dos tipos de arquitectura son más recientes y realizan parte del procesamiento de los datos dentro de la red, sin tener que enviarlos a la nube. Esto permite un aumento de la escalabilidad de la red, una baja latencia y los dispositivos pueden estar distribuidos físicamente.

  Realizan también un mapa con los principales factores que condicionan el desarrollo de aplicaciones BIoT. Algunos de ellos son: seguridad y privacidad, tamaño de la cadena, infraestructura, minado, gestión de multi-cadenas, usabilidad, latencia, tasa de transferencia efectiva (*throughput*), consumo energético, ancho de banda, etc..

  Los autores identifican los siguientes retos a los que se enfrenta la investigación en este ámbito:

- **Privacidad**: Además de los problemas de privacidad conocidos en las redes BC ya estudiadas, se añaden problemas propios de las redes IoT, como pueden ser la posibilidad de extraer datos privados de dispositivos IoT. Entre las posibles soluciones a estos problemas de privacidad se encuentran el despliegue de una red BC privada adicional que se encargue de monitorizar todas las acciones realizadas, utilizar criptografía homomórfica o generar claves diferentes con cada transacción con el fin de mantener el anonimato. Todas estas soluciones requieren mayor capacidad computacional de la que poseen la mayoría de dispositivos IoT actuales.
- **Seguridad**: Cualquier sistema informático debe cumplir 3 requisitos básicos para garantizar su seguridad que son la disponibilidad, la confidencialidad y la integridad. La disponibilidad está asegurada gracias a las características que aporta la TBC. La confidencialidad está asegurada para un usuario cualquiera si gestiona correctamente sus claves pública-privada, y algunas soluciones en este ámbito proponen una doble autenticación. Por último, aunque una de las características principales de la TBC es la inmutabilidad esto no es del todo cierto, ya que en la práctica la mayoría de redes BC están controladas por grupos u organizaciones que podrían alterar la integridad de los datos.
- **Eficiencia energética**: Es uno de los aspectos clave a tener en cuenta en las redes IoT, ya que la mayoría de los dispositivos que forman estas redes se alimentan con pequeñas baterías. Para reducir este consumo, se deben utilizar *"mini-blockchains"* que mantengan almacenadas únicamente las últimas transacciones. Además, el algoritmo de hashing utilizado por referencia es *SHA-256* pero existen múltiples algoritmos más rápidos y que conllevarían un menor consumo eléctrico, como son *Scrypt* o *X11*.
- **Latencia y throughput**: La red BC debe ser capaz de soportar grandes volúmenes de transacciones por unidad de tiempo en una escala en la que pocas redes BC actuales podrían trabajar.
- **Tamaño de la red, ancho de banda**: Con el paso del tiempo, aumenta el tamaño de la cadena de bloques que tienen que almacenar los nodos. Actualmente los dispositivos IoT no poseen la capacidad de almacenar más que una pequeña parte de la cadena, siendo necesario explorar nuevas técnicas de compresión u otras alternativas como utilizar una *mini-Blockchain* [véase Bru14] la cual está formada por un árbol de cuentas que almacena el estado actual de cada usuario. Sólo se almacena la transacción más reciente de cada usuario y la cadena de bloques sólo aumentaría su tamaño cuando se añaden nuevos usuarios a la red. Respecto al ancho de banda, al estar éste bastante limitado en redes IoT, el tamaño de los bloques y transacciones deberán ser ajustados.
- **Otros retos relevantes:** Además destacan otros retos que se deben tener en cuenta como el ratio de adopción, la usabilidad, la gestión de varias redes BC, las versiones y los *forks*, etc.

Además de lo antes visto, los autores incluyen una serie de recomendaciones y desafíos por resolver que atañen a la TBC en general, como los problemas de escalabilidad, limitaciones en los algoritmos de consenso, limitaciones a la hora de implementar contratos inteligentes o desarrollar métodos que eviten la centralización de redes BC, la necesidad de una mayor interoperabilidad y estandarización, así como una infraestructura BC que cumpla todos los requisitos de las aplicaciones BIoT y que pueda ser utilizada como "base" o referencia para este tipo de proyectos.

Los autores concluyen que las aplicaciones BIoT todavía están en una etapa muy primitiva, y aunque sí existen los primeros desarrollos y despliegues de este tipo de aplicaciones, es necesario abordar las demandas específicas de este campo con tecnología adicional, colaboración entre distintas partes interesadas, como gobiernos, organizaciones, etc., para llegar a un uso más amplio y en el que realmente la TBC aporte valor.

- En agosto de 2018 Panarello *et al.* [Pan+18] publican un estudio sistemático sobre la integración de IoT con la TBC. Identifican las principales debilidades de las redes IoT donde la TBC puede ser de ayuda, como son los ataques (p. ej. DDoS), los puntos únicos de fallo (p. ej. la nube), la confidencialidad, seguridad e integridad de los datos, la autenticación, la disponibilidad y el no repudio.

  Señalan numerosos retos utilizando la red de Bitcoin como base para identificarlos. El primer problema es el tamaño de la cadena de bloques que se incrementa cada día y el número de transacciones por segundo (en el caso de Bitcoin se crea un nuevo bloque cada 10 minutos, aproximadamente). Los autores señalan que si no se soluciona este problema la red podría empezar a caer en desuso. Las soluciones que proponen son reducir el tamaño del bloque separando los datos de su firma digital o incrementar el tamaño del bloque, aumentando así el número de transacciones por bloque. Relacionado con este problema también está la escalabilidad, que se puede observar con el tamaño de los datos, el tiempo de respuesta o el coste.

  Concluyen que es necesario desarrollar una solución que garantice la privacidad e integridad de los datos y diseñar un sistema capaz de gestionar la identidad única de cada dispositivo de forma que no se pueda alterar. Además, señalan que la investigación de la integración de IoT con TBC está en fase incipiente e indican otras posibles vías futuras de investigación como son *smart homes*, *smart cities*, *smart energy*, análisis sobre las posibles amenazas que pueden sufrir este tipo de sistemas y la construcción de sistemas completos de IoT con TBC en los cuales sus componentes sean estandarizados y fácilmente eliminados o añadidos al sistema.

- En octubre de 2018, Zheng *et al.* [Zhe+18] proporcionan una visión de la estructura que posee una red BC genérica, revisan algoritmos de consenso y aplicaciones BC, además de discutir los retos a los que se enfrentaba la TBC en ese momento

y recopilar los avances que se habían logrado de cara a combatir estos retos.

En concreto, analizan los algoritmos de consenso PoW, PoS, PBFT, dPoS, *Ripple*, *Tendermint*, los comparan en una tabla y muestran los avances conseguidos en este campo presentando varios algoritmos de consenso que mejoran algunas de las deficiencias de algoritmos de consenso tradicionales. Apuntan que un buen algoritmo de consenso debe ser eficiente, seguro y conveniente al escenario donde se utilizará. En cuanto a las aplicaciones BC, las agrupan en 5 categorías: Servicios públicos y sociales, Sistemas de reputación, seguridad y privacidad, IoT y financieras. Identifican 3 retos principales: la escalabilidad (detectan como vías de investigación la optimización del almacenamiento y un rediseño de la TBC), las fugas de privacidad (dos tipos de mecanismos para mejorar el anonimato: *mixing* y mecanismos de anonimato como *Zero-Knowledge Succinct Non-Interactive Argument of Knowledge* (zk-SNARK), donde los fondos de los usuarios y la cantidad de cada transacción están ocultos) y la *Selfish mining* donde destacan el trabajo de Solat y Potop-Butucaru [SP16], donde los bloques generados y aceptados por la red tienen un tiempo máximo y no pueden conseguir mayor recompensa reteniendo bloques.

Como trabajo futuro proponen crear un estándar en las pruebas realizadas a las redes BC para evitar la manipulación intencionada de los resultados y atraer así a más inversores, mejorar la gobernanza para terminar con la centralización intencionada de algunas redes BC, como Bitcoin, donde solo 5 *pools* de minería controlan toda la red, la combinación con Big Data (gestión de datos y analítica de datos), mejoras en el rendimiento de los contratos inteligentes actuales y la integración con aplicaciones que utilizan Inteligencia Artificial.

- En enero 2019 Wang *et al.* [Wan+19] publicaron un artículo centrándose en los nuevos avances en algoritmos de consenso diseñados especialmente para redes públicas o sin permisos (véase 2.3.1 en la página 28).

  Muestran diferentes estrategias de minería egoísta como son *Selfish Mining*, *Block withholding*, *Lie-in-wait* y *Pool hopping*, revelando los riesgos tanto para la red, como para los mineros honestos y las posibles soluciones. Diferencian entre 3 tipos de algoritmos de consenso para redes públicas: los basados en *PoX puzzles*, minería virtual (*virtual mining*) y protocolos híbridos. Respecto a las direcciones futuras que proponen en este aspecto son reducir el coste de la descentralización y mejorar el soporte para realizar una computación Big Data de forma segura.

- En enero de 2019, Hasanova *et al.* [Has+19a] publicaron un artículo en el que, después de clasificar los tipos de redes BC existentes en 3 categorías según su organización y accesibilidad, presentan los ataques posibles a los que se enfrenta cada tipo de red de forma genérica y proponen diferentes contramedidas que pueden ayudar a evitar o minimizar los ataques.

  Apuntan que la mejor garantía para evitar ataques es tener el *software* de los nodos siempre actualizado. El problema es que esto crearía un *hard-fork* hasta

que todos los nodos de la red tengan la misma versión. Los esfuerzos de los investigadores y desarrolladores se centran en la combinación de la TBC con tecnologías como el aprendizaje automático o las redes neuronales. De este modo se podría predecir la presencia de un nodo malicioso en la red en función de su comportamiento u optimizar los métodos actuales de "data *sharding*" o incluso utilizar la IA para optimizar los grupos de mineros (*mining pools*) y reducir el coste computacional necesario para minar un nuevo bloque (en algoritmos como PoW).

- Ese mismo mes, Feng *et al.* [Fen+19] publicaron un artículo donde analizan las amenazas referentes a la privacidad a las que se enfrenta la TBC y discuten los mecanismos criptográficos de defensa existentes. Para que una red BC cumpla con los requisitos de privacidad debe asegurar la privacidad tanto de la identidad de los usuarios, como de las transacciones.

  Revisan las diferentes metodologías para asegurar la privacidad de la identidad de los usuarios de la red (aunque no explican con qué criterio seleccionan la literatura), dando lugar a una tabla resumen con múltiples coin mixers, tanto centralizados como descentralizados. Además de los coin mixers, también se puede utilizar la firma en anillo o *"ring signature"*, ideada por Rivest *et al.* [RST01]. Ésta implica que cualquier usuario del anillo puede firmar una transacción, pero sin revelarse que usuario la firmó. Por último, también destacan la *Non-interactive Zero-Knowledge proof*, o prueba de conocimiento-cero no interactiva (NIZK) propuesta por Blum *et al.* [BFM88], mediante la cual se puede verificar la legitimidad de una transacción sin revelar ningún tipo de información adicional.

  En lo referido a garantizar la privacidad de las transacciones sugieren dos enfoques: la NIZK, al igual que en la privacidad de la identidad, y utilizar un sistema basado en criptografía homomórfica destacando dos de ellos, el esquema de compromiso de Pedersen [Ped92] y el sistema Paillier [Pai99].

  Para terminar los autores sugieren diferentes direcciones hacia las que se puede orientar la investigación en este campo: la escalabilidad a un coste "asumible", preservar la privacidad eliminando algunos supuestos actuales de confianza, mayor compatibilidad entre los sistemas desarrollados para mejorar algún aspecto anterior y la estructura de transacciones de cada red, y por último, la trazabilidad legal, es decir, que sólo las autoridades competentes puedan trazar el rastro de transacciones de algún usuario malicioso en caso de ser necesario, pero siendo imposible para el resto de usuarios.

- En febrero de ese mismo año, Hassan *et al.* [Has+19b] publican un artículo en el que estudian las aplicaciones de red basadas en TBC y discuten los retos a los que se enfrentan, que son la aplicabilidad, la sostenibilidad y la escalabilidad.

  Después de revisar todos los artículos obtenidos, identifican diferentes retos (trabajo futuro) a los que se enfrenta la TBC entre los que se encuentran la gobernanza y los problemas operacionales y regulatorios asociados como son el Reglamento

General de Protección de Datos (RGPD) o el derecho al olvido. También destacan la escalabilidad, donde la analizan desde el punto de vista del *throughput* y del almacenamiento. Por último, también estudian la usabilidad y gestión de claves, el anonimato, el uso de la inteligencia artificial y el *machine learning* en combinación con la TBC y las preocupaciones actuales en términos de seguridad y privacidad.

- En marzo 2019, Casino *et al.* [CDP19] publicaron una revisión sistemática en la cual utilizan la metodología propuesta por Briner y Denyer [BD12] y algunos aspectos de la declaración PRISMA propuesta por Moher *et al.* [Moh+09] para realizar una revisión científica, transparente y fácilmente reproducible. También tienen en cuenta la literatura gris. Analizan un total de 260 artículos científicos y 54 resultados de literatura gris publicados entre 2014 y 2018. Las categorías principales en las que se enmarcan las aplicaciones de la TBC son la financiera, gestión de datos, industria y negocio, privacidad y seguridad, educación, salud, IoT, gobernanza o integridad de los datos. Identifican las características (escalabilidad, privacidad, interoperabilidad, auditabilidad, latencia o visibilidad) necesarias para cada tipo de categoría.

  Destacan como trabajo futuro o aspectos a mejorar en este campo, la idoneidad de la TBC (solo utilizar TBC en los casos en los que sea realmente útil y aporte valor), latencia y escalabilidad, sostenibilidad (reducir el gasto energético y aplicar la potencia computacional al procesado de datos), resiliencia cuántica[1], adopción de la TBC e interoperabilidad y soluciones para la gestión de datos de forma segura y respetando la privacidad.

- En octubre de 2019 Dai *et al.* [DZZ19] publican un estudio en el que introducen la integración de TBC con IoT y proponen una arquitectura para este tipo de sistemas. Identifican las características propias de combinar IoT y TBC:

  · **Interoperabilidad** entre dispositivos o sistemas IoT y sectores industriales, que se consigue mediante la capa llamada "Blockchain-composite" que se posiciona en la cima de la red P2P con acceso uniforme a diferentes sistemas IoT.

  · **Trazabilidad de los datos:** Cada bloque almacenado en la cadena incluye una estampa de tiempo histórica para asegurar la correcta trazabilidad de los datos, siendo posible ubicar cada transacción almacenada de forma espacial y temporal.

  · **Fiabilidad de los datos:** Es asegurada gracias a las primitivas criptográficas asociadas a la TBC, como son la criptografía asimétrica, las funciones *hash* y la firma digital.

  · **Interacciones autónomas:** Se definen como la capacidad de interacción de un sistema IoT con otro sistema de cualquier tipo sin la necesidad de un

[1] **Resiliencia cuántica:** Grado de resiliencia de la TBC frente a los avances de la computación cuántica que puedan suponer la invalidez de algoritmos criptográficos utilizados en dicha tecnología.

tercero de confianza. Esto se puede conseguir utilizando contratos inteligentes.

Entre los retos abiertos de las redes IoT identifican la heterogeneidad de sistemas IoT, la complejidad de las redes, la escasa interoperabilidad, las limitaciones de los dispositivos IoT, posibles vulnerabilidades de privacidad y seguridad.

Además estudian las aplicaciones propuestas en la literatura para este tipo de sistemas *Blockchain-Based Internet of Things* (BIoT) y las organizan en las siguientes categorías:

- **Fabricación inteligente:** Mejora la interoperabilidad, reduce los costes al prescindir de un tercero de confianza y automatizan los *P2P business trading* (comercio directo entre pares).
- **Gestión de la cadena de suministro:** Aseguran la procedencia de los datos, reducen los costes de los servicios post-venta y mitigan los riesgos tradicionales de las cadenas de suministro.
- **Industria alimentaria:** Mejora la trazabilidad de los datos y la seguridad alimentaria.
- **Red inteligente ("Smart grid"):** Mejora la transparencia, asegura la privacidad y permite el intercambio seguro de energía.
- **Salud:** Mejora la seguridad, garantiza la privacidad y permite verificar la autenticidad.
- **IOV y UAV (Internet de los vehículos y autotripulados):** Asegura la confiabilidad de los mensajes, intercambio de energía entre vehículos eléctricos de forma segura y garantiza la confidencialidad mutua entre UAV.

Además, identifican los siguientes retos abiertos para la "*Blockchain of Things* (BCoT)":

- **Recursos limitados:** La mayoría de dispositivos IoT tienen almacenamiento limitado, poca batería, mínima capacidad computacional y capacidad de conectividad reducida. Los principales problemas generados al añadir TBC en este tipo de redes son la dificultad para llegar a un consenso y el gran almacenamiento que necesitan los nodos. Entre las posibles soluciones se encuentra incorporar tecnologías como MEC o *cloud computing* que podrían servir como almacén de datos a los nodos de la red.
- **Vulnerabilidades:** Aunque sólo por el hecho de añadir TBC se mejora la seguridad, ésta sigue siendo una línea a tener muy en cuenta en futuras investigaciones. Entre las vulnerabilidades destacan la posibilidad de sufrir ataques de espionaje (*eavesdropping*), interferencias (*jamming*), o ataques de *replay* (*replaying attacks*). Relacionado con el primer punto de esta lista, no

es factible implementar algoritmos de cifrado muy costosos a nivel computacional ya que los recursos de los dispositivos son limitados. También destacar la gestión de claves en un entorno distribuido, ataques al protocolo de enrutamiento BGP y las vulnerabilidades que pueden presentar los contratos inteligentes. Entre los avances en dar solución a estos problemas destacan soluciones que mejoran la seguridad de sistemas IoT sin utilizar *hardware* adicional. Otros autores proponen un sistema de generación de claves basado en la reciprocidad y la aleatoriedad de redes inalámbricas en redes que utilizan *LoRa*.

- **Fugas de privacidad:** El almacenamiento de las transacciones de forma inmutable en la cadena de bloques puede derivar en fugas de privacidad. La primera posible solución es utilizar un esquema de mezclado de monedas (*coin-mixers scheme*) aunque numerosos autores han demostrado que este tipo de esquemas tienen vulnerabilidades. En este artículo los autores proponen como solución el trabajo de Dorri *et al.* [DKJ19] en el cual se presenta un esquema de almacenamiento de datos optimizado y flexible que podría reducir el riesgo de sufrir fugas de privacidad.
- **Mecanismos de incentivo:** Diseñar un mecanismo de incentivos para este tipo de sistemas y que cumpla los requerimientos de los diferentes tipos de aplicaciones, es un tópico abierto. Los autores proponen utilizar créditos de reputación que deriva en ciertos incentivos en función de la reputación de cada nodo.
- **Dificultad del análisis de datos (*Big Data Analytics* (BDA)):** Los esquemas tradicionales de análisis para Big Data no pueden ser utilizados debido a la baja capacidad computacional de los dispositivos IoT. Además, es muy difícil analizar datos anónimos sin descifrarlos. Entre las posibles soluciones a estos problemas se encuentran la posibilidad de utilizar un servidor *Multi-access Edge Computing* (MEC) ya que en combinación con la computación en la nube puede mejorar el tiempo de respuesta, ayuda a preservar la privacidad y disminuye la latencia existente al integrar la computación en la nube en una red BC.
- **Escalabilidad:** La escalabilidad es un problema latente en cualquier sistema BC, una de las posibles mediciones es la tasa de transferencia efectiva o "*throughput*" de transacciones por segundo entre el número de nodos IoT y el número de trabajos simultáneos. Las posibles soluciones serían diseñar algoritmos de consenso más escalables o utilizar blockchains privadas o híbridas en redes IoT, ya que pueden procesar las transacciones mucho más rápido que en una red pública al limitar el número de usuarios de la red.

- En noviembre de 2019, Butijn *et al.* [BTH19] publican una revisión sistemática de la literatura multivocal a partir de la cual, responden 5 preguntas de investigación propuestas. El artículo está disponible en forma de *preprint* desde 2019, y aparece finalmente en revista en junio de 2020 (Butijn *et al.* [BTH20]).

Un aspecto interesante de este artículo es que explican con detalle la metodología seguida para la selección de artículos incluidos en la revisión. Siguen el protocolo propuesto por Kitchenham [Kit04] utilizando la expresión de búsqueda diseñada en base a las preguntas de investigación planteadas, dando como resultado un total de 967 resultados, de los cuales 869 son de literatura blanca o científica y 98 son de literatura gris. Después de aplicar los criterios de inclusión/exclusión y de calidad de la literatura gris, obtienen un total de 111 elementos para revisar, 75 de literatura blanca y 32 de literatura gris. Para analizar los datos se basan en la teoría *Formal Concept Analysis*, o análisis formal de conceptos (FCA) y el enfoque *Straussian Grounded Theory*, o teoría fundamentada, o muestreo teórico (GT).

Muestran las tendencias en los temas de publicación en la TBC desde 2008 hasta el año 2019, e identifican los retos actuales de la tecnología que son el rendimiento (latencia y *throughput*), el almacenamiento y la privacidad de los datos, la gobernanza de la red BC y la usabilidad. Por último, a raíz de su revisión identifican numerosas tendencias como el uso de bloques para enlazar datos, el creciente número de algoritmos de consenso existentes, el interés creciente por las Blockchain híbridas (ver sección 2.3.3 en la página 30) y un aumento de la adopción de la TBC.

- En enero 2020, Sengupta *et al.* [SRD20] identifican los diferentes ataques en redes IoT en 4 categorías, ataques físicos, ataques en red, ataques al *software* o ataques referidos a los datos, además de revisar las contramedidas propuestas en la literatura para cada tipo de ataque.

  - **Ataques físicos:** En este tipo de ataques el atacante está muy próximo a la red o a los dispositivos IoT. Algunos de los ataques identificados son: manipulación (p ej. un enlace de red o un dispositivo RFID), inyección de código maliciosos en un dispositivo, interferencias de radiofrecuencia, inyección de nodos falsos, etc.
  - **Ataques de red:** Son los ataques en los que existe una manipulación de la red y no es necesario estar físicamente cerca de ella. Algunos ejemplos de esta categoría son el ataque del análisis del tráfico, suplantación de identidad en *Radio Frequency Identification*, identificación por radiofrecuencia (RFID) (*RFID Spoofing*), ataques Sybil o el ataque de intermediario (*MITM attack*). Este último ataque tiene grandes impactos en las redes IoT y entre las contramedidas encontradas, Park y Kang [PK16] proponen un mecanismo descentralizado de autenticación de la comunicación entre dispositivos, donde cada sensor está involucrado en la generación y distribución de claves de sesión, mejorando así significativamente el rendimiento en dispositivos con recursos limitados.
  - **Ataques al software:** En este tipo de ataques se incluyen los virus, gusanos, caballos de Troya, *malware*, entre otros.

- **Ataques a los datos:** Se refiere a todo tipo de ataques que afecte a la inconsistencia de los datos almacenados en la BC, el acceso no autorizado a ellos o una brecha que permita acceder a datos personales, sensibles o confidenciales de forma no autorizada.

El creciente uso de las redes IoT e IIoT implica un mayor volumen de datos y de dispositivos conectados. La TBC puede ser usada para coordinar miles de millones de dispositivos conectados, eliminar puntos únicos de fallo creando así redes IoT más resilientes o añadir privacidad a los datos gracias a la criptografía utilizada.

Los autores identifican las siguientes ventajas al utilizar la TBC en redes IoT:

- **Aumenta la confianza y la seguridad:** Gracias a la naturaleza distribuida e inmutable de la TBC se podrían eliminar los puntos únicos de entrada para vulnerabilidades o atacantes, además de proporcionar no-repudio al firmar todas las transacciones y estar "encadenadas". Además, utilizando contratos inteligentes para la comunicación entre dispositivos IoT y las características de la TBC, el sistema es más confiable.
- **Mayor robustez:** La descentralización que aporta la TBC podría hacer que los dispositivos IoT sean más accesibles y se pueda prevenir más fácilmente algunos tipos de ataques, además al eliminar los intermediarios de confianza, ahorramos costes asociados que existen en una red IoT centralizada.
- **Autonomía:** Utilizando contratos inteligentes se contribuye a la eliminación de terceros de confianza o autoridades centrales, además de permitir que los dispositivos IoT actúen de forma independiente, siempre respetando las reglas establecidas en el contrato correspondiente.
- **Procedencia de los datos:** Es posible trazar el recorrido, y por ende la procedencia, de los datos dado que todas las transacciones se almacenan en la cadena de bloques y son firmadas por los dispositivos o entidades que las generan.
- **Justicia:** Gracias al *token* o criptomoneda asociada a la red BC que se utilice, se podría incentivar a las partes interesadas para conseguir un sistema más justo.

Pero la integración de las redes IoT con la TBC también tiene varias desventajas:

- **Escalabilidad:** Con el aumento del número de dispositivos conectados también aumenta la cantidad de datos generados lo que conlleva un aumento significativo de los costes de transacción y las necesidades de almacenamiento.
- **Sobrecarga en las comunicaciones:** El aumento de dispositivos conlleva también el aumento del número de transacciones de la red, que deben ser enviadas a todos los nodos de la red lo que deriva en una sobrecarga de

las comunicaciones. Tampoco podemos olvidar que los nodos necesitan estar actualizados y mantener la misma copia de la cadena, lo que añade más sobrecarga.

- **Eficiencia:** Para conseguir que una transacción sea aprobada y añadida a la cadena necesita ser verificada por varios nodos, lo que merma la eficiencia operacional.
- **Gasto de energía:** Para llegar a un consenso no podemos utilizar PoW, ya que sería inviable.

Las investigaciones en términos de seguridad en sistemas IoT/IIoT que utilizan TBC se centran básicamente en el diseño de protocolos de autenticación, desarrollo de esquemas de autorización, prevención de ataques, preservación de la privacidad, gestión segura de datos, proveer seguridad básica y asegurar la confianza.

Concluyen que es necesario investigar y proveer nuevos mecanismos para frenar las amenazas existentes en este tipo de sistemas.

- En abril de 2020, Di Francesco Maesa y Mori [DM20] publicaron un artículo donde analizan 6 aplicaciones Blockchain 3.0 prometedoras: voto electrónico, gestión de los registros sanitarios, gestión de la identidad, notaría descentralizada (enfocada en la protección intelectual) y gestión de la cadena de suministro.

  Definen las siguientes características adquiridas por las soluciones que se basan en la TBC: incremento de la transparencia, proceso de auditoría más sencillo, reducción de los costes e incremento de la velocidad de gestión y verificación. Concluyen que la transparencia es la característica principal que tienen en común todas las aplicaciones analizadas, aunque no es la única. Hay una tendencia actual a utilizar la TBC como un repositorio de confianza común que permite la interoperabilidad entre distintas aplicaciones. Otra característica muy importante que aporta la TBC es la posibilidad de ejecutar contratos inteligentes, ya que en los sistemas tradicionales no es posible tener un código predefinido y acordado que se ejecute de manera automática y/o periódica. Por último, también indican que utilizar una red BC ahorra costes, aunque a priori pueda parecer al contrario, ya que sustituye la necesidad de intermediarios externos de confianza.

- En ese mismo mes, Islam *et al.* [Isl+20] realizaron la búsqueda de diferentes palabras clave (que habían elegido previamente en base a unos objetivos) en los 4 motores de búsqueda científicos principales y en Google, obteniendo un total de 860 resultados, que reducen a 51 después de aplicar los criterios de inclusión/exclusión diseñados.

  Muestran las características de la TBC mediante un mapa de concepto, y además exponen diferentes tablas con las ventajas o las dificultades de la tecnología Blockchain en determinados contextos. Por ejemplo, destacan ventajas en el sector IoT,

como la independencia de un tercero de confianza o la seguridad añadida que proporciona la TBC en este tipo de redes. Y en cuanto a las dificultades, están muy centradas en el contexto económico, destacando la latencia, el *throughput*, la falta de regulación legal o la seguridad inadecuada (alta probabilidad de sufrir algún tipo de ataque).

Esta revisión tiene 2 limitaciones reseñables. La primera es que los resultados están limitados por las cadenas de texto utilizadas, por lo que es posible que se haya perdido algún artículo relevante. La segunda es que la lista de ventajas y dificultades está basada únicamente en los 51 artículos revisados, además la mayoría tratan sobre Bitcoin y se ciñen principalmente al contexto económico, por lo que, no puede ser extrapolable a todas las redes actuales, y menos aún a todos los contextos donde tiene aplicación la TBC.

- En mayo de 2020 Belchior *et al.* [Bel+20] publicaron un artículo sobre la interoperabilidad de los sistemas BC. Para realizar la revisión se basan en las pautas propuestas por Keele *et al.* [Kee+07]. Después de realizar las búsquedas pertinentes en los motores de búsqueda más conocidos (Google Scholar, ACM Digital Libray, IEEE Xplore) las cuales dieron como resultado un total de 332 elementos, de los cuales, 262 eran de literatura científica y 70 de literatura gris, aplicaron los criterios de inclusión y exclusión definidos para obtener un total de 70 elementos a revisar. Además cabe destacar que se pusieron en contacto con todos los autores de la literatura gris a revisar para obtener un punto de vista actualizado de su trabajo.

  En esta revisión los autores proporcionan un contexto sobre la interoperabilidad, exponen los estándares y las arquitecturas existentes y destacan varias definiciones extraídas de literatura blanca y gris. Además, categorizan las soluciones encontradas en: enfoques destinados a criptomonedas, motores Blockchain o conectores Blockchain.

  - **Enfoques destinados a criptomonedas:** Los principales casos de uso identificados por los autores son la escalabilidad e intercambio de activos, los exchanges, intercambio de criptomonedas y la posibilidad de compartir activos a través de diferentes redes BC.
  - **Motores Blockchain:** El principal caso de uso identificado por los autores es la creación de redes Blockchain personalizadas.
  - **Conectores Blockchain:** Los principales casos de uso identificados por los autores son la posibilidad de *dApps* compatibles con varias redes BC de diferente naturaleza, la reducción del riesgo, protocolos de propósito general y la interoperabilidad eficiente de los *trusted relays*.

  Identifican y analizan numerosas soluciones actuales para cada una de las categorías anteriores, además de los retos abiertos que tienen relación con la interoperabilidad de sistemas BC. El primero de ellos es la grieta que detectaron los

autores entre la teoría y la práctica, además de la fragmentación, la capacidad de descubrimiento entre sistemas y la privacidad. Por último, remarcan los retos más importantes a los que se enfrenta la TBC en la actualidad que son la seguridad, la privacidad y la escalabilidad.

- En junio de 2020 Wang *et al.* [Wan+20c] publican una revisión en la cual repasan todas las publicaciones científicas sobre aplicaciones Blockchain en IoT e *Industrial Internet of Things*, o Internet Industrial de las cosas (IIoT). Las aplicaciones identificadas fueron:

  - **Ciudades inteligentes:** La TBC podría ser de utilidad en los aparcamientos públicos ya que ofrece en tiempo real el estado de todos los parkings de la ciudad, reduciendo así las dificultades para encontrar un aparcamiento libre y aliviando la afluencia de tráfico. Aunque, cabe destacar que la mayoría de las aplicaciones en ámbitos como las ciudades inteligentes y hogares inteligentes aún permanecen en la etapa experimental, ya que utilizar TBC todavía supone un gran coste económico y computacional y existe una falta de estándares internacionales.
  - ***Mobile commerce*, o comercio móvil (M-commerce)**: soluciona algunos de los problemas de seguridad eliminando la necesidad de un tercero de confianza, estableciendo intercambios entre dispositivos gracias a contratos inteligentes previamente acordados entre las partes.
  - **Trazabilidad alimentaria:** Gracias a la TBC es posible almacenar todos los estados, con su información asociada, por los que pasa un producto permitiendo así trazar en cuestión de segundos cualquier producto o lote que se encuentre en la cadena de bloques.
  - **Almacenamiento en la nube:** Utilizando TBC evitamos los altos costes y problemas de seguridad asociados al usar almacenamiento en la nube, los dispositivos pueden comunicarse entre si e intercambiar datos de forma segura a través de contratos inteligentes, eliminando cualquier la necesidad de un tercero de confianza.
  - **Gestión de la autenticación y el acceso:** Este tipo de aplicaciones, tradicionalmente utilizan una arquitectura centralizada y un inicio de sesión con usuario y contraseña. Utilizando la TBC se propone una arquitectura distribuida que valida los dispositivos y usuarios de la red, además de reemplazar las contraseñas por certificados SSL específicos. Suponiendo que los datos introducidos en la BC son legítimos, la cadena no podrá ser modificada y tenemos un registro persistente del estado de dispositivos, configuración o cualquier otra información que sea de utilidad.
  - **Hogar inteligente:** Utilizando la TBC en este tipo de aplicaciones conseguimos una mayor seguridad en las comunicaciones entre dispositivos al utilizar claves compartidas, que deben ser validadas antes de realizar alguna acción en la red, como por ejemplo, encender una bombilla o cerrar una

puerta. Otra ventaja es la posibilidad de recibir datos periódicamente hasta que el usuario envíe una transacción de fin de transmisión al minero, útil por ejemplo para una cámara web.

- **Big data:** Uno de los problemas actuales de las transacciones en tiempo real es su riesgo potencial de ataques de doble gasto donde el mismo *token* de seguridad generado para una transacción concreta sea utilizado dos veces. La TBC proporciona una solución resistente a este tipo de problemas, reduciendo el coste económico que suponen los sistemas tradicionales.
- **Vehículos eléctricos (*Electric Vehicles Cloud and Edge* (EVCE)):** Toda la información de los intercambios es almacenada en una BC y no puede ser modificada, eliminando la necesidad de un tercero de confianza y permitiendo compartir energía entre vehículos manteniendo la privacidad y la seguridad.

También enumeran una serie de limitaciones comunes para este tipo de sistemas en el que se combina la TBC con IoT observadas en el transcurso de la revisión, las cuales son:

- **Bajo rendimiento:** El rendimiento obtenido con este tipo de sistemas es relativamente bajo en comparación con las bases de datos tradicionales
- **Problemas relacionados con la privacidad:** Aunque las transacciones son públicas y pseudo-anónimas, las partes interesadas podrían identificar patrones y establecer relaciones entre los datos y direcciones o identidad del usuario.
- **Incrementa la complejidad:** El aumento del número de transacciones conlleva también un aumento de la dificultad del algoritmo de minado (en PoW y similares). Es necesario diseñar protocolos y algoritmos eficientes energeticamente, ya que los dispositivos IoT en la actualidad poseen una baja capacidad de cómputo y energía.
- **Plataforma de test estandarizada:** Para que usuarios y desarrolladores puedan aplicar TBC en sus sistemas IoT o IIoT es necesario que puedan probar la estabilidad, rendimiento o seguridad del sistema o aplicación. Los autores señalan que es necesario construir una plataforma de "pruebas" estandarizada que cuente con el apoyo de la mayoría de la comunidad.

- En julio de 2020 Huang *et al.* [Hua+20] publican una revisión realizada con 66 estados del arte de diferentes categorías (protocolos de consenso, escalabilidad, análisis de datos, IoT, 5G, COVID-19, etc.) para obtener una visión actual y general del estado del arte en la TBC.

  Muestran los últimos avances en diferentes tópicos relativos a la TBC. En primer lugar, para la escalabilidad (latencia y *throughput*) indican que constituyen dos cuellos de botella críticos actuales en la TBC. Destacan el trabajo Hari *et al.*

[HKL19] en el cual proponen una mecanismo de confirmación de transacciones determinista llamado *ACCEL* en el cual mediante la identificación de bloques singulares, consiguen usarlos de determinada manera para reducir el retraso en la confirmación, incrementando así el *throughput*. Otro enfoque a destacar es Prism, propuesto por Yang *et al.* [Yan+19], un protocolo Blockchain que consigue, en las condiciones de los experimentos, hasta 70000 transacciones por segundo, manteniendo la seguridad de Bitcoin. Aunque si solo utilizan un *voter chain* existe una latencia de confirmación de 10 segundos, que puede ser solucionada utilizando múltiples *voter chain* con los consiguientes problemas de seguridad. En segundo lugar consideran el almacenamiento de datos eficiente, destacando 3 trabajos. En el primero de ellos *Erasure code-based* proponen un nuevo tipo de nodos BC ligeros utilizando el método del código de borrado (*erasure code*) para reducir el espacio de almacenamiento en redes BC. El segundo, *Jidar:Data-Reduction Strategy* propuesto por Dai *et al.* [Dai+19] es una estrategia de reducción de datos para Bitcoin en la que cada nodo solo tiene que almacenar las transacciones importantes y las ramas Merkle que contienen bloques completos. Por último, revisan el trabajo de Xu y Huang [XH20] titulado *Segment Blockchain* en el cual proponen un mecanismo de almacenamiento reducido de modo que cada nodo solo tenga que almacenar un segmento de la cadena de bloques.

En cuanto a los retos y trabajo futuro apuntan que es necesario seguir avanzando en los modelos teóricos planteados para mejorar la escalabilidad de las redes BC, diseñar mecanismos resilientes para las técnicas actuales de *sharding*, mejorar los protocolos actuales de *sharding* en términos de rendimiento (latencia y *throughput*), diseñar nuevos mecanismos que permitan una comunicación entre diferentes redes BC más rápida, proponer nuevas técnicas para ordenar y gestionar los bloques en sistemas que utilizan varias redes BC, nuevas soluciones para la aceleración del *hardware* con el fin de mejorar el rendimiento de los nodos y por tanto, el de la red. Además, indican la importancia de seguir investigando en tópicos como la privacidad, la seguridad en redes que tienen criptomonedas, la integración de TBC en el *Big Data* o la optimización del rendimiento en todas las capas de una red BC.

- En agosto de 2020 Bhushan *et al.* [Bhu+20] publicaron un artículo en el cual estudian la integración de TBC en redes IoT y presentan las aplicaciones más relevantes publicadas en la literatura científica, además de identificar los retos actuales y las direcciones futuras de la investigación en este tópico.

  Además de los propios retos actuales de las redes IoT como pueden ser la comunicación utilizando redes 4G o 5G, sistemas telemétricos o RFID, identifican los retos a los que se enfrentan las aplicaciones que combinan IoT con TBC.

  · **Eficiencia energética:** Los dispositivos IoT tradicionalmente poseen muy poca capacidad de cómputo y energía, mientras que las redes BC (p. ej. Bitcoin) exigen un alto consumo energético. Es por ello que es necesario

desarrollar nuevos protocolos de consenso que requieran un menor consumo energético.

- **Seguridad:** En principio una red IoT podrá mantener unos niveles de seguridad aceptables siempre que el sistema sea resistente a los ataques más comunes y la infraestructura centralizada sea de confianza. En las redes BC siempre puede existir algún nodo malicioso y uno de los ataques más comunes "heredado" de la TBC a redes IoT es el ataque del 51 % (ver sección 2.4.4.1 en la página 34).
- **Privacidad:** En una red BC el anonimato no está garantizado ya que un tercero podría analizar las transacciones y relacionar *hash* o dirección pública (obtenida de la clave pública) con identidades de usuarios reales. En una red IoT la complejidad aumenta y los dispositivos IoT podrían revelar datos privados de lo usuarios. Aunque existen numerosas propuestas de técnicas y métodos para mejorar la privacidad en estos campos, debido a los recursos limitados de los dispositivos IoT es difícil ponerlos en práctica.
- **Latencia y throughput:** Al incluir TBC en redes IoT aumenta la latencia y disminuye el número de transacciones por segundo, por tanto, es necesario encontrar soluciones que permitan disminuir la latencia y aumentar el throughput en este tipo de sistemas.
- **Infraestructura Blockchain:** A medida que pasa el tiempo y se generan transacciones el tamaño de la cadena se hace más grande aumentando así el tiempo de sincronización de los nuevos nodos (o re-sincronizaciones) y la necesidad de mayor capacidad de almacenamiento en los nodos completos. Como los dispositivos IoT no tienen la capacidad de almacenamiento suficiente es necesario explorar técnicas de compresión de la cadena o la utilización de nodos ligeros que almacenen solo las últimas transacciones. Aún así, es necesario que existan nodos que mantengan una copia completa de la cadena en la red, por lo que siempre habrá varios nodos con altas capacidades de almacenamiento aunque sea una red IoT.

Por último, los autores proponen las posibles líneas de investigación futuras a seguir en este ámbito basándose en la observación de las tendencias seguidas por los artículos revisados.

- **Problemas de escalabilidad en TBC:** Es necesario aumentar los requisitos de almacenamiento de datos debido al aumento de bloques en la cadena a lo largo del tiempo. Además, el throughput se ve reducido debido a la dificultad de alcanzar un consenso descentralizado en redes públicas (ver sección 2.3.1 en la página 28).
- **Dispositivos "IoT edge" restringidos:** Es necesario ampliar la capacidad computacional de los dispositivos IoT con recursos limitados, además de aumentar las capacidades de almacenamiento y computación de las puertas de enlace IoT conectadas a la red BC.

- **Infraestructura Blockchain:** En esta línea de investigación entra el abordaje de los requisitos de seguridad, estabilidad y mejoras criptográficas en redes BC, además de la necesidad de creación de una infraestructura que proporcione confianza a toda la red.
- **Redes móviles y BIoT:** Involucra la programación adecuada de los recursos. Además las redes BC pueden ayudar a las redes móviles permitiendo almacenar recursos virtualizados.
- **Privacidad en redes BC:** Direcciones utilizando un seudónimo para cumplir con los requisitos de privacidad, los contenidos en una red Blockchain sin permisos son públicos para facilitar su auditabilidad. Por último, mantener la integridad de los datos en una arquitectura por niveles.
- **Aprendizaje automático y Big Data para redes IoT descentralizadas:** En este punto se incluye el uso de redes BC para aplicaciones Big Data, utilizar la TBC para descentralizar redes IoT y autenticar los conjuntos de entrenamiento utilizados en el aprendizaje automático. Por último, incluye la toma de decisiones teniendo en cuenta la ética (cómo y por qué utilizar los datos generados por este tipo de sistemas).

- En agosto de 2020, Hunhevicz y Hall [HH20] categorizan los casos de uso más comunes en la TBC en la actualidad y observan que la mayoría utilizan dicha tecnología para mejorar la transparencia y optimización de procesos a través de la automatización y la eliminación de intermediarios. En la categorización no están todos ya que el campo de la TBC se encuentra en continua evolución y en cuestión de meses las categorías y casos de uso podrían necesitar una revisión. Presentan un diagrama de flujo diseñado en base a la revisión de 8 diagramas similares presentes en la literatura, mediante el cual en función de las necesidades de tu sistema te recomendará si necesitas o no una BC, y en caso afirmativo te sugerirá el tipo de Blockchain más adecuado.

### 3.1.3. Conclusiones

Una vez revisados los estados del arte previos, y detectados los tópicos emergentes más importantes, acometeremos nuestra propio estado del arte, centrado en estos tópicos y con el objetivo de actualizar las referencias a lo publicado en los últimos dos o tres años.

Reiterando lo ya dicho al inicio de esta sección, los tópicos identificados son:

- Mejora de la privacidad del almacenamiento de los datos
- Mejora de la capacidad de almacenamiento, que limita la escalabilidad
- Mejora de la latencia y el rendimiento, que limitan la escalabilidad
- Mejora de los mecanismos de gobernanza, para posibilitar una evolución consensuada de los protocolos de la cadena de bloques.

El resto de este capítulo se estructura como sigue: En la sección siguiente (3.2) se delinea la metodología que seguiremos para seleccionar los artículos a revisar; en la sección 3.3 (pág. 79) mostraremos una panorámica de los resultados obtenidos al aplicar la metodología, para entrar en la sección 3.4 (pág. 85) en más detalles de cada una de las publicaciones revisadas. Finalmente en la sección 3.5 (pág. 103) daremos respuesta a las preguntas de investigación, objetivo de este trabajo, .

## 3.2. Metodología

Para realizar nuestra revisión sistemática de la literatura científica en el campo de la TBC, centrada en los tópicos previamente seleccionados, seguiremos bastante fielmente la metodología seguida por Butijn *et al.* en su artículo titulado "*Blockchains: a Systematic Multivocal Literature Review*" [BTH19] (ya reseñado en la sección 3.1, en particular en la página 61). Seguidamente describiremos esta metodología, señalando en qué nos separamos del trabajo de Butijn *et al.*

La metodología comienza por plantear las preguntas de investigación, o *research questions*, que en nuestro caso estarían más centradas en los tópicos previamente identificados:

- **PI1:** ¿Cuáles son los últimos avances en la investigación sobre TBC en la actualidad, en particular en los campos de privacidad de los datos, la capacidad de almacenamiento de datos, la mejora de latencia y rendimiento y los mecanismos de gobernanza?
- **PI2:** ¿A qué retos se enfrenta la investigación en estos tópicos?
- **PI3:** ¿Qué tendencias se pueden inferir?

Las dos últimas cuestiones están muy relacionadas entre sí, ya que indican la dirección que está tomando y va a tomar la investigación en el área de la TBC. En la PI1 nos enfocamos en averiguar a qué retos se esta enfrentando actualmente la investigación en TBC y en que estado se encuentra cada uno de ellos, y en la PI2 y PI3 nos centramos en proponer ciertas rutas que podría seguir la investigación en este campo.

Para dar respuesta a estas cuestiones es necesario revisar lo publicado en los últimos años en estos temas. Para ello es conveniente seguir una metodología sistemática, de modo que la selección de trabajos consultados no sea arbitraria, sino que siga métodos objetivos y replicables.

Las fuentes que se pueden consultar en una revisión sistemática son múltiples. Garousi *et al.* [GFM19] las clasifican en diferentes espectros (literatura blanca, gris y negra), que se resumen en la tabla 3.2 en la página siguiente.

La literatura blanca, también conocida como literatura científica, engloba los artículos en revistas revisados por pares, conferencias y libros o capítulos de los mismos. Más adelante se establecerán criterios de calidad para evitar, por ejemplo, las conocidas como "revistas depredadoras". En cuanto a la literatura gris, puede aportar mucho valor

| Literatura Blanca | Literatura Gris | Literatura Negra |
|---|---|---|
| Artículos en revistas | Pre-prints | Ideas |
| Conferencias | E-prints | Conceptos |
| Libros | Reportes técnicos | Opiniones |
| | Presentaciones | |
| | Conjuntos de datos | |
| | Audio o vídeo | |
| | Blogs | |

**Tabla 3.2.-** Tipos de literatura según Garousi *et al.* [GFM19]

| 1º nivel<br>**Alta credibilidad** | 2º nivel<br>**Credibilidad moderada** | 3º nivel<br>**Baja credibilidad** |
|---|---|---|
| Libros | Reportes anuales | Blogs |
| Revistas | Noticias | Emails |
| Reportes gubernamentales | Presentaciones | Tweets |
| Libros blancos (whitepapers) | Vídeos | |
| | Webs de preguntas y respuestas | |
| | Artículos tipo Wiki | |

**Tabla 3.3.-** Tonos de la literatura gris según Garousi *et al.* [GFM19]

a una revisión como ésta ya que se incluye literatura producida por profesionales o expertos del sector que trabajan en la industria, gobierno, o cualquier otra entidad y que no publican sus resultados en editoriales comerciales ni se someten al proceso de revisión por pares. Además, Garousi *et al.* [GFM19] definen "tonos" (ver tabla 3.3) en la literatura gris que representan la credibilidad de la información según el medio en el que se haya publicado.

Teniendo en cuenta los tipos de literatura existentes, el siguiente paso es identificar en la literatura blanca todos los artículos científicos revisados por pares y en la literatura gris aquellos que puedan ser relevantes para nuestro estudio. Para ello necesitamos definir un protocolo para distinguir entre los elementos realmente interesantes y los que tienen poca calidad o interés para nuestra revisión. Una vez definido dicho protocolo de inclusión/exclusión, y después de realizar una búsqueda en los motores de búsqueda seleccionados:

1. Se obtiene una selección de artículos para revisar.
2. Se revisan dichos artículos.
3. Se realiza un análisis los datos obtenidos con la revisión.
4. Se extran las conclusiones.

Siguiendo la metodología para realizar revisiones sistemáticas propuesta por Kitchenham [Kit04] para la literatura blanca, en primer lugar es necesario elaborar una expresión regular que cubra todos los términos de búsqueda necesarios. El siguiente paso es aplicar la expresión regular completa o dividirla en sub-expresiones según el motor de búsqueda lo requiera. Una vez obtenidos los resultados de los motores de búsqueda se aplican los criterios de inclusión/exclusión que se muestran en la tabla 3.4 en la página siguiente para obtener el conjunto de artículos a analizar.

Para formar la expresión regular de los términos de búsqueda nos basamos en los tópicos que hemos identificado en la sección 3.1.3 en la página 70. Nuestra expresión regular se formará a partir de los cuatro componentes siguientes (en los que $\vee$ representa la operación booleana "o"):

($a$) "Blockchain" $\vee$ "DLT",

($b$) "data privacy"

($c$) "data storage" $\vee$ "latency" $\vee$ "throughput"

($d$) "governance"

La expresión regular resultante (en la que $\wedge$ representa la operación booelana "y") sería:

$$(a) \wedge ((b) \vee (c) \vee (d)) \tag{3.1}$$

La justificación de estos términos de búsqueda es que en los resultados queremos que obligatoriamente aparezcan las palabras Blockchain o DLT (ya que ambas suelen usarse para referirse a la tecnología de cadena de bloques), y después los tópicos previamente identificados, agrupando en la expresión ($c$) para mayor claridad los relacionados con la escalabilidad.

Una vez diseñada la expresión a utilizar, realizamos las búsquedas en los motores de búsqueda científicos Scopus, Science Direct y Springer en abril de 2020. Con el fin de analizar los resultados de todas las búsquedas realizadas y poder descartar los elementos que no cumplan con las restricciones planteadas, necesitamos hacer uso de ciertas técnicas para extraer la información necesaria de los motores de búsqueda de la forma más rápida, sencilla y automatizable posible. Para obtener los resultados de Science Direct en primer lugar se ha utilizado la técnica conocida como *"web scraping"* para obtener el listado de todos los artículos que cumplen las restricciones exigidas[1]. A continuación, realizamos una llamada a la API oficial de Science Direct para obtener los datos que necesitamos de cada artículo utilizando el DOI obtenido mediante *web scraping*. En el caso de Springer, utilizamos su API oficial para realizar las búsquedas y obtener todos los datos necesarios de cada uno de los resultados obtenidos. Por último, en el caso de Scopus fue realmente sencillo obtener los resultados formateados ya que

[1]Código disponible en el repositorio git: `https://bitbucket.org/ruben_uniovi/scientific_literature_web_scraping/`

| **Criterios de inclusión** | **Criterios de exclusión** |
|---|---|
| Artículos cuyo principal objetivo es proponer una nueva visión o solución para mejorar la privacidad de los datos, el almacenamiento de datos, la gobernanza, latencia o el tiempo de procesamiento en una red BC. | Artículos publicados en revistas científicas con un factor de impacto menor a 2.5 puntos o en conferencias con una calificación por debajo del nivel A. ***Motivo:*** *el factor de impacto nos permite eliminar ciertas revistas científicas que no cumplen unos mínimos estándares de calidad.* |
| Artículos que tratan sobre la privacidad de los datos, el almacenamiento de datos, la gobernanza, latencia o tiempo de procesamiento en una red BC. ***Motivo:*** *El objetivo de este trabajo fin de Máster es ofrecer una visión académica del avance de la TBC en sus puntos débiles hasta el momento.* | Artículos que proponen utilizar la TBC como vehículo para mejorar alguno de los siguientes aspectos: privacidad de los datos, almacenamiento de datos, gobernanza, latencia o tiempo de procesamiento. ***Motivo:*** *Siempre buscamos estos términos en referencia a la TBC en sí, no a una aplicación de la misma. Por ejemplo, utilizar la TBC para implementar un sistema de voto electrónico sería motivo de exclusión.* |
| Debe ser un artículo científico revisado por pares y publicado en una revista científica o un capítulo de un libro publicado por una editorial científica. | Artículos publicados con una fecha anterior al año 2018. |

**Tabla 3.4.-** Criterios de inclusión/exclusión para la literatura blanca.

desde su portal web nos permiten hacer búsquedas avanzadas con operadores booleanos y exportar los resultados con todos los campos necesarios a formato CSV.

El siguiente paso es eliminar los elementos repetidos y aplicar los criterios de inclusión/exclusión definidos anteriormente (véase tabla 3.4 en la página anterior). Se revisan después todos los elementos del conjunto de datos actual descartando de la revisión a todos aquellos cuyo título y/o resumen no encaje en nuestros objetivos previamente definidos. La figura 3.1 en la página 77 muestra el diagrama de flujo seguido en esta revisión, y el número de artículos que iban resultando de cada fase.

Realizamos el mismo proceso para la literatura gris, pero ahora seguimos las pautas propuestas por Garousi *et al.* [GFM19], para realizar revisiones de literatura gris. Estos autores definen 3 pautas: búsqueda, selección de la fuente y evaluación de la calidad.

Utilizamos el motor de búsqueda Google con la expresión regular (3.1), lo que requirió una adaptación porque Google no producía resultados correctos con expresiones tan complejas. En concreto, fue necesario añadir caracteres especiales para poder obtener los resultados esperados (antes de eso se obtenían sólo entre 10 y 20 resultados por búsqueda). Añadimos el operador + delante de cada palabra para forzar que los resultados incluyan todos los términos de búsqueda (equivalente al operador $\wedge$ de la fórmula), y el operador $*$ detrás de los términos "blockchain" y "DLT" para que actúe como comodín y no descarte los resultados que contienen esos términos en plural. Teniendo esto en cuenta, la expresión de búsqueda, representada de forma abstracta, sería:

$$(+a*) \wedge ((+b) \vee (+c) \vee (+d))$$

Y ya que Google no permite la mezcla de expresiones booleanas "and" y "or" (en realidad sí lo permite aparentemente, pero los resultados son erróneos), se descompuso la expresión anterior en las tres siguientes, en las que, aunque la fórmula usa un $\wedge$ en Google no es necesario operador alguno ya que implícitamente entiende que todos los términos buscados deben aparecer en el resultado:

$$(+a*) \wedge (+b)$$
$$(+a*) \wedge (+c)$$
$$(+a*) \wedge (+d)$$

Cada línea de la fórmula dio lugar a una consulta, y los resultados de todas ellas se agregaron posteriormente para simular la operación "or" ($\vee$).

Del resultado obtenido en cada consulta se admitirán las 5 primeras páginas (100 elementos). A partir de la página 5 de resultados, se aplicarán los criterios de calidad definidos por Garousi *et al.* [GFM19] y mostrados en la tabla 3.5 en la página siguiente y cuando más de la mitad de los resultados de una página no cumplan los criterios de calidad definidos, la búsqueda se da por concluida. Todos los elementos que no cumplan los criterios de calidad serán excluidos del conjunto de resultados.

| **Categoría** | **Condición** | **Mínimo de condiciones a cumplir** |
|---|---|---|
| Autoría | Está publicado en una organización con reputación o el autor tiene afiliación con una organización con reputación. | 2 de 3 |
| | El autor publicó otros trabajos en el campo. | |
| | El autor tiene experiencia en el campo (p. ej. Un puesto de trabajo). | |
| Objetividad del estudio | El establecimiento de las fuentes es objetivo. | Todas |
| | No hay conflictos de interés conocidos. | |
| | Las conclusiones están respaldadas por datos objetivos. | |
| Metodología | La fuente tiene un objetivo claramente definido. | 4 de 6 |
| | La fuente tiene una metodología claramente definida. | |
| | La fuente está apoyada por referencias documentadas y con cierta autoridad en el campo. | |
| | Los límites del estudio están claramente establecidos. | |
| | El trabajo cubre una pregunta específica. | |
| | El trabajo está enfocado en una población determinada. | |
| Fecha | Se especifica claramente la fecha del documento. | Todas |
| Posición frente a las fuentes relacionadas | La literatura gris o formal relacionada se ha discutido o vinculado. | Todas |
| Novedad | El documento enriquece o añade algo único a la investigación. | 1 de 2 |
| | El documento refuerza o refuta la posición actual. | |
| Impacto | La fuente contiene citas y enlaces para sostener los argumentos planteados. | Todas |

**Tabla 3.5.-** Criterios de calidad para la literatura gris.

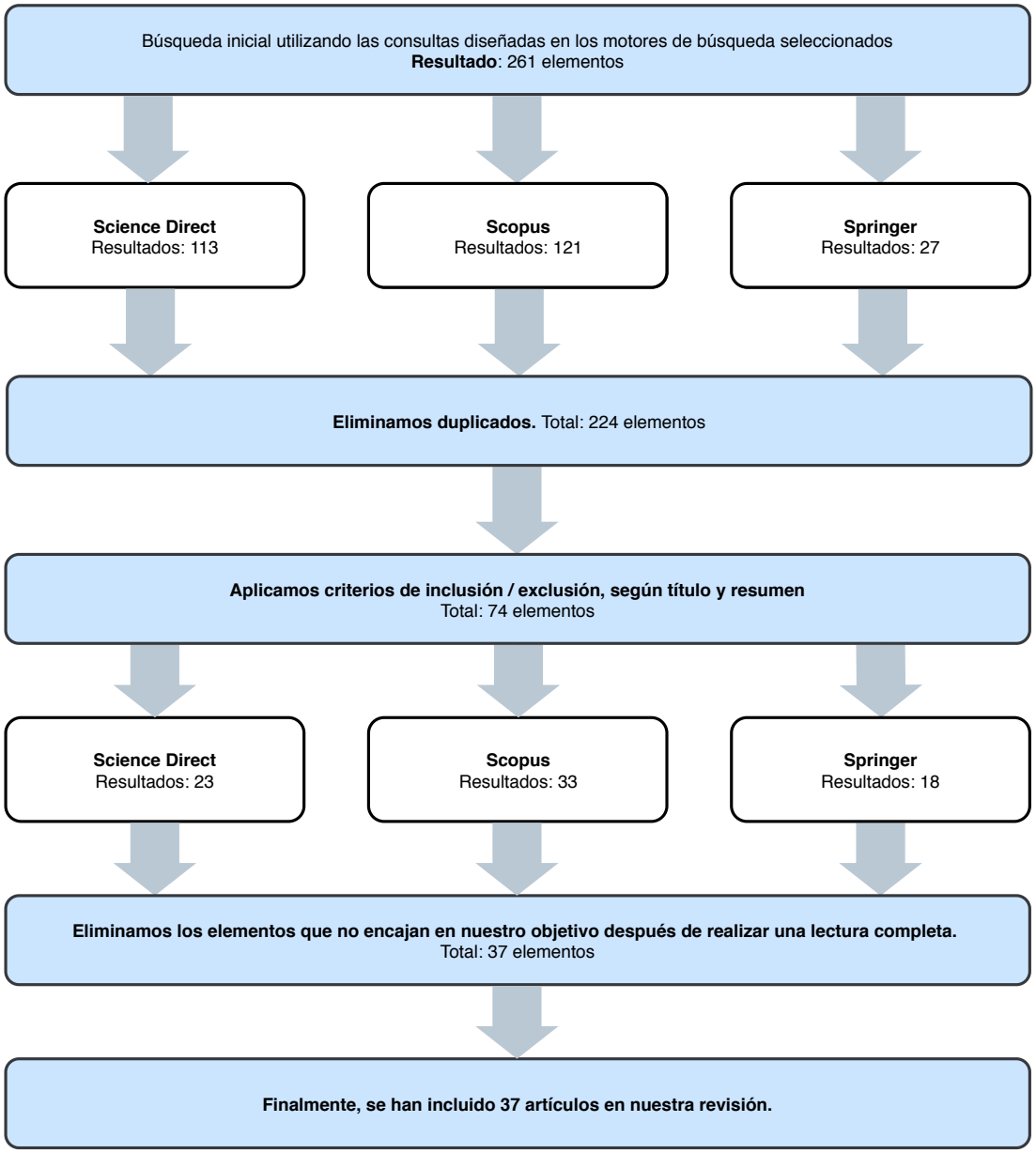


**Figura 3.1.-** Diagrama de flujo del proceso de revisión sistemática de la literatura científica (blanca).

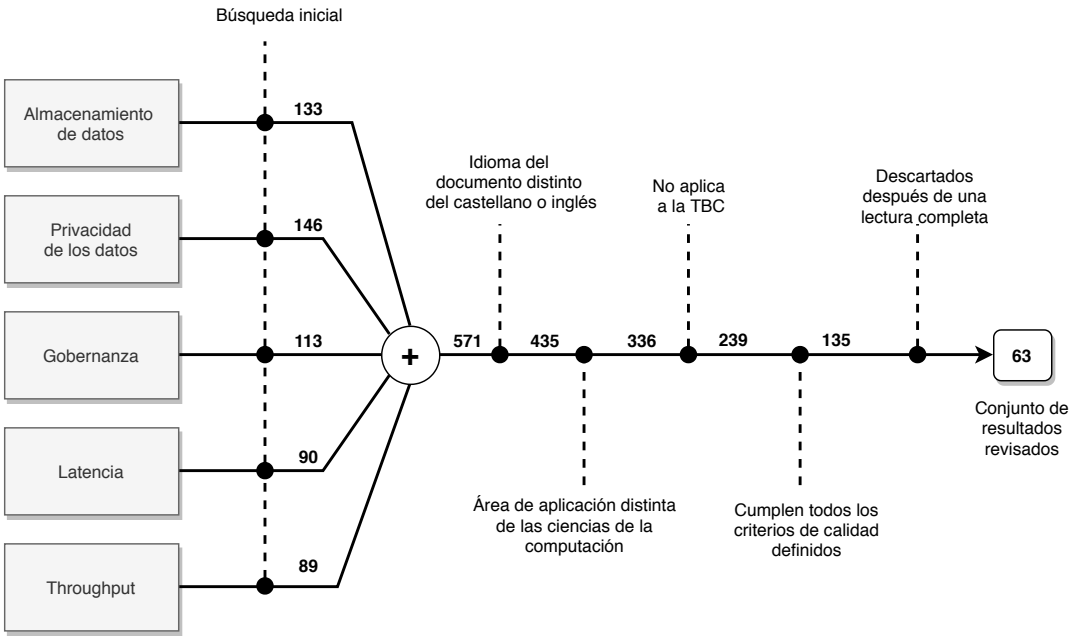


**Figura 3.2.-** Diagrama de flujo del proceso de revisión sistemática de la literatura gris.

## 3.3. Análisis de los resultados obtenidos

El número de artículos resultante de la metodología antes expuesta es bastante elevado, por lo que se muestran en esta sección algunas gráficas y tablas que ayuden a visualizar las tendencias generales, antes de pasar a la descripción más detallada de cada artículo, en la sección 3.4 en la página 85.

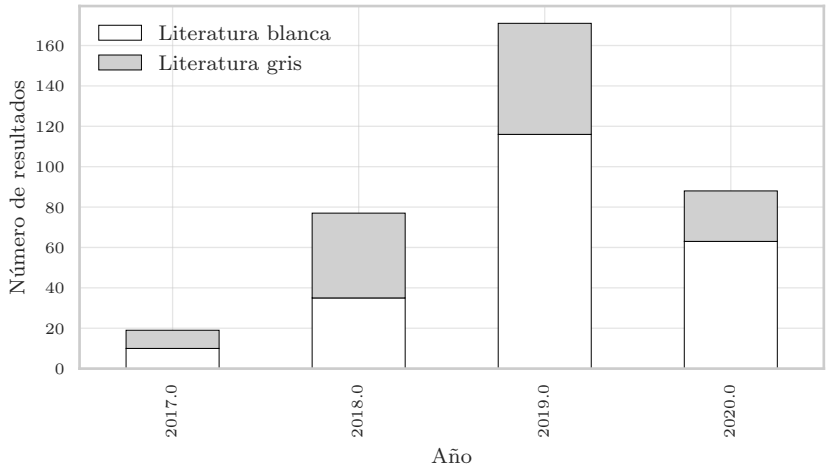


**Figura 3.3.-** Total de resultados por año tras la eliminación de duplicados y resultados irrelevantes

La figura 3.3 muestra el total de artículos obtenidos de las búsquedas, tras un preprocesamiento para eliminar duplicados de los resultados. Este preprocesamiento ha consistido en el caso de la literatura blanca en la eliminación (automatizada por un *script*) de los artículos cuyo DOI fuera el mismo, y en el caso de la literatura gris de la eliminación (también asistida por un *script*) de resultados cuyo título fuera idéntico (pues no hay DOI para los resultados de literatura gris) eliminando también los que ya estuvieran en la literatura blanca (ya que Google puede proporcionar entre sus resultados también artículos publicados en revistas que hubieran aparecido ya entre los resultados de los motores de búsqueda de literatura científica). En la literatura gris, además del preprocesamiento automático, se han inspeccionado manualmente todos los títulos de los artículos y se han descubierto muchos que, a pesar de tener referencias a los términos buscados, no guardaban relación alguna con la temática de este trabajo (por ejemplo, aparecían numerosos resultados sobre medicina). Estos resultados se eliminaron también, de forma manual.

En la literatura blanca el preprocesamiento redujo los 261 resultados inicialmente obtenidos a 224. En la literatura gris, el total inicial de 571 resultados se redujo a 239.

Un detalle adicional es que en los resultados de la literatura gris no es posible de forma general conocer la fecha de publicación, pues esa meta-información no está en los resultados proporcionados por Google. No obstante, en muchos de los resultados sí que se puede encontrar una fecha, porque esta aparece como parte del título, o como parte

de la URL accedida, o como parte del *snippet* o breve resumen que proporciona Google sobre cada resultado. Utilizando este método para estimar la fecha de publicación de cada resultado, se encontraron que de los 239 resultados, en 105 se podía estimar la fecha y en 134 no. La gráfica de la figura 3.3 obviamente incluye sólo los resultados para los que se pudo obtener fecha.

Debe tenerse en cuenta que el interés de esta figura es más bien anecdótico, ya que muchos de los resultados contabilizados en ella pueden considerarse "falsos positivos" y han sido descartados posteriormente atendiendo a los criterios previamente explicados. No obstante sí que hace patente el incremento de publicaciones con el paso del tiempo. Cuando se realizaron las búsquedas había transcurrido aún sólo cuatro meses de 2020, por lo que a final de año, si se siguen publicando al mismo ritmo, la barra de 2020 debería triplicar la altura mostrada, confirmando la tendencia alcista en el número de publicaciones.

### 3.3.1. Literatura blanca

La figura 3.4 muestra 147 artículos de los 224 resultados de la literatura blanca, que tratan de los temas investigados en este trabajo, atendiendo a la lista de palabras-clave (*keywords*) proporcionadas por los motores de búsqueda. Por ejemplo, si entre la lista de palabras-clave retornadas por el motor de búsqueda aparece la palabra "*Privacy*" se computa ese artículo como parte de la columna de color naranja. Los contadores de las palabras-clave "*Storage*", "*Latency*" y "*Throughput*" se han representado apilados en la misma columna, con diferentes tonos de verde, ya que cada uno representa un aspecto diferente de la *escalabilidad*.

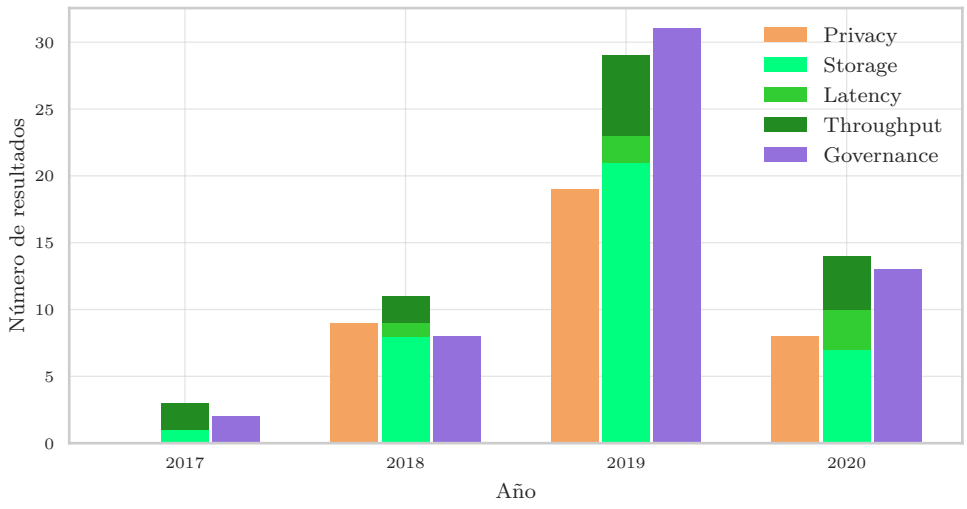


**Figura 3.4.-** Literatura blanca: número de resultados por tópico y por año

Un detalle a señalar con respecto a esta figura es el gran número de resultados (especialmente en 2019) relacionados con el término *governance*. Esta columna incluye

| Artículo | Temas tratados: Seguridad de los datos | IoT | Firma digital | Contratos inteligentes | Hyperledger Fabric | Criptografía homomorfica | Gestión de préstamos | Transparencia |
|---|---|---|---|---|---|---|---|---|
| Shrestha y Kim [SK19] | ● | ● | ○ | ○ | ○ | ● | ○ | ○ |
| Arora *et al.* [Aro+19] | ● | ○ | ● | ◐ | ○ | ○ | ○ | ○ |
| Loukil *et al.* [Lou+18] | ○ | ● | ○ | ● | ○ | ○ | ○ | ● |
| Wang *et al.* [WGC19] | ◐ | ○ | ○ | ○ | ● | ○ | ● | ○ |
| Wang *et al.* [Wan+20a] | ◐ | ○ | ○ | ○ | ○ | ○ | ● | ◐ |
| Gilda y Mehrotra [GM18] | ● | ○ | ○ | ● | ● | ○ | ○ | ◐ |

**Tabla 3.6.-** Matriz de análisis de la literatura blanca para los términos relacionados con la privacidad de los datos

en realidad un gran número de "falsos positivos", pues muchos de los artículos que incluyen "*Governance*" en su lista de palabras-clave, no tratan en realidad del problema de la gobernanza de la Blockchain, sino del uso de esta tecnología en asuntos de Gobierno. No es posible filtrar automáticamente para dejar fuera los no relacionados en este tópico, sino que ha sido necesario revisar estos resultados manualmente para quedarse sólo con los relevantes, cuyo número es mucho menor (ver sección 3.4.5 en la página 99)

Tras aplicar las fases del proceso ilustradas en la figura 3.1 para eliminar resultados según los criterios explicados en la sección acerca de la metodología (3.2), el número de artículos finalmente revisado fue de 38. Tras leer estos artículos es ya posible clasificarlos manualmente de forma más precisa en la temática adecuada, y confeccionar una lista de tópicos más ajustados que resultan de la lectura de esos artículos.

Los tópicos ya no serán tan generales como "Privacy", sino que, dentro del grupo de artículos que tratan ese tópico, será posible precisar sub-tópicos (o retos) más concretos, como "Necesidad de estándares", "Transparencia", etc. Y de forma análoga con los restantes tópicos.

El resultado de este análisis (lectura, clasificación y definición de sub-tópicos apropiados) se resume en las tablas 3.6, 3.7 y 3.8 en estas páginas.

La tabla 3.6 reúne los 6 artículos revisados que tratan del tema de la privacidad, y muestra los sub-tópicos más comúnmente encontrados en este área. Los detalles de cada artículo se explican en la sección 3.4.1.

La tabla 3.7 en la página siguiente reúne los 21 artículos revisados que tratan del tema de la escalabilidad, que aúna los tres términos "*Data Storage*", "*Latency*" y "*Throughput*", y muestra los sub-tópicos más comúnmente encontrados en este área.

| | Término | | | Temas tratados | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Artículo | *Data storage* | *Latency* | *Throughput* | Alta escalabilidad | Algoritmo de consenso | IoT | Micro-bloques | Anonimato | Alta Disponibilidad | Sharding | Ahorro de energía | Contratos inteligentes | Interoperabilidad | Consulta de datos eficiente | Datos compartidos seguros | Integridad de los datos | DHT |
| Dorri *et al.* [Dor+19] | ○ | ● | ○ | ● | ● | ● | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| Alrubei *et al.* [Alr+20] | ○ | ● | ○ | ● | ● | ● | ○ | ○ | ● | ○ | ◐ | ◐ | ○ | ○ | ○ | ◐ | ○ |
| Fan *et al.* [Fan+19] | ○ | ● | ○ | ● | ● | ● | ○ | ○ | ○ | ○ | ◐ | ○ | ○ | ○ | ○ | ◐ | ○ |
| Han *et al.* [Han+20] | ○ | ● | ○ | ● | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| Hosseini Bamakan *et al.* [HMB20] | ○ | ○ | ● | ● | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| Kim *et al.* [Kim+19] | ○ | ○ | ● | ● | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| Sanka y Cheung [SC18] | ○ | ○ | ● | ○ | ◐ | ○ | ○ | ○ | ◐ | ○ | ● | ◐ | ○ | ○ | ○ | ● | ○ |
| Xu *et al.* [Xu+17] | ○ | ○ | ● | ● | ● | ○ | ● | ○ | ◐ | ○ | ○ | ○ | ○ | ○ | ○ | ◐ | ○ |
| Xiong *et al.* [Xio+19] | ○ | ○ | ● | ● | ◐ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ◐ | ○ | ◐ | ○ | ○ |
| Yu *et al.* [Yu+19] | ○ | ○ | ● | ● | ● | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ |
| Zegzhda *et al.* [ZMM18] | ● | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ● | ◐ | ○ |
| Do y Ng [DN17] | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ◐ | ○ | ○ | ○ | ○ | ● | ◐ | ● | ○ |
| Peng [Pen18] | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ◐ | ○ | ○ | ○ | ○ | ◐ | ○ | ● | ○ |
| Qiao *et al.* [Qia+19] | ● | ○ | ○ | ○ | ○ | ● | ○ | ● | ○ | ○ | ○ | ◐ | ○ | ○ | ○ | ◐ | ○ |
| Shala *et al.* [Sha+19] | ● | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ◐ | ○ | ○ | ◐ | ● | ○ |
| Taylor *et al.* [Tay+20] | ● | ○ | ○ | ○ | ○ | ● | ○ | ● | ○ | ○ | ○ | ○ | ○ | ◐ | ● | ● | ○ |
| Fan *et al.* [Fan+20] | ● | ○ | ○ | ○ | ● | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ |
| Ali *et al.* [Ali+18] | ● | ○ | ○ | ● | ○ | ○ | ○ | ○ | ◐ | ○ | ◐ | ○ | ○ | ● | ◐ | ○ | ● |
| Wang *et al.* [Wan+20b] | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ◐ | ○ | ○ | ● | ○ | ○ | ○ | ◐ | ○ |
| Liu *et al.* [LWL18] | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ◐ | ● | ○ | ◐ | ◐ | ● |
| Roy *et al.* [Roy+19] | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ◐ | ○ | ○ | ○ | ● | ● | ○ | ◐ | ○ |

**Tabla 3.7.-** Matriz de análisis de la literatura blanca para los términos relacionados con la escalabilidad

| Artículo | Temas tratados: Necesidad de estándares | Algoritmo de consenso | Cadena de suministro | Ataque DAO | Transparencia | Autogobierno | Contratos inteligentes | Escalabilidad | Mecanismos de incentivo |
|---|---|---|---|---|---|---|---|---|---|
| Kim [Kim19] | ○ | ◐ | ○ | ○ | ○ | ● | ○ | ○ | ◐ |
| Zhihong y Jie [ZJ19] | ○ | ◐ | ○ | ○ | ○ | ● | ◐ | ○ | ● |
| Saadatmand *et al.* [SLS19] | ○ | ○ | ● | ○ | ◐ | ◐ | ◐ | ○ | ◐ |
| Hütten [Hüt19] | ○ | ○ | ○ | ● | ○ | ○ | ● | ◐ | ◐ |
| Merrill *et al.* [Mer+20] | ○ | ◐ | ○ | ○ | ● | ● | ○ | ○ | ◐ |
| Lim [Lim19] | ● | ○ | ○ | ○ | ○ | ● | ◐ | ○ | ◐ |
| Mariappan [Mar19] | ○ | ◐ | ● | ○ | ○ | ○ | ◐ | ○ | ◐ |
| Benedict [Ben19] | ● | ◐ | ○ | ○ | ◐ | ○ | ○ | ● | ◐ |
| Pelt *et al.* [Pel+20] | ○ | ● | ○ | ○ | ● | ○ | ● | ◐ | ○ |
| Crepaldi [Cre19] | ● | ○ | ○ | ● | ◐ | ◐ | ○ | ○ | ○ |
| John y Pam [JP18] | ◐ | ◐ | ○ | ○ | ◐ | ◐ | ● | ○ | ◐ |

**Tabla 3.8.-** Matriz de análisis de la literatura blanca para los términos relacionados con la gobernanza

Los detalles de cada artículo se explican en las secciones 3.4.2 (almacenamiento de datos), 3.4.3 (latencia) y 3.4.4 (tasa de transferencia).

Finalmente, la tabla 3.8 reúne los 11 artículos revisados que tratan del tema del gobierno de la Blockchain, y muestra los sub-tópicos más comúnmente encontrados en este área. Los detalles de cada artículo se explican en la sección 3.4.5.

### 3.3.2. Literatura gris

Para agrupar por tópico los resultados de la literatura gris no es posible utilizar la misma estrategia que se usó para la literatura blanca, puesto que en la blanca los motores de búsqueda utilizados proporcionaban como meta-información de cada artículo su lista de *keywords*, mientras que para la literatura gris, todo lo que Google proporciona es el título de la página y un breve resumen o *snippet*, ambos truncados a un cierto número de caracteres. Por tanto para la literatura gris sólo podemos basarnos en las palabras que aparecen en el título o en el *snippet*. Mediante un *script* se realizó el conteo de cuántos resultados de la literatura gris contienen la palabra "*Privacy*" ya sea en el título o en el *snippet*, separando además los que tienen información de la fecha de publicación de los que no, y lo mismo se hizo con cada una de las restantes palabras clave ("*Storage*", "*Latency*", "*Troughput* y "*Governance*"). El resultado es la figura 3.5 en la página siguiente, que muestra en tonos saturados el número de resultados que

contienen información de fecha, para cada tópico, y en tono menos saturado los que no tienen información de fecha. Esta gráfica usa los mismos colores para cada tópico que los usados en la literatura blanca (fig. 3.4), con diferentes tonos de verde para los tres tópicos relacionados con la escalabilidad.

Si nos centramos ya únicamente en los resultados de la literatura gris que contienen información del año de publicación, es posible realizar un histograma de número de resultados por tópico y por año, similar al presentado para la literatura blanca. Este histograma para el caso de la literatura gris puede verse en la figura 3.6.

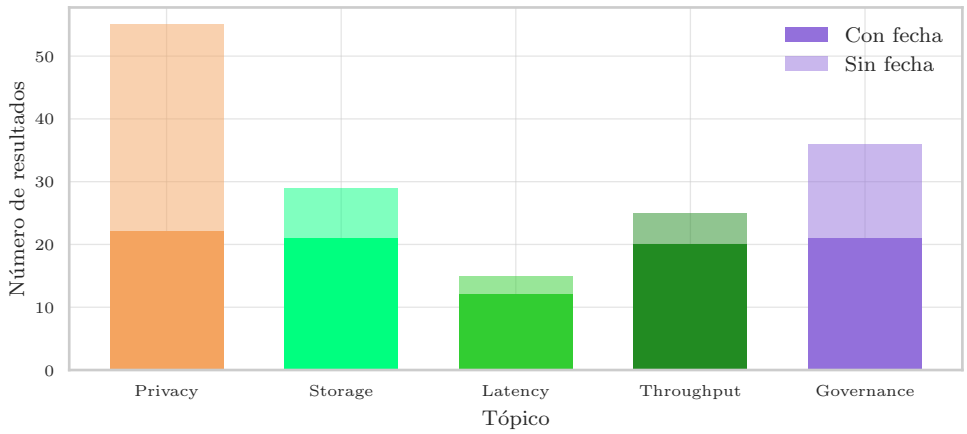


**Figura 3.5.-** Literatura gris: número de resultados por tópico

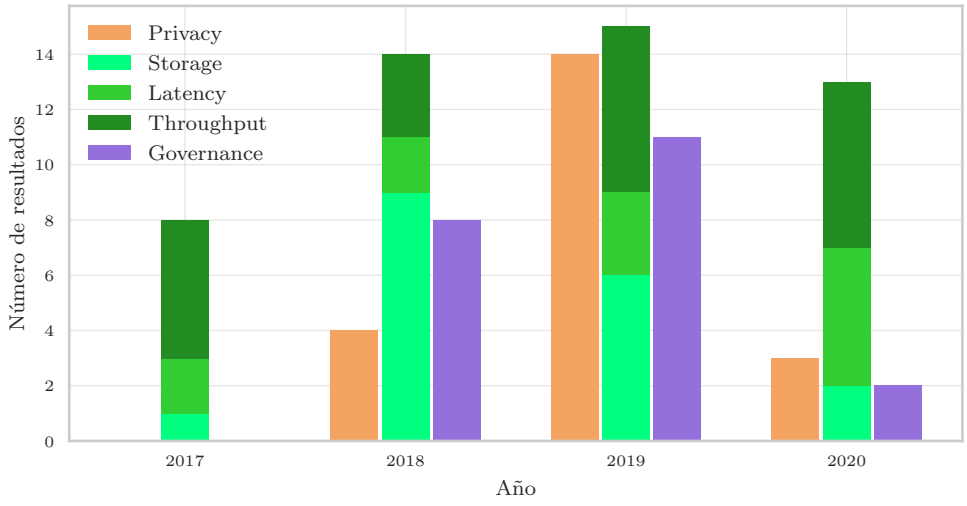


**Figura 3.6.-** Literatura gris: número de resultados por tópico y por año

## 3.4. Situación actual de los tópicos analizados

Es importante tener en cuenta que esta revisión sistemática solo se centra en los 5 tópicos seleccionados después de realizar la revisión de estados del arte previos en el apartado 3.1.2 pero siempre referidos a mejorar la tecnología Blockchain en general, descartando todos los artículos que no aporten una solución que pueda ser utilizada de un modo lo más generalista posible. Por ejemplo, si en una publicación proponen un esquema que podría mejorar la eficiencia en el almacenamiento de datos en un entorno IoT, lo incluimos en nuestra revisión, pero si por el contrario la solución es específica a una arquitectura o caso de uso IoT concreto, sin una posible aplicación más extensa fuera de ese escenario, lo excluiríamos. Además, en tópicos como la gobernanza entre los resultados obtenidos se encuentran numerosos artículos que no corresponden a lo que buscamos, ya que la palabra gobernanza es ambigua en este aspecto y muchos autores proponen utilizar la TBC para mejorar la gobernanza en el mundo real, en lugar de la gobernanza de las redes BC, que es lo que tiene interés para esta revisión.

### 3.4.1. Privacidad de los datos

En esta sección analizaremos los artículos referentes a cómo mejorar la privacidad y la confidencialidad de los datos almacenados en una red BC, incluyendo una descripción de los diferentes artículos revisados e identificando sus aportes en este tópico, restricciones y problemas. En este grupo de publicaciones destaca la inclusión de la TBC en redes IoT ya que la privacidad y seguridad de las mismas es un tema complejo de abordar y es difícil dar una solución eficiente y "estándar". Los mecanismos de seguridad que se utilizan comúnmente en las redes IoT son vulnerables a numerosos ataques que comprometen tanto la seguridad de la red, como la privacidad de los datos. Es por ello que, aprovechando las ventajas que ofrece la TBC como son la descentralización (ver sección 2.4.1) o la inmutabilidad (ver sección 2.4.2), no-repudio, disponibilidad, etc., se pueden usar las TBC como una alternativa a las soluciones centralizadas actuales en la nube.

De entre los artículos que no aportan ninguna técnica ni metodología nueva, sino que aplican técnicas bien conocidas, destacamos el trabajo de Arora *et al.* [Aro+19] y el de Gilda y Mehrotra [GM18]. En el primero se analiza el uso de dos conceptos conocidos como son el revocado de certificado digital (*revoke digital certificate*) y el seguimiento de certificados digitales (*tracking digital certificates*) con el fin de asegurar la privacidad de los datos almacenados en una red BC. En el segundo se propone un *framework* para *Hyperledger Fabric* que permite al administrador de datos de un colegio público dar acceso a determinados datos, previamente autorizados por los tutores legales del alumno, a organismos externos. Dicho acceso se encuentra regulado mediante contratos inteligentes.

El resto de artículos relevantes de la literatura blanca se resume a continuación:

- En octubre de 2018 Loukil *et al.* [Lou+18] proponen un sistema utilizando la TBC enfocado en asegurar la privacidad en cualquier punto de ciclo de vida. El sistema

se compone de una red pública y una red privada IoT (de una vivienda, edificio, etc.). Diferencian entre tres tipos de nodos: públicos, privados y de almacenamiento. A su vez existe un nodo privado (de altas capacidades de procesamiento y almacenamiento) que hace de puerta de enlace entre la red privada y la red pública. Aunque los dispositivos IoT se unen y abandonan la red dinámicamente en función de sus necesidades, la información siempre estará disponible ya que se almacena en la red Blockchain formada por todos los nodos, y todos los nodos tienen una copia de la cadena completa. Toda comunicación entre los nodos privados y los públicos se almacena en una red BC pública, y la comunicación entre los propietarios de los recursos IoT y los dueños de los datos se almacena en una red BC privada. Para asegurar el anonimato utilizan una pareja de claves pública-privada para las comunicaciones, que pueden ser distintas en cada comunicación para asegurar que no se puede seguir una "traza de transacciones" para relacionar transacciones con direcciones y direcciones con identidades de usuarios.

Utilizan una segunda red BC privada para permitir que los propietarios de los datos tengan el control de sus recursos IoT y se apoyan en las características propias de la TBC, como por ejemplo, la inmutabilidad para sostener esta idea. Dependiendo de la cantidad de nodos que posea esta red, a menor cantidad de nodos, mayor susceptibilidad a algunos de los ataques mostrados en la sección 2.4.4. Esta opción podría ser bastante desfavorable para el rendimiento del sistema, ya que se podría implementar otro tipo de estructura más eficiente en este escenario. Además, los nodos de la red deberían ser ligeros para poder operar y sincronizarse después de una caída de la forma más rápida posible. También debemos tener en cuenta la posible sobrecarga de la red en determinados momentos.

Por último, una de las restricciones más notables de este sistema es que las pruebas no se han realizado en una red pública real, sino en la red de pruebas (*testnet*) de Ethereum. Para la red privada utilizaron una red BC local (también de Ethereum) y utilizando muy pocos nodos. Quedaría por analizar el rendimiento del sistema en varios escenarios reales, con más o menos nodos pero utilizando la red principal de Ethereum (*mainnet*).

- Unos meses más tarde, en julio de 2019 Shrestha y Kim [SK19] proponen utilizar criptografía homomórfica en la integración de IoT con BC. Este mecanismo de cifrado que ha ganado popularidad en los últimos años proporciona cierto grado de seguridad y privacidad al sistema ya que los datos se cifran, pero no es necesario descifrarlos para poder operar con ellos, permitiendo así una gran flexibilidad operacional con dichos datos sin comprometer la privacidad de los mismos. Además presentan los retos a los que se enfrenta la investigación en este tópico y sugieren varias direcciones de las futuras investigaciones, como son, el anonimato y la privacidad de los datos, la seguridad de las redes que integran BC con IoT, la escalabilidad, aspectos legales, algoritmos de consenso o el almacenamiento de datos, entre otros.

- En noviembre de 2019 se publicaba el artículo de Wang *et al.* [WGC19] en el cual

se propone un sistema enfocado a redes privadas, en concreto para *Hyperledger*, haciendo uso de oráculos para gestionar eventos y de contratos inteligentes para ejecutar o no ciertas transacciones. Con el uso de firmas digitales consiguen mantener la privacidad de los datos gestionando préstamos de alivio de la pobreza. Al no existir un conector que permita trabajar entre registros centralizados y descentralizados, limita bastante el potencial de este sistema. Además ese conector debería desarrollarse casi *ad hoc* ya que la naturaleza de los sistemas centralizados difiere de uno a otro. Una posible solución sería crear un "estándar" de comunicación entre sistemas, pero esto implicaría la adaptación a este estándar de los sistemas centralizados ya existentes. Además, faltaría estudiar el rendimiento de este sistema en un entorno real donde se aprovechen todas sus características.

- La última publicación relevante encontrada en la revisión sistemática, se publicará en septiembre de 2020, en la cual Wang *et al.* [Wan+20a] presentan un sistema enfocado en la privacidad de los datos de los consumidores en *Open Banking*. Primero utilizan un modelo colaborativo para filtrar los datos por defecto basados en la teoría de empuje (*Nudge Theory*) y por último, haciendo uso de un sistema BC gestionan la privacidad de los datos a compartir. Aplicando la teoría del empuje eliminan el paso en el cual los nuevos usuarios eligen qué datos compartir, haciendo uso de un modelo de filtrado colaborativo (basándose en la edad, educación, posición económica, etc.). Posteriormente clasifican los datos en función de su naturaleza para ajustar los criterios de privacidad a aplicar en cada caso. En principio, los resultados de rendimiento del sistema propuesto son positivos y podrían cubrir de forma holgada las necesidades actuales de la mayoría de bancos, pero solo utilizan hasta 7 nodos y se ve una reducción importante de las transacciones por segundo en comparación con la utilización de 1 a 6 nodos. Esto implica que el sistema es más susceptible a todos los tipos de ataque que tienen que ver con un número de nodos limitado (ver sección 2.4.4).

Respecto a los resultados obtenidos y analizados de la **literatura gris**, un gran número de ellos están relacionados con el Reglamento General de Protección de Datos (RGPD). En Rantos *et al.* [Ran+18] se propone un *framework* para ayudar a la gestión de los datos generados por usuarios en redes IoT cumpliendo el RGPD. Los usuarios pueden formular y gestionar sus políticas de consentimiento. Otro autores como Flanagan *et al.* [Fla+19] recopilan técnicas mediante las cuales podemos conseguir la privacidad de la datos, de menos a más complejas: *on-chain*/*off-chain*, control de accesos basado en roles, pruebas de conocimiento cero y criptografía homomórfica completa, concluyen que la TBC se puede usar como una herramienta dentro de un amplio contexto de privacidad para ayudar conseguir aumentar el grado de transparencia y abre oportunidades de negocio que hace años eran impensables.

Entre los autores que analizan como influye el RGPD en los sistemas BC o que proponen nuevas soluciones de BC que cumplen dicho reglamento se encuentran Lima [Lim18], Truong *et al.* [Tru+18], Rantos *et al.* [Ran+18], Fernandez [Fer19], Wirth y Kolain [WK18], Finck [Fin18], Ibáñez *et al.* [IOS18], Compert *et al.* [CLP18] y Czarnecki

[Cza17]

Autores como Magas [Mag20] ofrecen una visión más crítica de la privacidad de los datos en redes BC, cuestionándose si utilizar TBC es la panacea para la privacidad de los usuarios y concluyen que aún se deben resolver los problemas de la BC con el RGPD, y la inmutabilidad y transparencia, que son 2 caras de la misma moneda.

Dejando a un lado el RGPD, hemos obtenido resultados enfocados en mejorar la privacidad de los datos como Gurevin [Gur19], que recomienda utilizar técnicas avanzadas como *Zero-Knowledge Proof*, o prueba de conocimiento-cero (ZKP) para alcanzar un alto grado de privacidad y concluyen que lo ideal sería utilizar TBC con ZKP y auditorias periódicas. *Orchid* de Bocetta [Boc20], es una red VPN construida en Ethereum que haciendo uso de los *tokens* de la red (ERC-20) permiten comprar ancho de banda de proveedores de servicio. Otros autores como Zyskind *et al.* [ZNP15] proponen utilizar TBC con DHT para almacenamiento de datos seguro y garantizando la privacidad. Las terceras partes utilizan SMPC sin acceder a los datos originales.

Kenyon [Ken19] alerta de un nuevo ataque conocido como envenenamiento de la privacidad (*Privacy poisoning*). El ataque implica cargar datos privados, como nombres, direcciones y números de tarjetas de crédito o cualquier tipo de material ilegal, en una BC, lo que implica un problema para los nodos de la red que almacenan dichos datos. Con relación a este nuevo ataque, la consultora *Gartner* (Omale [Oma19]) apunta que en el año 2022, el 75 % de las redes BC públicas sufrirán un "envenenamiento de la privacidad". En el año 2023, más del 25 % de las implementaciones de prueba de consentimiento impulsadas por el RGPD involucrarán Tecnología Blockchain (TBC), en comparación con menos del 2 % en 2018.

### 3.4.2. Almacenamiento de datos

En este apartado se trata otro de los aspectos más importantes en los que se centra la investigación en la TBC en estos momentos. El almacenamiento de datos como hemos visto en la sección 2.4.3.2, dada la naturaleza de la TBC, es muy problemático, ya que un gran volumen haría que se incrementara de forma muy rápida el tamaño necesario de los nodos de la red, pero por otro lado en muchos casos es necesario que los nodos sean "ligeros". Además del problema del tamaño de los datos almacenados, es necesario que además se ofrezca una alta seguridad en el almacenamiento. Los artículos revisados se centran principalmente en el almacenamiento seguro de datos, en una red BC típica o combinándola con una red IoT, así como a los diferentes ataques a los que están expuestos este tipo de sistemas.

Varios de los artículos en esta categoría no presentan realmente nuevas técnicas o ideas, sino que sistematizan y catalogan información relacionada. En este tipo entrarían los siguientes: Peng [Pen18], Taylor *et al.* [Tay+20] y Zegzhda *et al.* [ZMM18]. En el primero de ellos, publicado en noviembre de 2018, Peng [Pen18] expone las dificultades encontradas en el diseño de un sistema basado en TBC para el almacenamiento seguro de datos. El primero de los problemas es el diseño del algoritmo de consenso que es uno de los elementos clave y más complejos de diseñar de una red BC. La siguiente dificultad es cómo verificar cuando se pierde un bloque y cómo se resuelve una bifurcación de la

cadena. Por último, sugiere la necesidad de extender las transacciones tradicionales de BC para almacenar diferentes tipos de datos y facilitar la búsqueda, análisis y clasificación de los datos. Taylor *et al.* [Tay+20] realizan una revisión de la literatura científica centrándose en el almacenamiento de datos y apuntan que las investigaciones en el campo del almacenamiento y uso compartido de datos con BC de forma segura se enfocan en garantizar que los datos almacenados en una red BC sigan siendo resistentes a cambios no autorizados, en garantizar que las listas *hash* permitan la búsqueda de datos de forma segura y en que en cualquier momento del flujo se puedan comprobar la originalidad de los datos. Por último, Zegzhda *et al.* [ZMM18] comparan las amenazas de seguridad en sistemas de almacenamiento de datos centralizados (en la nube) y descentralizados (utilizando TBC). Concluyen que ambos tipos de sistemas comparten algunas amenazas de seguridad, pero en los sistemas descentralizados hay que añadir amenazas propias de la descentralización aunque sugieren que la mayoría se pueden evitar o combatir aumentando el número de nodos de la red.

Los restantes artículos relevantes encontrados en la literatura blanca se resumen a continuación:

- En 2017, Do y Ng [DN17] proponen *BlockDS* un sistema enfocado en proporcionar un almacenamiento de datos seguro y la posibilidad de realizar búsquedas privadas por palabras clave (cada usuario solo podrá "buscar" entre los datos que posee autorización, ya que los documentos encriptados se almacenan con meta-datos).

  El sistema está compuesto por:

  - **Almacenamiento de datos distribuido:** Se almacenan los datos de forma distribuida, haciendo uso de una red BC, asegurando así la integridad de los datos.
  - **Control de acceso anónimo:** Solo podrán acceder a los datos los usuarios cuyo certificado esté aprobado por el administrador de datos. El nodo BC no sabe la identidad de los usuarios que acceden a los datos.
  - **Búsqueda de palabras clave:** Este componente proporciona al usuario una forma de interactuar con el sistema y solicitar archivos para su descarga a través de palabras clave. Una vez encontrado el archivo podrá descargarlo cifrado desde el almacén de datos distribuido.

- En 2018 Ali *et al.* [Ali+18] y Liu *et al.* [LWL18] presentaron dos sistemas que hacen uso de *Distributed Hash Table*, o tabla hash distribuida (DHT). El primero de ellos, propuesto por Ali *et al.* [Ali+18] se basa en un marco de trabajo (*framework*) que proporciona almacenamiento descentralizado, procesamiento distribuido y capacidades de búsqueda eficiente para PingER project. Actualmente en PingER project almacenan los datos generados en múltiples *Monitoring Agents*, o agentes de monitorización (MAs). Con este *framework* eliminan la necesidad de utilizar un repositorio centralizado ya que sustituyen las rutas hacia los MAs que contiene los datos por transacciones de la red BC. Todos los MAs tienen la misma copia de los datos que hay en la red BC. Entre las ventajas que destacan se encuentran

la sostenibilidad y el aumento de la escalabilidad del sistema y la posibilidad de realizar una recolección y análisis de los datos en tiempo real. También mejorará la supervisión del rendimiento de PingER project que a su vez repercute en una mejora de la calidad del servicio.

La segunda plataforma que hace uso de DHT, propuesta por Liu *et al.* [LWL18], tiene como propósito almacenar y distribuir datos en redes IoT, centrándose en mejorar la escalabilidad y permitiendo a los propietarios de los datos tener un alto control sobre ellos e incluso generar beneficios económicos. La plataforma se compone de 3 capas: la red BC, una red P2P y una red de almacenamiento distribuido. La capa BC se encarga de almacenar y verificar las transacciones y el acceso a los datos, la red P2P está formada por miles de nodos que compiten por formar parte del sistema, y cuando uno de ellos es seleccionado deberá elegir si quiere ser nodo de almacenamiento o nodo *proxy*. Los nodos *proxy* se encargan de proporcionar la capacidad de cómputo necesaria para el cifrado de datos, mientras que los nodos de almacenamiento únicamente se limitan a almacenar los datos cifrados. Ambos reciben beneficios en forma de *tokens* de la red. La última capa es la encargada de almacenar los datos que generan los dispositivos IoT antes de ser cifrados. Entre sus características más importantes encontramos el almacenamiento cifrado basado en DHT y la distribución de datos utilizando re-encriptación. Además separan la capa de datos (P2P) de la capa de control (BC), interaccionando entre redes con técnicas como cifrado simétrico o re-encriptación. Concluyen que una arquitectura distribuida en paralelo ofrece unas grandes ventajas a las redes IoT.

- Ohara [Oha19] desarrolla un sistema para crear puntos de control en aplicaciones BC utilizando la propia red BC como almacenamiento. Además, implementa un simulador con el que obtiene resultados no esperados y concluye que es necesario construir un simulador más realista para realizar experimentos y poder sacar conclusiones.

- En septiembre de 2019, Shala *et al.* [Sha+19] proponen un sistema para calcular el nivel de confianza de cada nodo de una red *machine-to-machine* (M2M), analizando el rendimiento y funcionalidad de los nuevos servicios que proveen. Para asegurar una descentralización real de la red, todos los nodos de la red pueden comprobar a cualquier otro nodo, verificando su funcionalidad y rendimiento. El sistema consta de 3 partes diferenciadas: la primera, *Service Trust Evaluation*, evalúa el servicio que está prestando el nodo; la segunda, *Behavior Trust Evaluation*, evalúa el comportamiento del nodo basándose en la integridad del servicio; y la última, *Task Trust Evaluation*, evalúa la actividad como un *Test Agent* u otras tareas realizadas en la red M2M. Para reforzar este sistema incluyen TBC, ya que se asegura así la integridad de los datos y el no repudio, además de proveer un acceso más seguro y varias posibilidades de gestionar identidades. Cuando un *Test Agent* evalúa a un nuevo nodo o servicio, almacena en la BC el nivel de confianza, el identificador del servicio, información de contacto del servicio y el nombre de usuario del *Test Agent* que ha realizado la evaluación. Esta nueva

transacción llegará a los demás nodos que deberán alcanzar un consenso. Para terminar, comparan el algoritmo de consenso propuesto con otros ya conocidos como son, PoW, PoS, *Ripple*, dPoS, *Tangle* y Nano, obteniendo como resultado que su algoritmo tiene una mayor resiliencia respecto a ataques de transacciones maliciosas. Los experimentos están bastante limitados, ya que fueron ejecutados en máquinas virtuales en lugar de utilizar máquinas reales en diferentes redes y localizaciones geográficas. Puede ser que su algoritmo de consenso funcione muy bien en estas condiciones, pero en un entorno real, con un número elevado de nodos tenga un rendimiento menor.

- A finales del año 2019, Qiao *et al.* [Qia+19] presentan una arquitectura para el almacenamiento de datos utilizando el algoritmo de cifrado asimétrico ElGamal, de modo que pueden evitar la identificación del autor partiendo de los datos, o viceversa. Además, proponen un sistema de transacciones anónimas utilizando el protocolo de firma digital ciega y criptografía de curva elíptica.

- Cai *et al.* [CCY19] proponen un sistema en el que los datos de los usuarios se almacenen de forma descentralizada y puedan ser compartidos entre múltiples aplicaciones. Con ello, prometen garantizar la propiedad y privacidad de los datos de los usuarios. Los datos de los usuarios son almacenados *off-chain* utilizando una combinación entre *InterPlanetary File System* (*IPFS is the Distributed Web* [IPFS15]), o sistema de archivos interplanetario (IPFS) y SMPC. Aseguran además, una baja latencia de la red.

- Roy *et al.* [Roy+19] proponen un enfoque de almacenamiento de datos utilizando la TBC en el cual los usuarios pueden crear esquemas de datos dinámicos y consultas para obtener esos datos. Dicho enfoque se divide en varios componentes. El componente principal es el *motor de datos*, que está formado a su vez de cuatro componentes (institutos, proyectos, esquemas y registros) y que se encarga de todas las actividades relacionadas con la gestión de datos de los usuarios. El siguiente componente llamado *componente de esquemas* es el encargado de mantener una estructura con los datos que van a ser almacenados en el sistema y unas reglas a seguir. Por último, el *componente de registros*, considerado la pieza clave del mantenimiento de los datos, es el responsable de almacenar los datos en base a las reglas establecidas por el componente de esquemas. Entre los retos abiertos destacan la necesidad de integración de este sistema con otros sistemas existentes y un adaptador que asegure la integridad de los datos en cada consulta.

- Ya en 2020, Wang *et al.* [Wan+20b] diseñan un sistema electrónico de almacenamiento de datos basándose en la TBC . El sistema encripta la información utilizando el código Reed-Solomon. Utilizan contratos inteligentes para asegurar la información en la cadena y restringir el acceso. Y por último, incrementan la disponibilidad del sistema, basándose en el modelo RFM mejorado para distribuir la carga en los nodos, lo que aumenta la disponibilidad del sistema. Es sólo un

modelo teórico ya que solo está testeado en una red LAN, el algoritmo de consenso utilizado gasta demasiado tiempo y recursos y la comunicación entre nodos se realiza en texto plano.

- Por último, en abril de 2020 se publicó el trabajo de Fan *et al.* [Fan+20] en el cual se propone un marco de trabajo para el almacenamiento seguro de datos utilizando TBC. Utilizan una estructura bidimensional para la BC. La primera dimensión, llamada *cadena horizontal* se encarga de almacenar la información, y cada bloque de esta cadena tiene a su vez asociada una *cadena vertical*, que constituye la segunda dimensión. Cuando un dato necesita ser actualizado se genera un nuevo bloque con la actualización de esa información pero la actualización se guarda en la cadena vertical. También proponen un algoritmo de consenso para balancear la descentralización y los costes de comunicación denominado *Rotating Multiple Random Sampling*, o muestreo aleatorio con rotación múltiple (RMRS). Cuando es necesario almacenar un nuevo dato en la cadena, el algoritmo de consenso selecciona aleatoriamente $k$ nodos como *"maestros temporales"*, cuya finalidad es generar y difundir el nuevo bloque. El resto de nodos reciben las copias enviadas (el *hash* de los datos) por los maestros temporales, las almacenan temporalmente, las comparan y anotan el número de veces que se repite el mismo bloque. El cliente aceptará el bloque que tenga más de $k/2$ copias. El proceso de consulta de datos sería similar. Por último, proponen dos mecanismos para la protección de datos. El primero de ellos, llamado *"Single-block validation mechanism"*, se encarga de verificar la integridad de los bloques y de corregir los bloques erróneos. El segundo de ellos, denominado *"Periodic Blockchain verification mechanism"*, verifica periódicamente que todos los bloques de la cadena sean correctos, y en caso contrario se ejecutaría el mecanismo anterior para corregir el bloque incorrecto. Aunque los resultados obtenidos en los experimentos realizados son buenos, en trabajos posteriores deberán estudiar cual es el número mínimo de nodos para garantizar que se cumplen todos los requisitos y restricciones del sistema y si existe algún número de nodos a partir del cual se vea mermado el rendimiento.

  El prototipo en este artículo se realizó utilizando una lista enlazada de dos dimensiones, por tanto, no es muy extrapolable a una red BC en el mundo real que maneje una gran cantidad de datos. Según las pruebas de rendimiento de la seguridad del sistema que han realizado, para conseguir una seguridad *"completa"* el número de nodos maestros temporales debe ser mayor o igual al $2pN$, donde $p$ es la probabilidad de que cualquier nodo sea comprometido y $N$ es el número de nodos. A la hora de implementar este sistema se debe garantizar que el número de nodos totales y maestros temporales cumplen las condiciones mostradas anteriormente, ya que en caso contrario se perderían los aportes de este modelo. No se explica en el artículo el mecanismo propuesto para seleccionar nodos maestros temporales que permita generar bloques de forma concurrente sin afectar a la prometida seguridad o consistencia de los datos.

En la **literatura gris** analizada, son múltiples los autores que presentan o estudian las nuevas plataformas de almacenamiento de datos en BC que pretenden ser una alternativa al almacenamiento en la nube tradicional, como AWS o GCP, como son *Storj*, *Siacoin* o *Filecoin* (*How Blockchain Will Disrupt Data Storage* [BlockApps17], Rhodes [Rho18] y Whittle [Whi18], respectivamente).

También cabe destacar la importancia de la técnica conocida como *sharding* en los modelos de almacenamiento de datos, ya que permite un procesamiento paralelo de transacciones de forma simultánea en un gran número de nodos, en función de la granularidad aplicada. Autores como Nanaware [Nan18] añaden que se podría utilizar redes P2P para acelerar la velocidad de descarga de los archivos almacenados en una red BC. Kumar [Kum19] argumenta que la TBC tiene el potencial necesario para renovar toda la industria actual del almacenamiento de datos. Según PR Newswire [Newswire18], se espera que el mercado de almacenamiento de datos alcance beneficios de 88,91 mil millones de dólares en el año 2022, y que por lo tanto, la TBC será fundamental para el éxito de la industria del almacenamiento descentralizado, gracias a la escalabilidad, la eficiencia, la seguridad mejorada y el bajo coste de almacenamiento para el cliente en comparación con los sistemas tradicionales. También se indica que se utilizaría el *swarming* en este tipo de sistemas, concepto que también aparece en CIO Applications [CIOApps19]. Eriksson [Eri19] además de destacar el *sharding* como una técnica necesaria en este tópico, presenta dos plataformas interesantes, como son LevelDB, que se encarga de almacenar el estado completo de la BC, modificando la raíz del árbol Merkle con cada mínimo cambio producido en la red y Filecoin que es una red BC que permite el almacenamiento de datos utilizando IPFS. Otros autores como Tarasenko [Tar20] no son tan optimistas con respecto al almacenamiento de datos en BC, y aportan una visión real y actual de esta tecnología en este tópico, repasan las características de IPFS y concluyen que la TBC no sirve para almacenar grandes cantidades de datos. En su lugar, se utilizaría como intermediario y registro donde aparecen todas las transacciones, para así poder verificar que todo se cumple como debería. Aunque existen soluciones como ILCoin que permite almacenar datos *on-chain*, en este sentido Sturges [Stu20] discute el almacenamiento *off-chain* y *on-chain*, concluyendo que existe una clara ventaja del almacenamiento *off-chain* respecto al *on-chain*, ya que se ahorra en costes, se evitan nuevos problemas de seguridad, se ahorra tiempo de sincronización de nodos y no se satura la red.

### 3.4.3. Latencia

La latencia, junto a la tasa de transferencia efectiva o *throughput* (que veremos en la sección 3.4.4 en la página 95) son dos métricas utilizadas para medir el nivel de escalabilidad (ver sección 2.4.3) de un sistema BC. Mejorar la latencia, el *throughput* y por ende, la escalabilidad de las redes BC es una de las principales vías de investigación abiertas en este campo ya que es un objetivo necesario para conseguir una adopción masiva de cualquier sistema o servicio que este basado en la TBC.

- El primer artículo relevante en el apartado de latencia se publicó en diciembre

de 2019. En él Fan *et al.* [Fan+19] proponen un esquema de sincronización de relojes basado en TBC para utilizar en una red IoT de modo seguro y asegurando una baja latencia. Además, diseñan un nuevo algoritmo de consenso, *Proof Of Stability* adaptado a las características del escenario donde aplicaron su esquema. Utilizan una *"virtual chain"* que no es una BC completa, y la utilizan para lograr alcanzar un consenso escribiendo los ACK en los bloques *"virtuales"* de dicha cadena. Los resultados de la simulación apuntan a que parece interesante utilizar este nuevo enfoque en redes donde sea necesario realizar una sincronización de relojes de la forma más rápida y eficiente posible, pero en otras condiciones no superaría, por ejemplo, al algoritmo de consenso PoS, ni en seguridad (tolerancia a fallos), ni capacidad de escalar, ni en rendimiento, ni en consumo de energía. Por tanto, parece un esquema interesante en el contexto en el que lo plantean, pero fuera de él, acarrearía más problemas que soluciones.

- En el mismo mes, se publicó otro articulo que también combina la TBC con IoT, pero está vez enfocado en la seguridad y el anonimato. Para ello Dorri *et al.* [Dor+19] proponen un nuevo algoritmo de consenso enfocado en mejorar todos estos aspectos, a los cuales los algoritmos de consenso tradicionales no dan tanta importancia. El algoritmo de consenso propuesto es *Distributed Time-based Consensus algorithm*, o algoritmo de consenso distribuido basado en el tiempo (DTC), el cual reduce los retrasos y la sobrecarga al minar un nuevo bloque. Además incluye otro algoritmo, *Distributed Throughput Management*, o gestión distribuida del *throughput* (DTM), que se encarga de asegurar que la tasa de transferencia efectiva de la red no se desvíe mucho de la tasa media en la red. Aunque realizan numerosas simulaciones, donde su algoritmo obtiene un alto nivel de seguridad y anonimato, y reduce el ancho de banda y el tiempo de procesamiento necesarios, todavía sería necesario ejecutar las pruebas de rendimiento en un entorno IoT real para corroborar los resultados obtenidos.

- En junio de 2020, Han *et al.* [Han+20] realizaron un análisis de distintas redes de registro distribuido actuales en términos de rendimiento, analizando la latencia y la tasa de transferencia. Las redes seleccionadas han sido *Hyperledger Fabric*, *Ripple*, *Tendermint*, R3 Corda y HLF. Cada red ha sido desplegada en clústeres desde 2 hasta 32 nodos, utilizando una carga de trabajo similar. Los resultados que han obtenido confirman que los algoritmos de consenso clásicos no son adecuados para grandes redes. Concluyen que como trabajo futuro sería interesante continuar la investigación evaluando qué redes podrían escalar resolviendo el problema del consenso, y que aplicaciones IoT pueden funcionar sin consenso, solo con una simple verificación de "todos" los nodos.

- Finalmente, en julio de 2020 se publicó un artículo en el cual Alrubei *et al.* [Alr+20] analizaron varios sistemas IoT basados en TBC en términos de latencia y rendimiento, además de proponer, implementar y comparar su propio sistema. El sistema propuesto está formado por 16 nodos comunicados entre ellos por conexión inalámbrica y cada uno de ellos tiene un cliente de Ethereum, y un

contrato inteligente que realiza diversas funciones de gestión o control de los nodos. Concluyen que bajo una red 3G no es óptimo utilizar un tiempo entre bloques de 1, 2 o 3 segundos, ya que derivaría en problemas de sincronización. Con conexión Wi-Fi podría implementarse hasta con 1 segundo. Para otro tipo de aplicaciones donde no se necesite obtener los datos en tiempo real se podría utilizar un tiempo alto. También apuntan que es muy importante elegir bien el algoritmo de consenso, para este caso de uso. Con el que mejores resultados se obtienen es con Ethereum y PoA, aunque es muy dependiente de los nodos (que sean de confianza y honestos) y esta solución puede aumentar el grado de centralización de la red. Por último, es muy importante elegir bien el tamaño del bloque, ya que cuánto más grande sea dicho tamaño, mayor tiempo tardarán los dispositivos en realizar los cálculos oportunos. En el futuro, la tecnología 5G ayudará a disminuir la latencia en este tipo de redes, y se podrá integrar una red BC totalmente funcional eliminando así la centralización actual en terceros y asegurando la protección de los datos, tanto del usuario como de las transacciones. Uno de los problemas que tienen es que no todos los nodos consiguen sincronizarse con la cadena de bloques actual, debido a diferentes problemas con el ancho de banda, la calidad de la red en ese momento, etc.

De los resultados obtenidos en la **literatura gris**, varios de ellos se limitan a definir el problema de la escalabilidad o de la latencia en las redes BC, como por ejemplo, [Wil20] o [PlayersMoney19]. Sin embargo, también hay autores que proponen ecosistemas enfocados en la mejora de la latencia y el *throughput*, como *Ark Ecosystem* [Cat18] o una arquitectura como *Blockcique* (Forestier *et al.* [FVL18]) que aplica la técnica de *sharding* en transacciones para aumentar la escalabilidad de la red logrando uso eficiente de la red sumado a un aumento del *throughput* en cientos de transacciones por segundo. Los autores combinan 3 ideas básicas: una estructura de datos multihilo de tipo DAG donde cada bloque referencia al bloque previo de cada hilo, *transaction sharding*, que separa las transacciones en múltiples hilos permitiendo así que se puedan crear bloques de forma independiente en diferentes hilos, y un protocolo de consenso específico, un PoW modificado que permite la creación de bloques de forma paralela.

### 3.4.4. Tasa de transferencia efectiva (throughput)

Las redes BC existentes en la actualidad no destacan precisamente por su alta tasa de transferencia efectiva, y aunque hay algunas orientadas específicamente a mejorar esta tasa comparándose con las redes tradicionales como Harmony [Har20], no han llegado a cuajar entre los usuarios. Es por ello, que por ejemplo, Ethereum plantea tener listo en el tercer cuatrimestre del año 2020 su nueva red llamada *Ethereum 2.0*, la cual utilizará el algoritmo de consenso PoS y para conseguir aumentar la tasa de transferencia efectiva de modo notable utilizará *sharding*.

Existen numerosos enfoques para mejorar el *throughput* o la escalabilidad de un sistema modificando su configuración. Los más relevantes encontrados en nuestro estudio serían:

- Kim *et al.* [Kim+19] modifican la configuración inicial de *RocksDB* para mejorar el rendimiento en operaciones de lectura, además de presentar dos técnicas para lecturas paralelas y evitar el máximo número de lecturas posible. Señalan que un sistema BC requiere millones de transacciones por segundo (TPS) pero que en la actualidad no puede proporcionar un alto *throughput* por sí solo. Uno de los mayores cuellos de botella de los sistemas actuales es la gran cantidad de lecturas aleatorias que son necesarias para la sincronización de los nodos de una red BC. *RocksDB* consigue un equilibrio entre el rendimiento de las lecturas y las escrituras, aunque han observado una infra-utilización de la CPU por parte de *RocksDB*. Como trabajo futuro proponen mejorar la utilización de la CPU por parte de *RocksDB* y extender las pruebas de rendimiento a plataformas en la nube como AWS o GCP. Otra solución a destacar centrada en la red Bitcoin, es la propuesta por Sanka y Cheung [SC18] en la cual los autores presentan un sistema de almacenamiento en caché que almacena los datos de la BC en una caché que permite reducir la carga de trabajo del servidor cuando ésta es elevada. Utiliza un dispositivo FPGA y la técnica *memcached*. Los resultados obtenidos en los experimentos realizados con *Bitcoin Core* apuntan que cuando la caché está activada se mejora sustancialmente el rendimiento, disminuyendo el consumo energético y utilizando menos recursos.

- En julio de 2017, Xu *et al.* [Xu+17] proponen un nuevo algoritmo de consenso enfocado en las necesidades del comercio electrónico. Los nodos tienen dos capas para asegurar un alto *throughput* y permitir transacciones en tiempo real. Estas transacciones se guardan en orden cronológico y el nodo no puede modificarlas, reordenarlas o eliminarlas. Cuando el nodo recibe una transacción, se encarga de verificar que es legítima, y sí lo es la almacena temporalmente en *"micro bloques"*. Además se pueden crear varios *"micro bloques"* a la vez en diferentes nodos lo que aumenta el *throughput* del sistema notablemente. Cuando el micro-bloque tenga el suficiente tamaño o lleve un tiempo determinado siendo temporal, el nodo enviará la cabecera del micro-bloque a todos los nodos validadores de la red. Los nodos validadores crearán los bloques que añadirán a la cadena en función de los micro-bloques recibidos después de comprobar el *hash* de los datos y firmarlos. Los resultados que obtienen en el experimento realizado con 1000 nodos son prometedores, alcanzando una tasa de transferencia efectiva que superaría a la mayoría de plataformas e-commerce existentes en la actualidad y con una baja latencia en comparación con Bitcoin. Aún así, todavía necesitan avanzar en el algoritmo de cara a utilizarlo en un entorno real y con un gran número de transacciones por segundo, incluyendo mejoras en el almacenamiento de bloques y transacciones, en garantizar la consistencia de los datos, etc.

- En 2019 se publicó otro artículo relevante en el ámbito de la mejora de la tasa de transferencia efectiva en la TBC. Xiong *et al.* [Xio+19] presentan un sistema basado en TBC que es más seguro que los DNS tradicionales, muy escalable y con una alta tasa de respuesta (*throughput*). Para conseguir disminuir el grado de cen-

tralización y gestionar mejor los TLDs utilizan una red consorciada o híbrida (ver sección 2.3). Cada país mantiene varios nodos (construidos, gestionados y mantenidos por cada uno de ellos) localizados geográficamente según las directrices de la organización internacional y todos ellos forman el consorcio que gestiona la red BC. El sistema se basa en el esquema de firma digital en anillo propuesto por Liu y Wong [LW05] para asegurar el anonimato y que el proceso de votación sea justo, ya que para registrar un nuevo TLD se debe someter la propuesta a votación entre todos los nodos del consorcio. Además, utilizan un protocolo de fragmentación (*sharding protocol*) para dividir los nodos en "comités" más pequeños de forma uniforme y aleatoria para procesar registros de forma independiente y así aumentar el *throughput* de la red y evitar un bajo grado de descentralización.

Existen tres tipos de nodos según este enfoque:

- **Nodos consorcio:** Son responsables de las solicitudes en el TLD, la gestión de los nodos completos, generar parámetros de sistema y de generar bloques en la cadena principal.
- **Nodos completos:** Cualquier nodo que tenga un copia completa de la cadena principal puede ser un nodo completo, y pueden participar en el proceso de generación de bloques en las subcadenas, entre otras tareas.
- **Nodos ligeros:** sólo almacenan la cadena principal y parte de las subcadenas para conseguir reducir la carga de almacenamiento de los nodos.

Como trabajo futuro está pendiente solventar el almacenamiento redundante en los nodos y mejorar la eficiencia del algoritmo de firma utilizado (*Linkable Ring Signature*, o firma en anillo enlazada (LRS)).

- A finales del año 2019, se publica un artículo en el cual Yu *et al.* [Yu+19] proponen un nuevo algoritmo de consenso para redes públicas denominado *Proof of QoS* (*Quality of Service*) que mejora el *throughput* o tasa de transferencia efectiva y la imparcialidad. Se divide la red en pequeñas regiones, donde cada región elige a un nodo según su *Quality of Service*, o calidad del servicio (QoS) para que sea el "representante" de dicha región. Todos los nodos elegidos en las diferentes regiones se rigen a través de un algoritmo de consenso determinista *Byzantine Fault Tolerance* (Tolerante a fallos bizantinos) (BFT). Gracias a la combinación de estos dos algoritmos de consenso, conseguimos una alta tasa de transferencia efectiva y un entorno más justo para todos los nodos de la red. Además los autores definen varias restricciones que deben cumplirse para que este algoritmo de consenso pueda trabajar de forma segura. La primera de ellas es que los usuarios honestos utilizan un *software* oficial y no modificado, además de incidir en el obvio requisito de no compartir su clave privada públicamente Es importante también que la red dentro de una "región" se encuentre sincronizada en todo momento. Por último, los nodos maliciosos no pueden superar 1/3 del total de nodos en la red, y durante el arranque inicial de la red se presupone que todos

los nodos son de confianza. Los resultados de los experimentos que muestran son esperanzadores, aunque faltaría por ver cómo se comporta en un entorno real. Los autores también destacan el trabajo de Gudgeon *et al.* [Gud+20] en el cual se presenta un nuevo modelo de criptomoneda que tiene como objetivo aumentar la tasa de transferencia efectiva, aunque no proponen ningún nuevo protocolo de consenso.

- Finalmente, en septiembre de 2020 se publicará un estudio realizado por Hosseini Bamakan *et al.* [HMB20] en el que se revisan 12 algoritmos de consenso en términos de rendimiento, se analiza el *throughput* (transacciones por segundo, tiempo por bloque, tiempo de verificación de bloque y tamaño de bloque), la rentabilidad del minado (recompensas por minar, consumo de energía, comisión por transacción y dependencia de *hardware* específico), el grado de descentralización (gobernanza, modelo de permisos y modelo de confianza) y las vulnerabilidades y problemas de seguridad (ataque del doble gasto, ataque del 51 % y ataque Sybil). Los dos algoritmos de consenso que cumplen el mayor número de condiciones son DAG y *Proof-Of-Importance* (PoI), los dos son descentralizados, para redes públicas (sin permisos o *permissionless*), eficientes energéticamente, con una alta escalabilidad, seguros ante los ataques de 51 % y de doble gasto, sin dependencia de un *hardware* específico y con una alta velocidad. El siguiente algoritmo que encontramos en este ranking sería el *Practical Byzantine Fault Tolerance* (PBFT) un algoritmo válido para redes públicas y privadas, pero que no tiene una gran escalabilidad y su velocidad es baja.

De entre los resultados obtenidos de la **literatura gris** que cumplían los criterios de calidad previamente definidos, podemos destacar varios como el de BoogerWooger [BoogerWooger19] en el cual argumentan que utilizar el número de transacciones por segundo para medir el rendimiento de un sistema distribuido es muy ambiguo y conflictivo y en su lugar proponen diferentes métricas como pueden ser el número de transacciones por segundo locales (en el propio nodo), el número de bloques producidos por cada nodo, el tiempo medio que tarda un bloque en ser irreversible o el tiempo de procesamiento de bloques. En Swipe [Swipe19] se utiliza un algoritmo de consenso llamado *Proof-Of-Concept* (POC) mediante el cual han conseguido unas 1100 *Transactions Per Second* (TPS) con un tiempo medio de autorización de 0.5 segundos aplicándolo en el banco central de Brasil. En un intento de aumentar el *throughput*, Blocksplain [Blocksplain18] utiliza la técnica de *sharding*, y Clifford [Cli19] propone utilizar un protocolo denominado *compact blocks* (Core [BitcoinCore16]) que consiste en enviar únicamente el encabezado del bloque, una lista con los identificadores de las transacciones y un conjunto de transacciones que el remitente determine que es poco probable que el destinatario tenga en su poder, en lugar de enviar un bloque con todas las transacciones, reduciendo así el ancho de banda utilizado y por tanto, mejorando el *throughput*.

### 3.4.5. Gobierno

El gobierno o gobernanza de una red BC es uno de los aspectos fundamentales en la TBC y posiblemente el más complejo. En el pasado se han visto ejemplos de prácticas de gobernanza realizadas en algunas redes, como por ejemplo, el caso del ataque DAO (explicado en la sección 2.4.2) que desprenden una imagen bastante negativa de este asunto, ya que pone en jaque la descentralización (ver Sección 2.4.1) y la inmutabilidad "prometidas". La gobernanza de una red se podría definir como el liderazgo equitativo y justo en base a los intereses de todas las partes interesadas y siguiendo las normas impuestas en dicha red. Cuando la gobernanza de la red no es la adecuada, lo más común es que el poder lo sustenten unos pocos usuarios (generalmente los que más recursos tienen) y dominen a su antojo el rumbo de la red. El objetivo de los mecanismos de gobernanza y consenso a implementar debe ser el alinear los intereses de las diferentes partes interesadas de modo que todos ganen, sin perjudicar a ningún otro y respetando las normas de la red.

Una pequeña parte de los artículos revisados tratan de analizar hechos ocurridos que pusieron en jaque a alguna red BC, como puede ser el trabajo de Hütten [Hüt19] en el cual se analiza la gobernanza en la TBC, centrándose en lo ocurrido en la red Ethereum con el ataque DAO, o el análisis que realiza Mariappan [Mar19] sobre el impacto que tiene la TBC en los modelos actuales de negocio y de gobierno, centrándose en la India. En esta cateogría destacamos también el trabajo de Lim [Lim19] en el cual el autor establece tres niveles de gobernanza, aunque sólo aborda el primer nivel, que implica una gobernanza homogénea y ajustada al diseño previo. Utiliza un modelo de simulación computacional para estudiar los enfoques de gobernanza propuestos en la literatura. El autor concluye que es interesante seguir explorando este tipo de simulaciones utilizando motores de juegos (*game engines*) y que la idea de un autogobierno puede ser interesante pero siempre dentro del marco de trabajo (*framework*) de gobernanza adecuado.

Por otro lado, entre los artículos revisados encontramos varios muy teóricos y centrados en demostrar matemáticamente ciertas propiedades, como son el trabajo de Kim [Kim19] en el cual se propone un marco de trabajo teórico no determinista para resolver algunos de los ataques más comunes a los que están expuestas las redes BC y analiza el comportamiento de la red en todo momento para proveer decisiones que se adelanten a los posibles ataques, para así poder evitarlos o minimizarlos. El principal objetivo del modelo es evitar ataques y mantener la red con un alto grado de descentralización. Por su parte Saadatmand *et al.* [SLS19] presentan varios mecanismos para gobernar plataformas (independientemente de la tecnología utilizada), que dicen puede ser útil para la TBC. Presentan tres tipos de configuraciones: vertical, horizontal y modular. Sugieren que puede ser interesante aplicar la configuración vertical en la TBC, ya que muchas redes BC se gobiernan de forma centralizada, sobre todo al principio. En este tipo de metodología existe una amenaza competitiva entre terceros, que implica un mayor compromiso de esos terceros en la red BC. De forma teórica parece una metodología ideal para aplicar en las redes BC pero las previsiones que se puedan realizar en base al modelo teórico pueden no coincidir con los resultados obtenidos tras implementarlo

y testearlo en un entorno real. Por último, el artículo de John y Pam [JP18] tiene por objetivo principal la resolución del problema del diseño de un núcleo de gobernanza mediante el cual se puedan proponer nuevos esquemas o sistemas de gobernanza. Proporcionan una serie de puntos clave para la gobernanza de una red BC basados en un *Complex Adaptive System*, o sistema adaptativo complejo (CAS).

Aparte de los dos grupos anteriores, el resto de artículos revisados en la literatura blanca se enumeran seguidamente, en orden cronológico de publicación:

- En junio de 2019 se publicaron dos artículos muy interesantes en materia de gobernanza: Benedict [Ben19] y Crepaldi [Cre19]. En el primero de ellos, Benedict [Ben19] identifica los retos a los que se enfrentan los sistemas gobernados por una autoridad central, que son la ineficiencia y dificultad para aplicar la regulación acordada, la tendencia a concentrar el poder en un número reducido de usuarios, el alto coste de las jerarquías organizacionales, la alienación social o una estandarización demasiado simple que no tiene en cuenta a todos los sectores de la organización. También identifica los retos a los que se enfrentan los sistemas gobernados de forma descentralizada utilizando cualquier tipo de DLT, que son la aparición de usuarios no comprometidos o estructura de gobiernos oligárquicas, la resistencia de los contratos inteligentes y las DAOs a las sanciones legales y los problemas asociados a la supresión de los puntos de control centralizados que existen en los sistemas tradicionales. Concluyen que es necesario establecer estándares que regulen la gobernanza en DLT, ya que cada vez crece más la demanda de estándares por parte de inversores, usuarios y reguladores. En el segundo artículo, Crepaldi [Cre19] se centra en comparar los sistemas BC con los sistemas legales actuales e indica que a la TBC le hacen falta reglas secundarias o meta-reglas para conseguir solucionar algunos de los problemas actuales a los que se enfrenta la gobernanza en una red BC. La mayoría de redes BC públicas actuales utilizan sistemas de reglas primarias únicamente, sin meta-reglas, por lo que no podrían competir con otros sistemas de gobernanza tradicionales. Existen algunas redes como EOS, *Decred* o *Tezos* que han implementado una especie de meta-reglas parciales. Proponen dos líneas de investigación futuras que son el estudio y el diseño de meta-reglas para redes BC que utilicen procesos iterativos y deliberativos donde se tenga en cuenta la opinión de las distintas partes interesadas, y el desarrollo de un estándar para el diseño de dichas meta-reglas.

- En noviembre de 2019, Zhihong y Jie [ZJ19] proponen utilizar la TBC para tratar de eliminar los problemas actuales en las comunidades virtuales de aprendizaje añadiendo incentivos en forma de *tokens*. Los autores remarcan que los resultados de aplicar la TBC en este campo podrían no ser extrapolables a otro campo diferente. Existen diferentes tipos de incentivos, incentivo en forma de *token* o moneda, con un valor monetario real, en forma de privilegios, en forma de recompensas o en forma de castigo. Proponen mantener un incentivo en forma de *token* elevado al inicio del proyecto para captar más usuarios, y según vaya madurando el proyecto, ir disminuyendo esa cifra. Además se podrían usar esos *tokens* para

formar parte de las decisiones de gobernanza que se toman en la red. Además de la limitación de la extrapolación a otros sistemas basados en TBC, también se encuentra la limitación del estudio del mecanismo de incentivos propuestos, ya que los autores realizan sólo un análisis cualitativo, y el comportamiento de este tipo de comunidades es muy complejo y sería necesario realizar una investigación adicional empírica.

- En febrero de 2020 Merrill *et al.* [Mer+20] plantean que un protocolo de gobernanza de una red BC debería tener transparencia (todos los miembros de la red pueden seguir el proceso de gobernanza), imparcialidad (depende del objetivo de la red), aplicación automática (cuando se toma una decisión, se ejecuta de forma automática), flexibilidad (debe permitir realizar pequeños ajustes de configuración) y puntualidad (en casos graves, se debería llegar a un consenso de la forma más rápida posible). Además, analizan el modelo de recompensa por bloqueo de *tokens* (la base del protocolo PoS), que permite detectar el nivel general de apoyo a una medida concreta o si un grupo de nodos tiene demasiado poder. Concluyen que no existe ningún otro protocolo en la actualidad que mejore los resultados.

- Por último, destacamos el artículo más reciente de los revisados, y posiblemente el más completo, Pelt *et al.* [Pel+20] en el cual los autores realizan una importante labor de investigación y recopilación para definir la gobernanza de una red blockchain. Definen que la gobernanza es el conjunto de medios utilizados para conseguir que las partes interesadas de la red estén coordinados, controlados y vayan en la misma dirección. Dividen la gobernanza de una red BC en 6 dimensiones y 3 capas. A continuación se presenta una enumeración y breve descripción de cada dimensión:

  1. **Formación y contexto:** Se refiere a todos los aspectos que definen la red, como puede ser, su tipo de lanzamiento, su ideología, tipo de licencia utilizada, etc.
  2. **Roles:** Los usuarios de la red se dividen en roles, como pueden ser usuarios "base", desarrolladores y mineros. Además, se describen las estructuras jerárquicas observables entre ellos, las responsabilidades asignadas a los roles y si son responsables o no de sus acciones.
  3. **Incentivos:** Se refiere a que tipo de incentivos, y en qué cantidad, se proporciona a los usuarios dependiendo de su rol, además de cómo se financian los desarrolladores y por qué los nodos de la red querrían participar.
  4. **Afiliación:** Esta dimensión se enfoca en la forma en la que cada rol participa en la red o en como puede unirse y participar un nuevo nodo en la red.
  5. **Comunicación:** Se refiere a todo tipo de comunicación tanto fuera (*off-chain*) como dentro (*on-chain*) de la cadena existente entre los usuarios de la red. Incluye las herramientas de comunicación disponibles, así como los sistemas de coordinación o de seguimiento, y de reuniones o conversaciones informales.

6. **Toma de decisiones:** Se refiere a cómo se toman y se monitorizan las decisiones tomadas en las 3 capas (definidas en el siguiente párrafo). Incluyen los mecanismos de votación disponibles, los procesos de decisión, el mecanismo de consenso utilizado y los procedimientos para resolver los conflictos que puedan surgir en la red.

En cuanto a las tres capas que se definen en este artículo pueden verse como ordenadas "de fuera a adentro", siendo la primera capa la más "externa" y de alto nivel y la tercera capa la más "interna" y de bajo nivel:

1. **Comunidad fuera de la BC (*off-chain*):** En esta capa se incluyen todos los asuntos que tienen lugar en el mundo real, muestra cómo se define un proyecto de manera más general e intenta establecer vínculos de la comunidad con las capas inferiores (desarrollo *off-chain* y protocolo *on-chain*).
2. **Desarrollo fuera de la BC (*off-chain*):** Se refiere a todos los aspectos que tienen que ver con la gobernanza, que se encuentran en el mundo real y que están relacionados con cualquier aspecto de desarrollo o mantenimiento del software.
3. **Protocolo dentro de la cadena (*on-chain*):** En esta última capa se incluyen todos los procesos de toma de decisiones que ocurren *on-chain*, y todos los mecanismos de votación y reglas implementadas en la red BC.

Para finalizar, los autores identifican todos los elementos del *framework* propuesto (las 6 dimensiones y las 3 capas) en las redes Ethereum y EOS. Como trabajo futuro indican que sería interesante comprobar si seguir el marco de trabajo que proponen es indicativo de un "buen gobierno", además de la posibilidad de ampliar el marco a redes públicas y privadas o incluso añadir ampliaciones al *framework* propuesto.

Los esfuerzos de la **literatura gris** en el tópico de la gobernanza de redes BC se centran en clasificar los diferentes tipos de gobernanza que identifican, como por ejemplo, Leimgruber [Lei18] donde se explica la diferencia entre la gobernanza *off-chain* y *on-chain*, o el trabajo de Voight [Voi19] en el que expone 5 modelos de gobernanza posibles para redes BC: Democracia liquida, voto cuadrático, futarquía, registros seleccionados de *tokens* y fundaciones, consejos o asociaciones. En Blockchain Consultants [Blo19] se presenta un protocolo llamado *PeerAssets* (sacado de Peerchemist [Peerchemist16]), y se argumenta que dada naturaleza de la BC es posible utilizar un modelo de seguridad *Proof-Of-Timelime* (PoT), o lo que es lo mismo, un modelo FIFO. Además, utilizan el mecanismo *Pay-to-TagHash* (P2TH) para conseguir una búsqueda eficiente de transacciones basado en etiquetar de forma determinista las direcciones. Por último, señalan que las ventajas con respecto a otros protocolos similares son el control que tienen sobre la distribución de activos, y el sistema integrado de pagos mediante votaciones y dividendos.

También encontramos algunos resultados como el de Strauss [Str18] donde se indica que una buena gobernanza de las redes DLT garantiza que existan los mecanismos adecuados para llegar a un consenso y que existan los incentivos adecuados para mantener este consenso en el futuro; o el de Smits [Smi18] que señala los 4 pilares básicos que debe tener el gobierno de una red BC: transparencia, integridad, rendimiento efectivo y colaboración. Scribani [Scr18] señala los retos a los que se enfrenta la gobernanza en redes BC, bajo su punto de vista, que son la escalabilidad, la privacidad y la interoperabilidad entre redes BC.

## 3.5. Respondiendo a las preguntas de investigación

En este apartado se dará respuesta a las preguntas de investigación planteadas en la sección 3.2 en la página 71 en base al estado del arte sistemático realizado en este trabajo fin de Máster.

- **PI1**: ¿Cuáles son los últimos avances en la investigación sobre TBC en la actualidad, en particular en los campos de privacidad de los datos, la capacidad de almacenamiento de datos, la mejora de latencia y rendimiento y los mecanismos de gobernanza?

  Para dar respuesta a esta pregunta expondremos cada uno de los tópicos por separado con sus respectivos avances.

  - **Privacidad de los datos:** Destacamos el uso de dos cadenas *blockchain* con el objetivo de mejorar la privacidad de los datos (aunque todavía en un estado embrionario), el uso de criptografía homomórfica, que dota de mayor privacidad al sistema ya que permite realizar operaciones con datos cifrados, sin necesidad de descifrarlos en ningún momento del proceso, aumentando también el grado de confianza de los usuarios de la red. Desde su aplicación en 2018, el Reglamento General de Protección de Datos (RGPD) ha sido protagonista de numerosos elementos de literatura gris. Aunque en un principio muchos autores creían que la tecnología BC no cumpliría el nuevo reglamento, la realidad es que todas las redes *blockchain* pueden cumplirlo de forma relativamente sencilla. También destacar los últimos avances en la utilización de técnicas como ZKP o DHT para mejorar la seguridad y privacidad de los datos, y la aparición de un nuevo tipo de ataque de envenenamiento de la privacidad.
  - **Almacenamiento de datos:** Los últimos avances en el almacenamiento de datos están estrechamente relacionados con algunos avances del tópico anterior. Algunos autores en este tópico refuerzan la idea de utilizar DHT, mejorando la escalabilidad de la red, y la privacidad y seguridad de los datos almacenados. A la vez, toma fuerza la idea de utilizar varias redes en paralelo para mejorar las ventajas que ofrece la utilización de TBC, sin aumentar de forma notable sus desventajas, ya sea utilizando redes P2P u otra red

BC. También surge la necesidad de crear estándares para conseguir una mayor interoperabilidad entre redes BC o sistemas tradicionales. Por último, recalcar el enfoque novedoso de utilizar una doble cadena para conseguir un aumento del rendimiento en redes BC que necesiten almacenar grandes cantidades de datos de forma efectiva.

En la literatura gris, se destaca el uso y estudio de las técnicas de *sharding* y *swarming* para mejorar el rendimiento de redes BC orientadas al almacenamiento de datos. Además, se discute sobre la forma más adecuada de almacenar los datos, entre *off-chain* y *on-chain*, concluyendo que ofrece mayores ventajas el almacenamiento *off-chain* y puede conservar las mismas garantías que el almacenamiento *on-chain* gracias a diferentes mecanismos y técnicas que garanticen, por ejemplo, la no modificación de los datos.

- **Latencia y *throughput*:** La investigación en el tópico de la latencia se centra en el desarrollo de algoritmos de consenso que reduzcan la latencia de la red BC, y en la correcta elección del tamaño del bloque con el mismo objetivo. Respecto al *throughput* existen numerosas propuestas de sistemas que mejoran la tasa de transferencia efectiva de la red, pero aún en un estado embrionario. Algunas de las soluciones incluyen la utilización del sistema *memcached*, dispositivos FPGA, o técnicas como el *sharding*. Otros autores se enfocan en el diseño de nuevos algoritmos de consenso que mejoren el *throughput*.

  Las tendencias de la literatura gris en este tópico se dirigen hacia una mejora tanto de la latencia como del *throughput* haciendo uso de técnicas como *sharding*, estructuras de datos como DAG, nuevos algoritmos de consenso o protocolos como *compact blocks*.

- **Gobierno de la red BC:** La gobernanza quizás es uno de los tópicos clave y más complejos de una red BC. Sin duda, gran parte de los artículos revisados coinciden en la necesidad de estándares sobre gobernanza de redes BC y el desarrollo de meta-reglas completas para poder competir con el resto de sistemas, pero numerosos investigadores tratan este tema desde un punto de vista excesivamente teórico. De entre los autores que intentan "estandarizar" las características que debería tener un protocolo de gobierno de una red BC, todos coinciden en que debe ser transparente, imparcial, flexible, automático y rápido (bajo tiempo de respuesta ante incidentes).

- **PI2**: ¿A qué retos se enfrenta la investigación en estos tópicos?

  Para dar respuesta a esta pregunta expondremos cada uno de los tópicos por separado con sus respectivos retos.

  - **Privacidad de los datos:** Los principales retos en este tópico están íntimamente relacionados con las redes IoT. Estos retos incluyen mejora de la seguridad en general, la transparencia sin olvidar la privacidad de los datos, técnicas como la criptografía homomórfica, ZKP o DHT. Además, un

reto abierto y muy preocupante es el crecimiento de ataques conocidos como envenenamiento de la privacidad.

- **Almacenamiento de datos:** Entre los retos a los que se enfrenta el almacenamiento de datos encontramos la mejora de la interoperabilidad entre redes BC o sistemas externos, y la necesidad de estándares, además de una preocupación por la alta disponibilidad de los datos, la integridad de los mismos o el anonimato.
- **Latencia y *throughput*:** Los principales retos en este tópico también están íntimamente relacionados con las redes IoT. Entre ellos se encuentran el desarrollo de nuevas técnicas que mejoren el almacenamiento redundante o nuevos algoritmos de consenso centrados en mejorar la escalabilidad. También es importante destacar los retos que incluyen una mejora de la eficiencia energética, ya que es muy necesaria en sistemas IoT.
- **Gobierno de la red BC:** Uno de los mayores retos a los que se enfrenta la investigación en este tópico es la necesidad de estándares, ya sean metareglas como proponen algunos autores, o estándares genéricos que permitan la creación de nuevos protocolos o algoritmos de consenso centrados en respetar los puntos claves del gobierno en una red BC. Los dos últimos retos a destacar, son la mejora de los incentivos y el desarrollo o mejora de mecanismos de autogobierno.

- **PI3: ¿Qué tendencias se pueden inferir?**

  Después de analizar los resultados obtenidos se puede ver claramente cómo las redes IoT se encuentran muy relacionadas con la TBC, y plantean numerosos retos en los tópicos investigados. Esto podría deberse a la creciente utilización de dispositivos IoT, tanto en el ámbito doméstico, como industrial, ciudades inteligentes, etc. Unido a esto cobran mayor importancia los problemas propios de las redes y dispositivos IoT, como son la falta de potencia computacional, la imposibilidad de almacenar grandes cantidades de datos, los riesgos de seguridad a los que se enfrentan, tanto físicos como cibernéticos, etc. lo que posiciona a la TBC entre las posibles soluciones a la mayoría de estos problemas, si se resuelven con éxito los problemas pendientes.

  Además, es destacable el creciente interés por la técnica conocida como *sharding* que utilizan y exploran las posibilidades dentro de la TBC numerosos autores de varios de los tópicos elegidos y tanto en la literatura blanca (aunque en menor medida), como en la gris. También podría tener cabida la técnica conocida como *swarming*, que aunque tiene una menor presencia en los artículos analizados, podría ser de gran utilidad en este tipo de sistemas.

  Por último, son numerosos los autores que señalan la necesidad de estándares en la TBC, tanto de diseño de nuevos algoritmos de consenso, como de protocolos o mecanismos que permitan una mayor interoperabilidad entre sistemas BC y no

BC, así como la necesidad de regular legalmente algunos aspectos de la tecnología, como por ejemplo en la privacidad de los datos.

Capítulo 4

# Conclusiones y trabajo futuro

En este último capitulo se exponen las conclusiones más relevantes a las que se ha llegado después de realizar el presente documento, además se indican las posibles líneas de trabajo futuras en base a los resultados de esta investigación.

Se han cumplido los dos objetivos fundamentales de este trabajo. Por un lado se ha proporcionado una visión teórica de la TBC en la que se explica su funcionamiento básico, las primitivas criptográficas necesarias para asegurar sus características, su uso en el escenario tradicional de las criptomonedas así como otros posibles usos en otros ámbitos, y se han listado las críticas y problemas de seguridad más importantes que atañen a los sistemas basados en TBC. Además, se ha realizado un estudio del arte sistemático, comenzando por una recopilación inicial de los estados del arte previos, su lectura y catalogación, para obtener dentro de este amplio campo un subconjunto de palabras clave identificadas como tendencias importantes en los estudios del arte previos, para después realizar nuestro propio estudio sistemático (que incluye literatura blanca y gris) centrado en esas palabras clave.

Para este estudio se ha realizado una exhaustiva búsqueda en los motores de búsqueda científicos más importantes con el fin de obtener el listado de artículos potencialmente relevantes en relación a estos temas, así como en buscadores más genéricos como Google para obtener también resultados de la llamada "literatura gris", que no es literatura científica revisada por pares, pero que suele contener también avances importantes en la materia. La larga lista de resultados obtenida ha sido filtrada con una serie de criterios basados en la calidad y en métodos propuestos en otros estudios, para reducirla a los resultados realmente relevantes, los cuales se han leído, resumido y clasificado.

En base a este trabajo se han podido detectar una serie de tendencias y retos en las tecnologías BC que se han enumerado en la sección 3.5 y de los cuales los más importantes serían:

- Es compartida por muchos autores la preocupación por la **falta de estándares** que asienten las bases para el desarrollo de nuevos protocolos, algoritmos de consenso, mecanismos de incentivo, etc. y que puedan dar lugar a una mayor interoperabilidad entre sistemas.
- Es notable **el interés y la sinergia existente entre la TBC y las redes IoT**, ya que gran parte de los artículos revisados se centran en mejorar algún aspecto

de la TBC pero situados en el escenario del IoT o incluso el IIoT.

- Para **mejorar la privacidad de los datos** en este tipo de sistemas están integrando TBC técnicas como la criptografía homomórfica, estructuras de datos como *Distributed Hash Table*, o tabla hash distribuida (DHT) o protocolos de tipo ZKP.

- Además, observamos una clara tendencia a introducir **técnicas "heredadas" de las bases de datos tradicionales** como son el *sharding*, o más recientemente el *swarming*, ya que tienen un gran potencial para mejorar la TBC en términos de latencia, *throughput*, rendimiento al almacenar y consultar datos, etc.

- En términos de gobierno, en general, los esfuerzos de los investigadores se centran en **proponer estándares o guías** que permitan establecer las bases para gobernar este tipo de sistema, garantizando unos criterios de calidad mínimos.

Es lógico que debido a la limitación en el número de motores de búsqueda científicos utilizado no se muestre la totalidad de artículos científicos de los últimos dos años relacionados con los tópicos estudiados. Por ejemplo, puede resultar curioso el amplio interés que despiertan técnicas como el *sharding* en la literatura no científica, pero sin embargo, pocos de los estudios científicos analizados hacen uso de esta técnica para la BC, bien conocida sin embargo en el ámbito de las bases de datos tradicionales. Esta discrepancia entre la literatura blanca y la gris podría deberse a que muchas técnicas se desarrollan y publican en línea antes de enviarse a revistas, y teniendo en cuenta además los plazos de revisión y publicación en éstas últimas, podríamos estar ante un tópico que vaya a ser tendencia en el futuro en la literatura blanca. No obstante quizás no sea ésta la causa de la discrepancia. Podría merecer la pena un estudio más detallado de este tópico

Coincidiendo con los retos identificados en este trabajo fin de Máster, la consultora Gartner [Gartner19] señala también la mala escalabilidad e interoperabilidad de las redes BC en la actualidad, e indican que estos aspectos mejorarán sustancialmente al trabajar en sinergia con redes IoT o la inteligencia artificial. En la citada referencia se prevé que en el año 2023 los sistemas basados en TBC alcanzarán una alta escalabilidad, permitiendo una gran adopción por parte de grandes empresas.

Como trabajo futuro, sería interesante ampliar la revisión sistemática realizada en este trabajo fin de Máster para cubrir un mayor número de motores de búsqueda científicos y así descubrir nuevos artículos (tanto por qué se publicaron después de la fecha de realización de este trabajo, o por que no estaban incluidos en los motores de búsqueda seleccionados inicialmente), con el fin de reforzar los retos y las tendencias inducidas de la lectura de los artículos científicos revisados o incluso poder obtener nuevas conclusiones relevantes.

Por último, concluir que este trabajo fin de Máster podría dar lugar al inicio de una Tesis doctoral en el ámbito de la Tecnología Blockchain (TBC) y centrada en alguna de las líneas futuras expuestas en este mismo documento, ya que es una tecnología con mucho potencial y que aún se encuentra en un estado incipiente, sobretodo, cuando

se compara con sistemas tradicionales de uso masivo, como pueden ser *VISA*, GCP o AWS.

Apéndice A

# Glosario

## Acrónimos

**ACK** Acuse de recibo, o *acknowledgement* (vid. pág. 94)

**API** interfaz de programación de aplicaciones, o *Application Programming Interfaces* (vid. pág. 73, 119)

**ASIC** *Application-Specific Integrated Circuit*, o circuito integrado para aplicaciones específicas (vid. pág. 25)

**AWS** *Amazon Web Services* (vid. pág. 93, 96, 109)

**BC** Blockchain (vid. pág. 2–4, 7–50, 52, 54–59, 61–70, 73, 74, 83, 85–90, 92–97, 99–108, 114–122)

**BCoT** *Blockchain of Things* (vid. pág. 60)

**BDA** *Big Data Analytics* (vid. pág. 61)

**BFT** *Byzantine Fault Tolerance* (Tolerante a fallos bizantinos) (vid. pág. 97, 120)

**BGP** *Border Gateway Protocol*, o protocolo de puerta de enlace de frontera (vid. pág. 36, 61, 121)

**BIoT** *Blockchain-Based Internet of Things* (vid. pág. 52, 54, 56, 60, 70)

**CAS** *Complex Adaptive System*, o sistema adaptativo complejo (vid. pág. 100)

**DAG** *Directed Acyclic Graph*, o grafo acíclico dirigido (vid. pág. 47, 48, 95, 98, 104, 119)

**DAO** *Decentralized Autonomous Organization*, o organización autónoma descentralizada (vid. pág. 39, 99, 114, 116)

**DDoS** *Distributed Denial of Service*, o ataque distribuido denegación de servicio (vid. pág. 34, 37, 56)

**DHT** *Distributed Hash Table*, o tabla hash distribuida (vid. pág. 88–90, 103, 104, 108, 119)

**DLT** *Distributed Ledger Technologies*, o tecnologías de registro distribuido (vid. pág. 1, 5, 47, 48, 73, 100, 103)

**DNS** *Domain Name System*, o sistema de nombres de dominio (vid. pág. 96, 118)

**dPoS** *Delegated Proof of Stake* (vid. pág. 6, 24, 26, 31, 57, 91, 117)

**DTC** *Distributed Time-based Consensus algorithm*, o algoritmo de consenso distribuido basado en el tiempo (vid. pág. 94)

**DTM** *Distributed Throughput Management*, o gestión distribuida del *throughput* (vid. pág. 94)

**ECDSA** *Elliptic Curve Digital Signature Algorithm*, o algoritmo de firma digital de curva elíptica (vid. pág. 10, 13, 15)

**edDSA** *Edwards-curve Digital Signature Algorithm* (vid. pág. 10, 13)

**ERC** *Ethereum Requests for Comments*, o solicitud de comentarios para Ethereum (vid. pág. 88, 117)

**EVCE** *Electric Vehicles Cloud and Edge* (vid. pág. 67)

**EVM** *Ethereum Virtual Machine*,o máquina virtual de Ethereum (vid. pág. 39, 40, 121)

**FCA** *Formal Concept Analysis*, o análisis formal de conceptos (vid. pág. 62)

**FIFO** *First In First Out* (vid. pág. 102)

**FPGA** *Field-Programmable Gate Array*, o matriz de puertas lógicas programable en campo (vid. pág. 96, 104)

**GCP** *Google Cloud Platform* (vid. pág. 93, 96, 109)

**GT** *Straussian Grounded Theory*, o teoría fundamentada, o muestreo teórico (vid. pág. 62)

**HLF** *Hyperledger Fabric* (vid. pág. 94)

**ICO** *Initial Coin Offering*, u oferta inicial de moneda (vid. pág. 118)

**IIoT** *Industrial Internet of Things*, o Internet Industrial de las cosas (vid. pág. 8, 63, 64, 66, 67, 108)

**IoT** *Internet of Things*, o Internet de las cosas (vid. pág. 8, 44, 47, 52, 54–57, 59–64, 66–70, 85–88, 90, 94, 104, 105, 107, 108, 118)

**IOV** *Internet Of Vehicles*, o Internet de los vehículos (vid. pág. 60)

**IPFS** *InterPlanetary File System* (*IPFS is the Distributed Web* [IPFS15]), o sistema de archivos interplanetario (vid. pág. 46, 91, 93)

**LAN** *Local Area Network*, o red de área local (vid. pág. 92)

**LPoS** *Leased Proof-of-Stake* (vid. pág. 25)

**LRS** *Linkable Ring Signature*, o firma en anillo enlazada (vid. pág. 97)

**M-commerce** *Mobile commerce*, o comercio móvil (vid. pág. 66)

**M2M** *machine-to-machine* (vid. pág. 90)

**MA** *Monitoring Agent*, o agente de monitorización (vid. pág. 89)

**MAC** *Message Authentication Code*, un caso de aplicación de una función *hash* que usa además una clave para garantizar el origen (vid. pág. 10)

**MD5** *Message-Digest Algorithm 5* (vid. pág. 11)

**MDC** *Modification Detection Code*, un caso de uso de una función *hash* para detectar alteraciones de un mensaje (vid. pág. 11)

**MEC** *Multi-access Edge Computing* (vid. pág. 60, 61)

**MITM** *Man-In-The-Middle*, u hombre en el medio (vid. pág. 62)

**NIZK** *Non-interactive Zero-Knowledge proof*, o prueba de conocimiento-cero no interactiva (vid. pág. 58)

**OTS** *One-Time Settlement*, o liquidación única (vid. pág. 38)

**P2P** *Peer-to-Peer* (vid. pág. 35, 36, 54, 59, 60, 90, 93, 103, 114, 116)

**P2TH** *Pay-to-TagHash* (vid. pág. 102)

**PBFT** *Practical Byzantine Fault Tolerance* (vid. pág. 26, 57, 98)

**PHR** *Personal Health Records* (vid. pág. 42)

**PingER project** *Ping End-to-end Reporting* (vid. pág. 89, 90)

**PoA** *Proof of Authority* (vid. pág. 26, 34, 95)

**PoB** *Proof of Burn* (vid. pág. 25)

**POC** *Proof-Of-Concept* (vid. pág. 98)

**PoET** *Proof of Elapsed Time* (vid. pág. 25)

**PoI** *Proof-Of-Importance* (vid. pág. 98)

**PoQ** *Proof of QoS* (*Quality of Service*) (vid. pág. 97)

**PoS** *Proof of Stake* (vid. pág. 6, 23–25, 31, 34, 35, 57, 91, 94, 95, 101, 117, 119)

**PoT** *Proof-Of-Timelime* (vid. pág. 102)

**PoW** *Proof of Work* (vid. pág. 6–8, 12, 22–26, 28, 29, 31, 35, 37, 39, 40, 48, 57, 58, 64, 67, 91, 95, 114, 119, 120)

**PRISMA** *Preferred Reporting Items for Systematic reviews and Meta-Analyses* (vid. pág. 59)

**QoS** *Quality of Service*, o calidad del servicio (vid. pág. 97, 120)

**RFID** *Radio Frequency Identification*, identificación por radiofrecuencia (vid. pág. 62, 68)

**RFM** *Recency, Frequency and Monetary*,o frescura, frecuencia y valor monetario (vid. pág. 91)

**RGPD** Reglamento General de Protección de Datos (vid. pág. 58, 59, 87, 88, 103)

**RMRS** *Rotating Multiple Random Sampling*, o muestreo aleatorio con rotación múltiple (vid. pág. 92)

**RSL** Revisión Sistemática de Literatura (vid. pág. 51)

**SMPC** *Secure Multi-Party Computing*, o protocolo seguro (vid. pág. 88, 91)

**SSL** *Secure Socket Layer*, capa de *sockets* segura (vid. pág. 66)

**TBC** Tecnología Blockchain (vid. pág. 1–52, 54–60, 62–71, 74, 85–94, 96, 99–101, 103, 105, 107, 108, 115, 119, 122)

**TFM** Trabajo Fin de Máster (vid. pág. 4)

**TLD** *Top-Level Domain*, o dominio de nivel superior (vid. pág. 97)

**TPS** *Transactions Per Second* (vid. pág. 96, 98)

**UAV** *Unmanned Aerial Vehicle* o vehículo aéreo no tripulado (vid. pág. 60)

**VPN** *Virtual Private Network*, o red privada virtual (vid. pág. 88)

**zk-SNARK** *Zero-Knowledge Succinct Non-Interactive Argument of Knowledge* (vid. pág. 57)

**ZKP** *Zero-Knowledge Proof*, o prueba de conocimiento-cero (vid. pág. 88, 103, 104, 108)

## Términos y conceptos

**árbol binario** Es un tipo de estructura de datos árbol en la que cada nodo tiene únicamente dos nodos hijos (vid. pág. 13, 114)

**árbol Merkle** Es un tipo de árbol binario que utiliza funciones hash para "comprimir" la información de cara a la verificación de pertenencia de un nodo al árbol, ya que todo el árbol se resume en un único *hash* denominado *Merkle root* o raíz de Merkle. Para más detalles véase la sección 2.2.1.2 (vid. pág. 7, 8, 13, 19, 93)

**ataque DAO** Se denomina "ataque DAO" al ataque sufrido en el proyecto DAO en 2016 (proyecto desplegado en la red Ethereum) y que derivo en el robo de 3.6M de Ether y la división de la comunidad de la red que conllevó a un *hard-fork* diviendose la cadena en dos partes, los que no aceptaron el robo (Ethereum) y los que continuaron con la transacción del robo efectiva (Ethereum Classic) (vid. pág. 32, 100)

**ataque del 51 %** Nombre genérico de los ataques que requieren que el atacante posea más de la mitad del poder de cómputo de la red (vid. pág. 23, 30, 34, 69, 98)

**ataque Sybil** Es un tipo de ataque muy común en redes P2P con sistemas de reputación, ya que su objetivo principal es crear múltiples identidades para conseguir aumentar la reputación de un determinado usuario de la red. Para más detalles véase la sección 2.4.4 (vid. pág. 22, 26, 34, 35)

**Bitcoin** Primera red BC creada por una persona o grupo de personas bajo el pseudónimo de "Satoshi Nakamoto" en 2009. Utiliza el algoritmo de consenso *Proof of Work* (PoW). También se denomina bitcoin a la moneda virtual e intangible asociada a dicha red (vid. pág. 1, 2, 4, 6–8, 10, 15, 17, 20, 22, 23, 28, 33, 35, 37, 45, 56, 57, 65, 68, 96, 115, 116, 119, 120, 122)

***blockchain*** o cadena de bloques es una estructura de datos compuesta por bloques que son enlazados entre sí mediante directivas criptográficas que aseguran su inmutabilidad. Todos los bloques son añadidos a la cadena de forma definitiva después de llegar a un consenso y añadir cierto número de bloques posteriores a la misma. La cadena de bloques se almacena de forma distribuida entre todos los nodos que forman la red BC (vid. pág. 4, 8, 103)

**Blockchain 1.0** La primera generación de redes BC o *Blockchain 1.0* engloba a las primeras redes BC cuya única aplicación era las criptomonedas. Esta generación ha sido inaugurada por la red Bitcoin (vid. pág. 2, 8)

**Blockchain 2.0** En la segunda generación de redes BC o *Blockchain 2.0* se añade el concepto de contratos inteligentes, generación inaugurada por la red Ethereum y que supone un gran avance en la TBC (vid. pág. 2, 8, 10)

**Blockchain 3.0** En la tercera generación de redes BC o *Blockchain 3.0* se incluyen todas las redes BC en las que se despliegan aplicaciones descentralizadas o *dApps*, que hacen uso de contratos inteligentes, además intentan mejorar los puntos débiles de sus precedesoras, como puede ser, la escalabilidad. Un ejemplo de este tipo de redes es EOS (vid. pág. 8, 10, 64)

**bloque** Conjunto de información almacenado en cada uno de los elementos de la BC. Para más detalles véase la sección 2.2.2 (vid. pág. 13, 18, 19, 122)

**bloque anterior** Sinónimo de bloque padre (vid. pág. 17, 19, 20)

**bloque padre** En una *blockchain* cada bloque (salvo el inicial) tiene un bloque padre, que es el bloque previo referenciado a través del *hash del bloque previo* (vid. pág. 20, 115, 117)

**clave privada** En la criptografía de clave asimétrica la clave privada es una de las dos claves generadas, la que no se comunica. La clave privada debe mantenerse en secreto, ya que es la que proporciona la seguridad del cifrado asimétrico. En bitcoin se utiliza para generar la clave pública aplicando la multiplicación de curva elíptica (vid. pág. 14–17, 35, 86)

**clave pública** En la criptografía de clave asimétrica la clave pública es una de las dos claves generadas, la que se comunica. Siempre está matematicamente relacionada con la clave privada, ya que se genera a partir de esta. A partir de la clave pública, aplicando determinadas directivas criptográficas, se obtiene la dirección pública a través de la cual puede enviar o recibir *tokens* (vid. pág. 12, 14, 15, 17, 25, 48, 69, 86, 115, 116)

***coin mixer*** Es un servicio o herramienta que permite a los poseedores de criptomonedas obtener mayor privacidad, evitar la trazabilidad de las monedas y mantener su anonimato. El usuario envía cierta cantidad al coin mixer que se encargará de devolverselas al usuario, normalmente en varias transacciones y desde diferentes cuentas (vid. pág. 58)

***coin-mixers scheme*** Engloba cualquier método que implique la ofuscación de fondos monetarios sustituyendo unos por otros, eliminando así cualquier posibilidad de trazar los fondos originales (vid. pág. 61)

**contrato inteligente** Es un programa informático que se almacena y ejecuta en una red BC y que permite la ejecución de acuerdos entre diferentes partes interesadas de forma automática sin necesidad de un tercero de confianza. Para más detalles véase la sección 2.2.4 (vid. pág. 3, 8, 10, 26, 27, 33, 39, 42, 43, 45, 46, 56, 57, 60, 61, 63, 64, 66, 85, 87, 91, 95, 100, 115, 117, 118, 121, 122)

**Corda** Es un *framework* que da lugar a la creación de redes BC privadas, principalmente orientadas a la banca. Una de las características principales de este *framework* es que las transacciones no se difunden a todos los nodos, únicamente tienen acceso a ellas las partes interesadas (vid. pág. 30, 94)

**criptografía homomórfica** Es una técnica criptográfica que consiste en realizar operaciones básicas con datos cifrados sin la necesidad de descifrarlos. Se diferencian dos tipos, si el algoritmo permite realizar o bien la suma o bien el producto se denomina **parcialmente homomórfico**, y si permite realizar las dos operaciones se denomica **totalmente homomórfico** (vid. pág. 55, 58, 86, 87, 103, 104, 108)

***dApp*** "decentralized Application" (Aplicación descentralizada). Término utilizado para referirse a todas las aplicaciones que se ejecutan de manera descentralizada en una red P2P y que comunmente también tienen una interfaz de usuario. Al hacer uso de contratos inteligentes no requieren de una autoridad central que controle la *dApp*. En Ethereum se definen 3 tipos de *dApps*: administración de fondos, integración de dinero con eventos externos del mundo real y DAOs (vid. pág. 8, 10, 33, 65, 115, 117)

***Delegated Proof of Stake*** Este algoritmo de consenso está considerado como una versión mejorada del algoritmo de consenso *Proof of Stake*. Son algoritmos muy similares, pero la principal diferencia es que en éste se utiliza un sistema de votación democrática para elegir a los nodos que pueden añadir nuevos bloques a la cadena. Al realizarse bajo votación de todos los participantes de la red, los nodos productores de bloques están interesados en ser honestos, eficientes y aportar valor a la red (en forma de proyectos asociados, ideas de mejora, etc.). Para más detalles véase la sección 2.2.3 (vid. pág. 31, 111, 117)

**dirección** En redes como Bitcoin, la dirección es un conjunto de 26 a 35 carácteres alfanuméricos generado a partir de la codificación en Base58 del *hash* de la clave pública del usuario al que pertenece. Las direcciones tienen asociadas las transacciones enviadas y recibidas a través de las cuales se calcula el saldo (vid. pág. 12, 15)

**doble gasto** Engloba a cualquier tipo de ataque que afectea a una red BC y que mediante cualquier técnica o mecanismo consiga utilizar el mismo *token* o criptomoneda dos veces, conllevando a un doble gasto de la misma "moneda". Ver sección 2.4.4.1 (vid. pág. 34, 35, 67)

**EOS** Es una red BC surgida a partir de una ICO diseñada con el fin de permitir una escalabilidad tanto vertical como horizontal para las *dApps*. Utilizan el algoritmo de consenso *Delegated Proof of Stake* (dPoS) (vid. pág. 1, 8, 28, 100, 102, 115)

**ERC-20** Es un estándar sobre las funciones y eventos que debe contener un contrato inteligente para permitir ejecutar transacciones de envío del propio *token* de forma fiable y segura (vid. pág. 39, 88)

**Ether** Criptomoneda asociada a la red Ethereum (vid. pág. 32, 114)

**Ethereum** Red BC pública que utiliza el algoritmo de consenso *Proof of Stake* (PoS) y fue la primera red en permitir la ejecución de contratos inteligentes (vid. pág. 1, 2, 8, 22, 27, 28, 32, 33, 39, 46, 86, 88, 94, 95, 99, 102, 111, 114–118, 121)

***exchange*** Son plataformas de intercambio que permiten convertir criptomonedas a monedas por decreto o *fiat*, y viceversa (vid. pág. 35, 36, 38, 65)

**Filecoin** Es una red BC orientada al almacenamiento de datos de forma cooperativa y proporcionando un método de recuperación de los mismos. (vid. pág. 93)

**firma digital** Se denomina firma digital al conjunto de datos asociados a un mensaje que permite asegurar la integridad del mensaje y la identidad del firmante. En términos generales, una firma digital debe ser no repudiable por el firmante, reconocible de manera sencilla tanto por el firmante, como por terceros y su generación debe ser fácil y barata (vid. pág. 10, 15, 56, 59, 87, 91, 97)

***fork*** El término *fork* tiene dos acepciones:

1. Bifurcación en la cadena de bloques que ocurre cuando dos o más bloques se añaden de manera casi simultánea de modo que acaban teniendo el mismo bloque padre. Esto deriva en dos potenciales cadenas de bloques "válidas" que en algún momento una de ellas será desechada y la otra continuará como la cadena principal
2. Creación de proyectos BC partiendo de un proyecto BC anterior. Este es el sentido en que se usa generalmente *fork* en proyectos de *software*. Se puede considerar una especie de actualización o mejora de la red BC, a la vez que se mantiene la red original, ya sea porque no todos los usuarios o desarrolladores están de acuerdo con el cambio, o ya sea por mantener la antigua red por motivos históricos o de retrocompatibilidad.

   Dentro de esta segunda acepción se puede distinguir entre *soft-fork* y *hard-fork*

(vid. pág. 20, 24, 35, 39, 55, 88, 118, 121)

***framework*** o marco de trabajo o referencia, se podría definir como una estructura que sirve de guía para la construcción de un sistema o plataforma y que permite

un desarrollo más rápido y con mayor calidad (vid. pág. 30, 44, 85, 87, 89, 99, 102, 116, 118, 119)

***hard-fork*** Es una modificación del protocolo de una red BC (véase la segunda acepción de *fork*), de tal naturaleza que requiere una actualización obligatoria de los nodos para continuar en la red. Los nodos que no se actualicen, estarían trabajando sobre otra cadena de bloques previa. Este caso ha ocurrido en la red Ethereum, derivando en dos cadenas (redes) independientes: Ethereum y Ethereum Classic (vid. pág. 32, 57, 114, 117)

***hash*** Es el resultado de la ejecución de una función *hash*. Este resultado está formado por una serie de bytes de longitud fija para cada algoritmo de *hash*, y que no depende de la longitud de la entrada. En criptografía las funciones deben cumplir una serie de propiedades adicionales para imposibilitar la creación de diferentes entradas que puedan dar lugar al mismo *hash*. Para más detalles véase la sección 2.2.1.1 (vid. pág. 6–8, 10–15, 17–19, 22, 55, 59, 69, 89, 92, 96, 112, 114, 116, 119, 120)

***hijack*** Engloba a cualquier tipo de ataque que tenga como resultado el robo o secuestro temporal de cualquier tipo de sistema informático (p. ej. conexiones de red, modificar redirecciones de un servidor DNS, sesiones de un navegador, etc.) (vid. pág. 34, 36)

***Hyperledger Fabric*** Es un *framework* orientado a la creación de redes BC privadas. Permite el despliegue de contrato inteligente a los que denominan *Chaincodes*. Una de sus características es que permite una gran personalización, tanto en términos de consenso, como permitiendo elegir el lenguaje de programación para desarollar los *Chaincodes* (vid. pág. 30, 85, 87, 94, 111)

**ICO** Una *Initial Coin Offering*, u oferta inicial de moneda (ICO) se define como la financiación a partir de la emisión de una cantidad de *tokens* o criptomonedas propias de la red BC que se encuentra en fase de desarrollo o únicamente es un prototipo. Hoy en día este modelo es muy críticado debido a las numerosas estafas que se han cometido utilizando este tipo de financiación (vid. pág. 117)

**LevelDB** Es un sistema de base de datos de tipo clave/valor construido por Google. Se considera un sistema de base de datos NoSQL, sin soporte para índices y con los datos ordenados por clave. Las aplicaciones lo utilizan como una biblioteca, ya que no posee interfaz web ni permite interactuar con él mediante línea de comandos. Algunos de los sistemas que utilizan LevelDB son, por ejemplo, Bitcoin Core, AutoCAD 2016 o *Minecraft: Pocket Edition* (vid. pág. 93)

***LoRa*** Tecnología de modulación de las redes *LoRaWAN* que permite intercambiar pequeñas cantidades de datos a baja velocidad con un largo alcance y bajo consumo de energía, utilizada en dispositivos con limitaciones energéticas, típicamente en IoT (vid. pág. 61)

***memcached*** Sistema distribuido de código abierto de tipo clave-valor para almacenar objetos en memoria, utilizado como caché a nivel de aplicación en numerosos sistemas tradicionales. Utiliza *Distributed Hash Tables* (DHTs) (vid. pág. 96, 104)

**meta-regla** Se define como una regla que indica qué hacer y cómo hacerlo. Se podría decir que es una regla para construir otras reglas (vid. pág. 100)

**minar** Se puede definir el término de "minar" en el ámbito de la TBC como el conjunto de procesos llevados a cabo para validar y procesar transacciones/bloques en una red BC. Otro término relacionado es de *minería* que se define como la validación y el registro de las transacciones o bloques en una red BC. Por último, se llaman *mineros* en la red Bitcoin a los nodos que poseen un *hardware* altamente especializado en la *minería* y con altas capacidades de cómputo, y que participan en la red intentando resolver la prueba de trabajo para añadir nuevos bloques a la cadena. En redes que no utilizan PoW el concepto es similar, pero varía la forma en la que añaden los bloques y muchas veces, el término usado para referirse a estos nodos (vid. pág. 6, 7, 17, 22–25, 28, 31, 35–40, 48, 57, 58, 94, 98, 101)

***mixing*** Técnica mediante la cual se intenta anonimizar la procedencia de fondos, mediante el movimiento de dichos fondos entre distintos usuarios (vid. pág. 57)

**Multichain** Es un *framework* para la creación de redes públicas o privadas y que permite un alto grado de personalización. Proporcionan una API y una interfaz de línea de comandos para poder operar con la red desplegada. En concreto, este *framework* está centrado en la creación (transparente para el usuario) y despliegue de redes BC privadas para uso bancario (vid. pág. 30)

**Nano** Es un tipo de red que utiliza DAG, en lugar de Blockchain (BC). Sus principales características son una baja latencia, alto rendimiento y utiliza una arquitectura propia denominada *block-lattice*. Como algoritmo de consenso utilizan *Proof of Stake* (PoS) (vid. pág. 8, 48, 91)

***nonce*** Número que solo puede ser utilizado una vez (abreviatura de *number that can be used only once*) es un número aleatorio utilizado en combinación con las funciones *hash* para evitar la manipulación de la información de los bloques en una red BC (vid. pág. 7, 17, 19, 22)

***off-chain*** Se denomina *off-chain* a todo proceso que ocurre fuera de la red BC (vid. pág. 10, 34, 35, 87, 91, 93, 101, 102, 104)

***on-chain*** Se denomina *on-chain* a todo proceso que ocurre dentro de la red BC (vid. pág. 10, 87, 93, 101, 102, 104)

***permissioned*** Es un tipo de redes BC en la cual es necesario tener permiso para formar parte de ellas. La idea detrás de este tipo de redes es aprovechar todas las ventajas que ofrece la TBC pero sin desperdiciar el control que proporciona una

autoridad central que controle las acciones de los usuarios y el funcionamiento de la red (vid. pág. 28, 120)

***permissionless*** Es un tipo de red BC en la cual cualquier usuario puede formar parte de la red y de las decisiones de consenso, siempre que cumpla las condiciones propias de la red. De forma general, este tipo de redes suelen ser muy descentralizadas, transparentes e inmutables (en comparación con las redes *permissioned*), pero también tienen una menor escalabilidad, mayor tiempo entre transacciones y no suelen ser energéticamente eficientes (p. ej. Bitcoin) (vid. pág. 28, 69)

***Proof of Authority*** Es un algoritmo de consenso cuya principal característica es que para ser un nodo validador (y poder añadir nuevos bloques a la cadena) se debe proporcionar la identidad real. Cada nodo validador tendrá una determinada reputación. De entre todos los nodos validadores (número limitado por la red), se elige aleatoriamente uno en cada ronda, que se encargará de firmar los bloques (vid. pág. 34, 112)

***Proof of QoS*** Es un algoritmo de consenso cuyo funcionamiento básico consiste en la división de la red en diferentes regiones. Cada región designará un nodo como validador en función de su calidad de servicio (QoS). El conjunto de nodos elegidos participarán en un algoritmo de consenso de tipo BFT (vid. pág. 97, 113)

***Proof of Stake*** Es uno de los algoritmos de consenso más populares junto con *Proof of Work* y nace con el objetivo de solucionar algunos problemas conocidos asociados a PoW. En este algoritmo no hace falta una gran capacidad de computación ni un consumo eléctrico desmesurado para añadir un nuevo bloque a la cadena. En su lugar se definen una serie de criterios como pueden ser cantidad de *tokens*, o el tiempo que lleva el usuario con dichos *tokens*, y luego de forma aleatoria se elegirá a los nodos que cumplan dichos criterios y que puedrán añadir nuevos bloques. Para más detalles véase la sección 2.2.3 (vid. pág. 23, 31, 34, 113, 116, 117, 119)

***Proof of Work*** Es un protocolo de consenso utilizado por primera vez en la red Bitcoin cuya principal característica es requerir una prueba de trabajo computacional a los nodos que desean añadir nuevos bloques a la cadena. Dichos bloques luego deberán ser verificados por el resto de nodos mineros de la red, asegurándose así que nodos maliciosos que quieran añadir bloques de forma incontrlada, deban consumir gran cantidad de recursos para ello. Para más detalles véase la sección 2.2.3 (vid. pág. 8, 22, 31, 48, 113, 114, 120)

***puzzle friendly*** Es una propiedad de las funciones *hash* utilizadas en BC que es necesaria para poder implementar el algoritmo de *Proof of Work*. Para la definición rigurosa de esta propiedad véase la sección 2.2.1.1 (vid. pág. 12, 22)

**reglas secundarias** Son *"meta-reglas"* por las cuales todas las demás reglas del sistema son identificadas y entendidas como válidas, es decir, las que se consideran como leyes válidas dentro de ese sistema (vid. pág. 100)

***routing-awareness*** En español, "conciencia (o conocimiento) del enrutado", se refiere a cualquier técnica o mecanismo por el cual un nodo tiene conocimiento de las diferentes rutas que propone determinado algoritmo (p. ej. BGP) en algunas situaciones y tiene la capacidad de detectar cuando el servidor de enrutado opera de forma anómala. En ese caso, utilizará las rutas que conoce (vid. pág. 36)

**Science Direct** Base de datos bibliográfica de artículos de revistas científicas. URL: `https://www.sciencedirect.com/` (vid. pág. 73)

**Scopus** Base de datos de referencias bibliográficas y citas de la empresa Elsevier, de literatura científica y revisada por pares. URL: `https://www.scopus.com/search/form.uri?display=basic` (vid. pág. 73)

***selfish mining*** Es una práctica realizada en redes BC, principalmente en Bitcoin, en la cual un grupo de mineros retienen los últimos bloques válidos creando una cadena paralela que será añadida a la cadena principal, revertiendo todos los bloques que hayan añadido otros mineros a la cadena principal. Esta práctica pone en jaque la descentralización de la red (vid. pág. 57)

***sharding*** Este término proviene de las bases de datos tradicionales. En BC la idea es similar, se dividen los datos de las transacciones en porciones que llaman *"shards"*, las cuales procesan diferentes nodos de forma simultánea (vid. pág. 58, 68, 93, 95, 97, 98, 104, 105, 108)

***smart contract*** Véase contrato inteligente (vid. pág. 26)

***soft-fork*** Es una modificación del protocolo de una red BC (véase la segunda acepción de *fork*), que mantiene la compatibilidad con versiones anteriores, lo que hace que la actualización de los nodos sea opcional. Normalmente consiste en alguna modificación de las reglas de consenso (vid. pág. 117)

**solidity** Es un lenguaje de programación de alto nivel orientado al desarollo de contratos inteligentes que serán ejecutados de forma óptima en la *Ethereum Virtual Machine*,o máquina virtual de Ethereum (EVM), y por tanto, en la red Ethereum. Es un lenguaje *Turing-completo* y su sintaxis es muy similar a la de *JavaScript*. Como curiosidad, *Solidity* actualmente soporta herencia y herencia múltiple (vid. pág. 27)

**Springer** Editorial que publica libros y publicaciones científicas revisadas por pares de campos como la ciencia, la tecnología y la medicina. URL: `https://www.springer.com/gp` (vid. pág. 73)

***stake*** Es el bloqueo temporal de cierta cantidad de monedas o *tokens*, que se vuelven inutilizables hasta que se haga "*unstake*" de las mismas, y que simbolizan el compromiso con la red BC del nodo que lo ha hecho. Este bloqueo tiene asociadas ciertas ventajas como la posibilidad de utilizar recursos de la red, o poder votar

a un productor de bloques (las ventajas variarán en función de la red BC) (vid. pág. 23, 24, 35)

***swarming*** La técnica de *swarming* o "enjambre" reduce la latencia y aumenta el rendimiento ya que los datos se recuperan en paralelo desde los nodos más cercanos y convenientes. Un "enjambre" consta de numerosos nodos distribuidos geográficamente y, por lo tanto, acceder a ellos a través de una red descentralizada conduce a una mayor confiabilidad y escalabilidad (vid. pág. 93, 104, 105, 108)

***token*** En el ámbito de la TBC, un token es un bien similar a las criptomonedas pero que es originada dentro de una red BC y suelen representar propiedad, productos o similares. Existen estándares que regulan este tipo de bienes como son el ERC-20 o el ERC-1155, entre otros (vid. pág. 6, 23–25, 31, 34, 35, 39, 47, 63, 67, 88, 90, 100–102, 115–118, 120, 121)

**transacción** Unidad mínima de información almacenada en la BC. Generalmente varias transacciones se agrupan en un bloque (vid. pág. 13, 18)

***trusted relay*** Componente de confianza, comúnmente un contrato inteligente, que permite la comunicación entre diferentes redes *blockchain* (vid. pág. 65)

***wallet*** Se denomina "wallet" o billetera a todo *hardware* o *software* que permite la gestión y el almacenamiento de la pareja de claves público/privada de cualquier red BC y que puede iniciar transacciones y realizar otras operaciones (vid. pág. 6, 16, 17, 38)

***whale*** Se denomina *whales* o *ballenas* a los nodos o grupos de nodos que trabajan de forma conjunta y que poseen una gran cantidad de la criptomoneda propia de la red. En redes como Bitcoin existe cierta preocupación por este tipo de usuario de la red ya que podría derivar en el control de la red por parte de ese número reducido de individuos (vid. pág. 24, 25)

***white paper*** En el ámbito de la TBC se define como un documento técnico publicado antes de el lanzamiento de una nueva moneda digital o un proyecto basado en *blockchain*. Incluye características del producto o servicio incluyendo aspectos comerciales, financieros, o cualquier tipo de información que pueda resultar relevante para un futuro usuario o inversor (vid. pág. 7, 8, 51)

# Bibliografía

La bibliografía está estructurada en secciones. En la primera se muestran todas las referencias correspondientes a la **literatura blanca** examinada y comentada con detalle en la sección 3.4, más las referencias a los estados del arte previos comentados en la sección 3.1. La segunda contiene todas las referencias correspondientes a la **literatura gris** que se comentan también en la sección 3.4. La tercera y cuarta recogen el resto de referencias que se han consultado para completar el conocimiento del autor, de cara a redactar alguno de los apartados de la memoria, en especial la introducción del capítulo 1 y la presentación general de la tecnología Blockchain del capítulo 2. Este conjunto de fuentes se ha separado a su vez en dos: documentos que podrían considerarse literatura "blanca" (libros, artículos, etc.) y resto de materiales en linea (manuales de *software*, foros de discusión, páginas web comerciales, etc.) dando lugar a las secciones tercera (**Otras fuentes citadas**) y cuarta (**Otras fuentes en línea consultadas**), respectivamente, de esta bibliografía. Cada entrada bibliográfica incluye al final de la misma y entre paréntesis el listado de páginas en que es citada en esta memoria.

## Literatura blanca


[Ali+18] Saqib Ali, Guojun Wang, Bebo White y Roger Leslie Cottrell. "A blockchain-based decentralized data storage and access framework for pinger". En: *2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*. IEEE. 2018, págs. 1303-1308. DOI: 10.1109/TrustCom/BigDataSE.2018.00179 (vid. págs. 82, 89).

[Alr+20] S. M. Alrubei, E. A. Ball, J. M. Rigelsford y C. A. Willis. "Latency and Performance Analyses of Real-World Wireless IoT-Blockchain Application". En: *IEEE Sensors Journal* 20.13 (2020), págs. 7372-7383 (vid. págs. 82, 94).

[Aro+19] Dev Arora, Siddharth Gautham, Harshit Gupta y Bharat Bhushan. "Blockchain-based Security Solutions to Preserve Data Privacy and Integrity". En: *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*. IEEE. 2019, págs. 468-472 (vid. págs. 81, 85).

[Ben19] Gayan Benedict. "Challenges of DLT-enabled ScalableGovernance and the Role of Standards". En: *Journal of ICT Standardization* (2019) (vid. págs. 83, 100).

[BTH19] Bert-Jan Butijn, Damian A. Tamburri y Willem-Jan Van Den Heuvel. "Blockchains: a Systematic Multivocal Literature Review". En: *arXiv* (2019). arXiv: 1911.11770 [cs.CR] (vid. págs. 2, 53, 61, 71).

[BTH20] Bert-Jan Butijn, Damian A. Tamburri y Willem-Jan van den Heuvel. "Blockchains: A Systematic Multivocal Literature Review". En: *ACM Comput. Surv.* 53.3 (jun. de 2020). ISSN: 0360-0300. DOI: 10.1145/3369052 (vid. pág. 61).

[CCY19] Ting Cai, Wuhui Chen y Yang Yu. "BCSolid: A Blockchain-Based Decentralized Data Storage and Authentication Scheme for Solid". En: *Communications in Computer and Information Science Blockchain and Trustworthy Systems* (2019), págs. 676-689. DOI: 10.1007/978-981-15-2777-7_55 (vid. pág. 91).

[Cre19] Marco Crepaldi. "Why blockchains need the law: Secondary rules as the missing piece of blockchain governance". En: *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*. 2019, págs. 189-193 (vid. págs. 83, 100).

[DN17] Hoang Giang Do y Wee Keong Ng. "Blockchain-based system for secure data storage with private keyword search". En: *2017 IEEE World Congress on Services (SERVICES)*. IEEE. 2017, págs. 90-93 (vid. págs. 82, 89).

[Dor+19] Ali Dorri, Salil S. Kanhere, Raja Jurdak y Praveen Gauravaram. "LSB: A Lightweight Scalable Blockchain for IoT security and anonymity". En: *Journal of Parallel and Distributed Computing* 134 (2019), págs. 180-197. ISSN: 0743-7315. DOI: 10.1016/j.jpdc.2019.08.005 (vid. págs. 82, 94).

[Fan+19] Kai Fan *et al.* "A blockchain-based clock synchronization Scheme in IoT". En: *Future Generation Computer Systems* 101 (2019), págs. 524-533. ISSN: 0167-739X. DOI: 10.1016/j.future.2019.06.007 (vid. págs. 82, 94).

[Fan+20] Yuqi Fan *et al.* "A blockchain-based data storage framework: A rotating multiple random masters and error-correcting approach". En: *Peer-to-Peer Networking and Applications* (2020). DOI: 10.1007/s12083-020-00895-5 (vid. págs. 82, 92).

[GM18] Shlok Gilda y Maanav Mehrotra. "Blockchain for Student Data Privacy and Consent". En: *2018 International Conference on Computer Communication and Informatics (ICCCI)*. IEEE. 2018, págs. 1-5 (vid. págs. 81, 85).

[Gud+20] Lewis Gudgeon, Pedro Moreno-Sanchez, Stefanie Roos, Patrick McCorry y Arthur Gervais. "SoK: Layer-Two Blockchain Protocols". En: *Financial Cryptography and Data Security*. Ed. por Joseph Bonneau y Nadia Heninger. Cham: Springer International Publishing, 2020, págs. 201-226. ISBN: 978-3-030-51280-4 (vid. pág. 98).

[Han+20] Runchao Han, Gary Shapiro, Vincent Gramoli y Xiwei Xu. "On the performance of distributed ledgers for Internet of Things". En: *Internet of Things* 10 (2020). Special Issue of the Elsevier IoT Journal on Blockchain Applications in IoT Environments, pág. 100087. ISSN: 2542-6605. DOI: 10.1016/j.iot.2019.100087 (vid. págs. 82, 94).

[Har20] Harmony. *Harmony – Open Consensus for 10B*. 2020. URL: https://www.harmony.one/ (en línea, visitado el 20-06-2020) (vid. pág. 95).

[HMB20] Seyed Hosseini Bamakan, Amirhossein Motavali y Alireza Babaei Bondarti. "A survey of blockchain consensus algorithms performance evaluation criteria". En: *Expert Systems with Applications* 154 (2020), pág. 113385. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2020.113385 (vid. págs. 82, 98).

[Hüt19] Moritz Hütten. "The soft spot of hard code: blockchain technology, network governance and pitfalls of technological utopianism". En: *Global Networks* 19.3 (2019), págs. 329-348 (vid. págs. 83, 99).

[JP18] Thomas John y Mantri Pam. "Complex adaptive blockchain governance". En: *MATEC Web of Conferences*. Vol. 223. EDP Sciences. 2018, pág. 01010 (vid. págs. 83, 100).

[Kim+19] Hyojin Kim, Jong-Hyeok Park, Steve Hyuntae Jung y Sang-Won Lee. "Optimizing RocksDB for Better Read Throughput in Blockchain Systems". En: *2019 23rd International Computer Science and Engineering Conference (ICSEC)*. IEEE. 2019, págs. 305-309 (vid. págs. 82, 96).

[Kim19] Song-Kyoo Kim. "Blockchain Governance Game". En: *Computers & Industrial Engineering* 136 (2019), págs. 373-380. ISSN: 0360-8352. DOI: 10.1016/j.cie.2019.07.004 (vid. págs. 83, 99).

[Lim19] Hock Chuan Lim. "Self and regulated governance simulation: Exploring governance for blockchain technology". En: (2019) (vid. págs. 83, 99).

[Lou+18] Faiza Loukil, Chirine Ghedira-Guegan, Khouloud Boukadi y Aicha Nabila Benharkat. "Towards an End-to-End IoT Data Privacy-Preserving Framework Using Blockchain Technology". En: *Web Information Systems Engineering – WISE 2018*. Ed. por Hakim Hacid, Wojciech Cellary, Hua Wang, Hye-Young Paik y Rui Zhou. Cham: Springer International Publishing, 2018, págs. 68-78. ISBN: 978-3-030-02922-7 (vid. págs. 81, 85).

[LW05] Joseph K. Liu y Duncan S. Wong. "Linkable Ring Signatures: Security Models and New Schemes". En: *Proceedings of the 2005 International Conference on Computational Science and Its Applications - Volume Part II*. ICCSA'05. Singapore: Springer-Verlag, 2005, págs. 614-623. ISBN: 3540258612. DOI: 10.1007/11424826_65 (vid. pág. 97).

[LWL18] Shaowei Liu, Jing Wu y Chengnian Long. "Iot meets blockchain: parallel distributed architecture for data storage and sharing". En: *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE. 2018, págs. 1355-1360. DOI: 10.1109/Cybermatics_2018.2018.00233 (vid. págs. 82, 89, 90).

[Mar19] Selvarani Mariappan. "Blockchain Technology: Disrupting the Current Business and Governance Model". En: *International Journal of Recent Technology and Engineering (IJRTE)* (2019) (vid. págs. 83, 99).

[Mer+20] Paul Merrill, Thomas H. Austin, Justin Rietz y Jon Pearce. "Ping-Pong Governance: Token Locking for Enabling Blockchain Self-governance". En: *Mathematical Research for Blockchain Economy* (2020), págs. 13-29. DOI: 10.1007/978-3-030-37110-4_2 (vid. págs. 83, 101).

[Oha19] Mamoru Ohara. "A Study on Checkpointing for Distributed Applications Using Blockchain-Based Data Storage". En: *2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC)*. IEEE. 2019, págs. 116-1161 (vid. pág. 90).

[Pai99] Pascal Paillier. "Public-key cryptosystems based on composite degree residuosity classes". En: *International conference on the theory and applications of cryptographic techniques*. Springer. 1999, págs. 223-238 (vid. pág. 58).

[Pel+20] Rowan van Pelt, Slinger Jansen, Djuri Baars y Sietse Overbeek. "Defining Blockchain Governance: A Framework for Analysis and Comparison". En: *Information Systems Management* (2020), págs. 1-21 (vid. págs. 83, 101).

[Pen18] Sun Peng. "Application and Research of Blockchain Technology in P2P Network Distributed Data Storage". En: *International Conference on Applications and Techniques in Cyber Security and Intelligence ATCI 2018*. Ed. por Jemal Abawajy, Kim-Kwang Raymond Choo, Rafiqul Islam, Zheng Xu y Mohammed Atiquzzaman. Springer International Publishing, 2018, págs. 953-960 (vid. págs. 82, 88).

[Qia+19] Zhi Qiao, Congcong Zhu, Zhiwei Wang y Nianhua Yang. "Anonymous IoT Data Storage and Transaction Protocol Based on Blockchain and Edge Computing". En: *Science of Cyber Security Lecture Notes in Computer Science* (2019), págs. 181-189. DOI: 10.1007/978-3-030-34637-9_13 (vid. págs. 82, 91).

[Roy+19] Koushik Roy, Nur Islam, Tarango Khan y Mohammad Monirujjaman Khan. "A Novel Approach to Data Storage Using Blockchain Technology". En: *2019 International Conference on Information Technology (ICIT)*. IEEE. 2019, págs. 245-250 (vid. págs. 82, 91).

[SC18] Abdurrashid Ibrahim Sanka y Ray CC Cheung. "Efficient high performance FPGA based NoSQL caching system for blockchain scalability and throughput improvement". En: *2018 26th International Conference on Systems Engineering (ICSEng)*. IEEE. 2018, págs. 1-8 (vid. págs. 82, 96).

[Sha+19] Besfort Shala, Ulrich Trick, Armin Lehmann, Bogdan Ghita y Stavros Shiaeles. "Novel trust consensus protocol and blockchain-based trust evaluation system for M2M application services". En: *Internet of Things* 7 (2019), pág. 100058. ISSN: 2542-6605. DOI: 10.1016/j.iot.2019.100058 (vid. págs. 82, 90).

[SK19] Rakesh Shrestha y Shiho Kim. "Chapter Ten - Integration of IoT with blockchain and homomorphic encryption: Challenging issues and opportunities". En: *Role of Blockchain Technology in IoT Applications*. Ed. por Shiho Kim, Ganesh Chandra Deka y Peng Zhang. Vol. 115. Advances in Computers. Elsevier, 2019, págs. 293-331. DOI: 10.1016/bs.adcom.2019.06.002 (vid. págs. 81, 86).

[SLS19] Fatemeh Saadatmand, Rikard Lindgren y Ulrike Schultze. "Configurations of platform organizations: Implications for complementor engagement". En: *Research Policy* 48.8 (2019). The Digital Transformation of Innovation and Entrepreneurship, pág. 103770. ISSN: 0048-7333. DOI: 10.1016/j.respol.2019.03.015 (vid. págs. 83, 99).

[Tay+20] Paul J. Taylor, Tooska Dargahi, Ali Dehghantanha, Reza M. Parizi y Kim-Kwang Raymond Choo. "A systematic literature review of blockchain cyber security". En: *Digital Communications and Networks* 6.2 (2020), págs. 147-156. ISSN: 2352-8648. DOI: 10.1016/j.dcan.2019.01.005 (vid. págs. 82, 88, 89).

[Wan+20a] Hao Wang, Shenglan Ma, Hong-Ning Dai, Muhammad Imran y Tongsen Wang. "Blockchain-based data privacy management with Nudge theory in open banking". En: *Future Generation Computer Systems* 110 (2020), págs. 812-823. ISSN: 0167-739X. DOI: 10.1016/j.future.2019.09.010 (vid. págs. 81, 87).

[Wan+20b] Jitao Wang, Guozi Sun, Yu Gu y Kun Liu. "Distributed Electronic Data Storage and Proof System Based on Blockchain". En: *Blockchain Technology and Application*. Ed. por Xueming Si *et al.* Singapore: Springer Singapore, 2020, págs. 48-67 (vid. págs. 82, 91).

[WGC19] Hao Wang, Chaonian Guo y Shuhan Cheng. "LoC — A new financial loan management system based on smart contracts". En: *Future Generation Computer Systems* 100 (2019), págs. 648-655. ISSN: 0167-739X. DOI: 10.1016/j.future.2019.05.040 (vid. págs. 81, 86).

[Xio+19] Zhentian Xiong, Zoe L. Jiang, Shuqiang Yang, Xuan Wang y Junbin Fang. "SSHTDNS: A Secure, Scalable and High-Throughput Domain Name System via Blockchain Technique". En: *Network and System Security* (2019), págs. 272-287. DOI: 10.1007/978-3-030-36938-5_16 (vid. págs. 82, 96).

[Xu+17] Yuqin Xu *et al.* "E-commerce Blockchain Consensus Mechanism for Supporting High-Throughput and Real-Time Transaction". En: *Collaborate Computing: Networking, Applications and Worksharing* (2017), págs. 490-496. DOI: 10.1007/978-3-319-59288-6_46 (vid. págs. 82, 96).

[Yu+19] Bin Yu, Joseph Liu, Surya Nepal, Jiangshan Yu y Paul Rimba. "Proof-of-QoS: QoS based blockchain consensus protocol". En: *Computers & Security* 87 (2019), pág. 101580. ISSN: 0167-4048. DOI: 10.1016/j.cose.2019.101580 (vid. págs. 82, 97).

[ZJ19] Li Zhihong y Zhang Jie. "Online Knowledge Community Governance Based on Blockchain Token Incentives". En: *Communications in Computer and Information Science* (2019), págs. 64-72. DOI: 10.1007/978-981-15-1209-4_5 (vid. págs. 83, 100).

[ZMM18] D. P. Zegzhda, D. A. Moskvin y A. V. Myasnikov. "Assurance of Cyber Resistance of the Distributed Data Storage Systems Using the Blockchain Technology". En: *Automatic Control and Computer Sciences* 52.8 (2018), págs. 1111-1116. DOI: 10.3103/s0146411618080400 (vid. págs. 82, 88, 89).

## Literatura gris

[BitcoinCore16] Bitcoin Core. *Compact Blocks FAQ*. 2016. URL: https://bitcoincore.org/en/2016/06/07/compact-blocks-faq/ (en línea, visitado el 25-08-2020) (vid. pág. 98).

[Blo19] Blockchain Consultants. *Blockchain Governance Models*. 2019. URL: https://blockchainconsultants.io/blockchain-governance-models (en línea, visitado el 29-08-2020) (vid. pág. 102).

[BlockApps17] *How Blockchain Will Disrupt Data Storage*. 2017. URL: https://blockapps.net/blockchain-disrupt-data-storage/ (en línea, visitado el 02-08-2020) (vid. pág. 93).

[Blocksplain18] Blocksplain. *Blockchain speeds & the scalability debate*. Blocksplain. 2018. URL: https://blocksplain.com/2018/02/28/transaction-speeds/ (en línea, visitado el 25-08-2020) (vid. pág. 98).

[Boc20] S. Bocetta. *Blockchain-based VPNs: The Next Step in Privacy Tech?* 2020. URL: https://securityboulevard.com/2020/04/blockchain-based-vpns-the-next-step-in-privacy-tech/ (en línea, visitado el 29-08-2020) (vid. pág. 88).

[BoogerWooger19] BoogerWooger. *The Key Metrics to Measure Blockchain Network Performance*. Hackernoon. 2019. URL: https://hackernoon.com/how-to-measure-blockchain-network-performance-key-metrics-en1234u4 (en línea, visitado el 25-08-2020) (vid. pág. 98).

[Cat18] Michael Catt. *Blockchain Fundamentals: Latency & Capacity — Featuring the Ark Ecosystem*. 2018. URL: https://medium.com/ku-blockchain-institute/blockchain-fundamentals-featuring-the-ark-ecosystem-part-1-af1f9052e579 (en línea, visitado el 29-08-2020) (vid. pág. 95).

[CIOApps19] CIO Applications. *What makes blockchain an answer to data storage*. 2019. URL: https://www.cioapplications.com/news/what-makes-blockchain-an-answer-to-data-storage-nid-3236.html (en línea, visitado el 14-08-2020) (vid. pág. 93).

[Cli19] Jordan Clifford. *Bandwidth and the Blockchain*. 2019. URL: https://medium.com/scalar-capital/bandwidth-and-the-blockchain-2ad35c57dbdf (en línea, visitado el 25-08-2020) (vid. pág. 98).

[CLP18] C. Compert, M. Luinetti y B. Portier. *Blockchain and GDPR: How blockchain could address five areas associated with GDPR compliance*. 2018. URL: https://iapp.org/media/pdf/resource_center/blockchain_and_gdpr.pdf (en línea, visitado el 27-08-2020) (vid. pág. 87).

[Cza17] J. Czarnecki. *Blockchains and Personal Data Protection Regulations Explained*. 2017. URL: https://www.coindesk.com/blockchains-personal-data-protection-regulations-explained (en línea, visitado el 27-08-2020) (vid. pág. 87).

[Eri19] N. Eriksson. *In Depth: Where is Blockchain Data Stored?* 2019. URL: https://coinnounce.com/in-depth-where-blockchain-data-is-stored/ (en línea, visitado el 08-08-2020) (vid. pág. 93).

[Fer19] A. Fernandez. *Can distributed apps store sensitive data being GDPR compliant?* 2019. URL: https://datascience.aero/apps-store-sensitive-data-gdpr-compliant/ (en línea, visitado el 25-08-2020) (vid. pág. 87).

[Fin18] M. Finck. *Blockchains and Data Protection in the European Union*. 2018. URL: https://edpl.lexxion.eu/data/article/12327/pdf/edpl_2018_01-007.pdf (en línea, visitado el 27-08-2020) (vid. pág. 87).

[Fla+19] A. Flanagan, F. Maclean, M. Sun, N Hewett y R. Liao. *Inclusive Deployment of Blockchain for Supply Chains: Part 4 – Protecting Your Data*. 2019. URL: http://www3.weforum.org/docs/WEF_Inclusive_Deployment_of_Blockchain_for_Supply_Chains_Part_4_Report.pdf (en línea, visitado el 05-08-2020) (vid. pág. 87).

[FVL18] S. Forestier, D. Vodenicarevic y A. Laversanne-Finot. *Blockclique: scaling blockchains through transaction sharding in a multithreaded block graph*. 2018. URL: https://arxiv.org/abs/1803.09029 (en línea, visitado el 29-08-2020) (vid. pág. 95).

[Gur19] M. Gurevin. *Blockchain, Data Privacy, and ZKP: Does blockchain ensure data privacy?* 2019. URL: https://medium.com/octabase/blockchain-data-privacy-and-zkp-769e390d0f64 (en línea, visitado el 27-08-2020) (vid. pág. 88).

[IOS18] Luis-Daniel Ibáñez, Kieron O'Hara y Elena Simperl. "On Blockchains and the General Data Protection Regulation". En: (jul. de 2018) (vid. pág. 87).

[Ken19] H. Kenyon. *Privacy "poisoning" poses threat to companies using blockchain*. 2019. URL: https://phys.org/news/2019-04-privacy-poisoning-poses-threat-companies.html (en línea, visitado el 29-08-2020) (vid. pág. 88).

[Kum19] T. Kumar. *Role Of Blockchain In Data Storage*. 2019. URL: https://www.blockchain-council.org/blockchain/role-of-blockchain-in-data-storage (en línea, visitado el 15-08-2020) (vid. pág. 93).

[Lei18] P. J. Leimgruber. *Introduction to Blockchain Governance*. 2018. URL: https://blog.district0x.io/introduction-to-blockchain-governance-bc6eea42ada3 (en línea, visitado el 29-08-2020) (vid. pág. 102).

[Lim18] C. Lima. *Blockchain-GDPR Privacy by Design: How Decentralized Blockchain Internet will Comply with GDPR Data Privacy*. 2018. URL: https://blockchain.ieee.org/images/files/pdf/blockchain-gdpr-privacy-by-design.pdf (en línea, visitado el 25-08-2020) (vid. pág. 87).

[Mag20] J. Magas. *Blockchain Storage Offers Security, but Leaves Data Transparent*. 2020. URL: https://cointelegraph.com/news/blockchain-storage-offers-security-but-leaves-data-transparent (en línea, visitado el 27-08-2020) (vid. pág. 88).

[Nan18] S. Nanaware. *Blockchain for Data Storage*. 2018. URL: https://medium.com/@sukantkhurana/blockchain-for-data-storage-8c05c00af6fe (en línea, visitado el 15-08-2020) (vid. pág. 93).

[Newswire18] PR Newswire. *The cloud storage market size is expected to grow from USD 30.70 billion in 2017 to USD 88.91 billion by 2022, at a Compound Annual Growth Rate (CAGR) of 23.7 %*. PR Newswire. 2018. URL: https://www.prnewswire.com/news-releases/the-cloud-storage-market-size-is-expected-to-grow-from-usd-3070-billion-in-2017-to-usd-8891-billion-by-2022-at-a-compound-annual-growth-rate-cagr-of-237-300597852.html (en línea, visitado el 15-08-2020) (vid. pág. 93).

[Oma19] G. Omale. *Gartner Predicts for the Future of Privacy 2019*. 2019. URL: https://www.gartner.com/smarterwithgartner/gartner-predicts-2019-for-the-future-of-privacy/ (en línea, visitado el 29-08-2020) (vid. pág. 88).

[Peerchemist16] Peerchemist. *PeerAssets Whitepaper*. 2016. URL: http://peerassets.github.io/WhitePaper/ (en línea, visitado el 29-08-2020) (vid. pág. 102).

[PlayersMoney19] *Settlement Latency Blockchain Technology Review – The Future of Trading?* 2019. URL: https://www.playersmoney.com/settlement-latency-blockchain-technology/ (en línea, visitado el 25-08-2020) (vid. pág. 95).

[Ran+18] Konstantinos Rantos, George Drosatos, Konstantinos Demertzis, Christos Ilioudis y Alexandros Papanikolaou. "Blockchain-based Consents Management for Personal Data Processing in the IoT Ecosystem". En: jul. de 2018, págs. 738-743. DOI: 10.5220/0006911007380743 (vid. pág. 87).

[Rho18] D. Rhodes. *Blockchain Data Storage Could Soon Be The New Standard*. 2018. URL: https://coincentral.com/blockchain-data-storage/ (en línea, visitado el 02-08-2020) (vid. pág. 93).

[Scr18] J. Scribani. *Blockchain Governance: How Boundaries Can Help the Blockchain to Scale*. 2018. URL: https://www.visualcapitalist.com/blockchain-governance-scale/ (en línea, visitado el 29-08-2020) (vid. pág. 103).

[Smi18] W. Smits. *Blockchain Governance: What Is It, What Types Are There and How Does It Work in Practice?* 2018. URL: https://watsonlaw.nl/en/blockchain-governance-what-is-it-what-types-are-there-and-how-does-it-work-in-practice/ (en línea, visitado el 29-08-2020) (vid. pág. 103).

[Str18] L. Strauss. *Governing Distributed Ledgers: Everything You Need to Know*. 2018. URL: https://www.da.ventures/post/governing-distributed-ledgers-everything-you-need-to-know (en línea, visitado el 29-08-2020) (vid. pág. 103).

[Stu20] C. Sturges. *How Can Blockchain Improve Data Storage?* 2020. URL: https://cointelegraph.com/news/how-can-blockchain-improve-data-storage (en línea, visitado el 05-08-2020) (vid. pág. 93).

[Swipe19] Swipe. *DLT for Fast Payments: a Proof of Concept*. Swipe. 2019. URL: https://medium.com/@swipetech/dlt-for-fast-payments-a-proof-of-concept-69be579e3d89 (en línea, visitado el 25-08-2020) (vid. pág. 98).

[Tar20] E. Tarasenko. *How to Use Blockchain to Store Data*. 2020. URL: https://merehead.com/blog/how-to-use-blockchain-to-store-data (en línea, visitado el 08-08-2020) (vid. pág. 93).

[Tru+18] N. Truong, K. Sun, G. Lee e Y. Guo. *GDPR-Compliant Personal Data Management: A Blockchain-based Solution*. 2018. URL: https://arxiv.org/pdf/1904.03038 (en línea, visitado el 25-08-2020) (vid. pág. 87).

[Voi19] S. Voight. *William Mougayar on the Challenges and Reality of Decentralized Blockchain Governance*. 2019. URL: https://blog.blockstack.org/william-mougayar-on-the-challenges-and-reality-of-decentralized-blockchain-governance/ (en línea, visitado el 29-08-2020) (vid. pág. 102).

[Whi18] B. Whittle. *Storing Documents on the Blockchain: Why, How, and Where*. 2018. URL: https://gen-blocks.com/storing-documents-on-the-blockchain/ (en línea, visitado el 02-08-2020) (vid. pág. 93).

[Wil20] Maisie Williams. *What is latency in blockchain?* 2020. URL: https://www.quora.com/What-is-latency-in-blockchain (en línea, visitado el 25-08-2020) (vid. pág. 95).

[WK18] Christian Wirth y Michael Kolain. "Privacy by blockchain design: a blockchain-enabled GDPR-compliant approach for handling personal data". En: *Proceedings of 1st ERCIM Blockchain Workshop 2018*. European Society for Socially Embedded Technologies (EUSSET). 2018 (vid. pág. 87).

[ZNP15] G. Zyskind, O. Nathan y A. Pentland. *Decentralizing Privacy: Using Blockchain to Protect Personal Data*. 2015. URL: https://web.media.mit.edu/~guyzys/data/privacy_slides.pdf (en línea, visitado el 29-08-2020) (vid. pág. 88).

## Otras fuentes citadas

[And+19] Merlinda Andoni *et al.* "Blockchain technology in the energy sector: A systematic review of challenges and opportunities". En: *Renewable and Sustainable Energy Reviews* 100 (2019), págs. 143-174. ISSN: 1364-0321. DOI: 10.1016/j.rser.2018.10.014 (vid. pág. 46).

[Ant17] A. Antonopoulos. *Mastering Bitcoin*. 2nd. O'Reilly Media, Inc., 2017. ISBN: 9781491954386 (vid. pág. 1).

[ANT18] A. Alketbi, Q. Nasir y M. A. Talib. "Blockchain for government services — Use cases, security benefits and challenges". En: *2018 15th Learning and Technology Conference (L T)*. 2018, págs. 112-119 (vid. pág. 43).

[Bac02] Adam Back. "Hashcash - A Denial of Service Counter-Measure". En: *HashCash* (sep. de 2002) (vid. pág. 8).

[BD12] Rob B Briner y David Denyer. "Systematic review and evidence synthesis as a practice and scholarship tool". En: *Handbook of evidence-based management: Companies, classrooms and research* (2012), págs. 112-129 (vid. pág. 59).

[Bel+20] Rafael Belchior, André Vasconcelos, Sérgio Guerreiro y Miguel Correia. "A Survey on Blockchain Interoperability: Past, Present, and Future Trends". En: *arXiv* (2020). arXiv: 2005.14282 (vid. págs. 53, 65).

[Ber+18] Oscar Bermeo-Almeida *et al.* "Blockchain in Agriculture: A Systematic Literature Review". En: *Technologies and Innovation*. Ed. por Rafael Valencia-García, Gema Alcaraz-Mármol, Javier Del Cioppo-Morstadt, Néstor Vera-Lucio y Martha Bucaram-Leverone. Cham: Springer International Publishing, 2018, págs. 44-56. ISBN: 978-3-030-00940-3 (vid. pág. 45).

[BFM88] Manuel Blum, Paul Feldman y Silvio Micali. "Non-Interactive Zero-Knowledge and Its Applications". En: *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*. STOC '88. Chicago, Illinois, USA: Association for Computing Machinery, 1988, págs. 103-112. ISBN: 0897912640. DOI: 10.1145/62212.62222 (vid. pág. 58).

[Bhu+20] B. Bhushan, C. Sahoo, P Sinha y A. Khamparia. "Unification of Blockchain and Internet of Things (BIoT): requirements, working model, challenges and future directions". En: *Wireless Networks AB* (ago. de 2020). DOI: 10.1007/s11276-020-02445-6 (vid. págs. 53, 68).

[BP18] F. Bencic e I. Podnar. "Distributed Ledger Technology: Blockchain Compared to Directed Acyclic Graph". En: *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. 2018, págs. 1569-1570. DOI: 10.1109/ICDCS.2018.00171 (vid. pág. 1).

[But+] Vitalik Buterin *et al.* "A next-generation smart contract and decentralized application platform". En: () (vid. pág. 8).

[BWL18] D. Burkhardt, M. Werling y H. Lasi. "Distributed Ledger". En: *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*. 2018, págs. 1-9 (vid. pág. 47).

[CDP19] Fran Casino, Thomas K. Dasaklis y Constantinos Patsakis. "A systematic literature review of blockchain-based applications: Current status, classification and open issues". En: *Telematics and Informatics* 36 (2019), págs. 55-81. ISSN: 0736-5853. DOI: 10.1016/j.tele.2018.11.006 (vid. págs. 4, 53, 59).

[CL99] Miguel Castro y Barbara Liskov. "Practical Byzantine Fault Tolerance". En: *Proceedings of the Third Symposium on Operating Systems Design and Implementation*. OSDI '99. New Orleans, Louisiana, USA: USENIX Association, 1999, págs. 173-186. ISBN: 1880446391 (vid. pág. 26).

[Dai+19] X. Dai, J. Xiao, W. Yang, C. Wang y H. Jin. "Jidar: A Jigsaw-like Data Reduction Approach Without Trust Assumptions for Bitcoin System". En: *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. 2019, págs. 1317-1326 (vid. pág. 68).

[DKJ19] Ali Dorri, Salil S. Kanhere y Raja Jurdak. "MOF-BC: A memory optimized and flexible blockchain for large scale networks". En: *Future Generation Computer Systems* 92 (2019), págs. 357-373. ISSN: 0167-739X. DOI: 10.1016/j.future.2018.10.002 (vid. pág. 61).

[DM20] Damiano Di Francesco Maesa y Paolo Mori. "Blockchain 3.0 applications survey". En: *Journal of Parallel and Distributed Computing* 138 (2020), págs. 99-114. ISSN: 0743-7315. DOI: 10.1016/j.jpdc.2019.12.019 (vid. págs. 53, 64).

[Doc06] Filip Dochy. "A guide for writing scholarly articles or reviews for the Educational Research Review". En: *Educational Research Review* 4 (ene. de 2006), págs. 1-2 (vid. pág. 51).

[Dor+17] Ali Dorri, Marco Steger, Salil Kanhere y Raja Jurdak. "BlockChain: A Distributed Solution to Automotive Security and Privacy". En: *IEEE Communications Magazine (In press)* 55 (abr. de 2017). DOI: 10.1109/MCOM.2017.1700879 (vid. pág. 44).

[Dou02] John R. Douceur. "The Sybil Attack". En: *Revised Papers from the First International Workshop on Peer-to-Peer Systems*. IPTPS '01. Berlin, Heidelberg: Springer-Verlag, 2002, págs. 251-260. ISBN: 3540441794 (vid. pág. 35).

[DZZ19] H. Dai, Z. Zheng e Y. Zhang. "Blockchain for Internet of Things: A Survey". En: *IEEE Internet of Things Journal* 6.5 (2019), págs. 8076-8094. DOI: 10.1109/JIOT.2019.2920987 (vid. págs. 53, 59).

[FB99] A. Fox y E. A. Brewer. "Harvest, yield, and scalable tolerant systems". En: *Proceedings of the Seventh Workshop on Hot Topics in Operating Systems*. 1999, págs. 174-178 (vid. pág. 32).

[Fen+19] Qi Feng, Debiao He, Sherali Zeadally, Muhammad Khurram Khan y Neeraj Kumar. "A survey on privacy protection in blockchain system". En: *Journal of Network and Computer Applications* 126 (2019), págs. 45-58. DOI: 10.1016/j.jnca.2018.10.020 (vid. págs. 53, 58).

[FF18] Tiago M Fernández-Caramés y Paula Fraga-Lamas. "A Review on the Use of Blockchain for the Internet of Things". En: *IEEE Access* 6 (2018), págs. 32979-33001 (vid. págs. 53, 54).

[Gai+20] K. Gai, J. Guo, L. Zhu y S. Yu. "Blockchain Meets Cloud Computing: A Survey". En: *IEEE Communications Surveys Tutorials* (2020), págs. 1-1 (vid. pág. 47).

[GCZ18] K. Gai, K. R. Choo y L. Zhu. "Blockchain-Enabled Reengineering of Cloud Datacenters". En: *IEEE Cloud Computing* 5.6 (2018), págs. 21-25 (vid. pág. 47).

[GFM16] Vahid Garousi, Michael Felderer y Mika V. Mantyl. "The Need for Multivocal Literature Reviews in Software Engineering: Complementing Systematic Literature Reviews with Grey Literature". En: *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*. EASE '16. Limerick, Ireland: Association for Computing Machinery, 2016. ISBN: 9781450336918. DOI: 10.1145/2915970.2916008 (vid. pág. 51).

[GFM19] Vahid Garousi, Michael Felderer y Mika V. Mantyla. "Guidelines for including grey literature and conducting multivocal literature reviews in software engineering". En: *Information and Software Technology* 106 (2019), págs. 101-121. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2018.09.006 (vid. págs. 71, 72, 75).

[Grä+18] Wolfgang Gräther *et al.* "Blockchain for Education: Lifelong Learning Passport". En: *Proceedings of 1st ERCIM Blockchain Workshop 2018* (2018). URL: https://dl.eusset.eu/handle/20.500.12015/3163 (vid. pág. 46).

[Has+19a] Huru Hasanova, Ui-jun Baek, Mu-gon Shin, Kyunghee Cho y Myung-Sup Kim. "A survey on blockchain cybersecurity vulnerabilities and possible countermeasures". En: *International Journal of Network Management* 29.2 (2019). e2060 NEM-18-0162.R1, e2060. DOI: 10.1002/nem.2060. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nem.2060 (vid. págs. 53, 57).

[Has+19b] Fakhar ul Hassan *et al.* "Blockchain And The Future of the Internet: A Comprehensive Review". En: *ArXiv* abs/1904.00733 (2019) (vid. págs. 53, 58).

[HH20] Jens J. Hunhevicz y Daniel M. Hall. "Do you need a blockchain in construction? Use case categories and decision framework for DLT design options". En: *Advanced Engineering Informatics* 45 (2020), pág. 101094. ISSN: 1474-0346. DOI: j.aei.2020.101094 (vid. págs. 53, 70).

[HKL19] A. Hari, M. Kodialam y T. V. Lakshman. "ACCEL: Accelerating the Bitcoin Blockchain for High-throughput, Low-latency Applications". En: *IEEE INFOCOM 2019 - IEEE Conf. on Computer Communications*. 2019, págs. 2368-2376 (vid. pág. 67).

[HS91] Stuart Haber y W. Scott Stornetta. "How to Time-Stamp a Digital Document". En: *Advances in Cryptology-CRYPT0' 90 Lecture Notes in Computer Science* (1991), págs. 437-455. DOI: 10.1007/3-540-38424-3_32 (vid. pág. 7).

[Hua+20] Huawei Huang, Wei Kong, Sicong Zhou, Zibin Zheng y Song Guo. "A Survey of State-of-the-Art on Blockchains: Theories, Modelings, and Tools". En: *arXiv* (2020). arXiv: 2007.03520 (vid. págs. 53, 67).

[Isl+20] Iyolita Islam, Kazi Munim, Shahrima Oishwee, A.K.M. Islam y Muhammad Nazrul Islam. "A Critical Review of Concepts, Benefits, and Pitfalls of Blockchain Technology using Concept Map". En: *IEEE Access* 8 (abr. de 2020), págs. 68333-68341. DOI: 10.1109/ACCESS.2020.2985647 (vid. págs. 53, 64).

[Kee+07] Staffs Keele *et al. Guidelines for performing systematic literature reviews in software engineering*. Inf. téc. Technical report, Ver. 2.3 EBSE Technical Report. EBSE, 2007 (vid. pág. 65).

[Kit04] Barbara Kitchenham. "Procedures for performing systematic reviews". En: *Keele, UK, Keele University* 33.2004 (2004), págs. 1-26 (vid. págs. 62, 73).

[Kon+18] Ioannis Konstantinidis *et al.* "Blockchain for Business Applications: A Systematic Literature Review". En: *Business Information Systems*. Ed. por Witold Abramowicz y Adrian Paschke. Cham: Springer International Publishing, 2018, págs. 384-399 (vid. pág. 44).

[Ksh17] Nir Kshetri. "Blockchain's roles in strengthening cybersecurity and protecting privacy". En: *Telecommunications Policy* 41.10 (2017). Celebrating 40 Years of Telecommunications Policy – A Retrospective and Prospective View, págs. 1027-1038. ISSN: 0308-5961. DOI: 10.1016/j.telpol.2017.09.003 (vid. pág. 45).

[KUE17] Fabian Knirsch, Andreas Unterweger y Dominik Engel. "Privacy-preserving blockchain-based electric vehicle charging with dynamic tariff decisions". En: *Computer Science - Research and Development* 33.1-2 (2017), págs. 71-79. DOI: 10.1007/s00450-017-0348-5 (vid. pág. 44).

[Lam98] Leslie Lamport. "The part-time parliament". En: *ACM Transactions on Computer Systems* 16 (1998), págs. 133-169. DOI: 10.1145/279227.279229 (vid. pág. 7).

[LSP82] Leslie Lamport, Robert Shostak y Marshall Pease. "The Byzantine Generals Problem". en. En: *ACM Transactions on Programming Languages and Systems* 4.3 (jul. de 1982), págs. 382-401. ISSN: 01640925. DOI: 10.1145/357172.357176. (En línea, visitado el 19-07-2019) (vid. págs. 26, 47).

[Mer88] Ralph C. Merkle. "A Digital Signature Based on a Conventional Encryption Function". En: *Advances in Cryptology — CRYPTO '87*. Ed. por Carl Pomerance. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988, págs. 369-378. ISBN: 978-3-540-48184-3 (vid. pág. 13).

[MF18] Faraz Masood y Arman Faridi. "An Overview of Distributed Ledger Technology and its Applications". En: *International Journal of Computer Sciences and Engineering* 6 (oct. de 2018), págs. 422-427. DOI: 10.26438/ijcse/v6i10.422427 (vid. pág. 1).

[Mik17] E. Mik. "Smart contracts: terminology, technical limitations and real world complexity". En: *Law, Innovation and Technology* 9.2 (2017), págs. 269-300. DOI: 10.1080/17579961.2017.1378468 (vid. pág. 1).

[Moh+09] David Moher, Alessandro Liberati, Jennifer Tetzlaff y Douglas G Altman. "Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement". En: *BMJ* 339 (2009). DOI: 10.1136/bmj.b2535. eprint: https://www.bmj.com/content/339/bmj.b2535.full.pdf (vid. pág. 59).

[MPJ18] B. Mohanta, S. Panda y D. Jena. "An Overview of Smart Contract and Use Cases in Blockchain Technology". En: *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. Oct. de 2018. DOI: 10.1109/ICCCNT.2018.8494045 (vid. pág. 1).

[Muh+19] Saad Muhammad *et al.* "Exploring the Attack Surface of Blockchain: A Systematic Overview". En: *ArXiv* abs/1904.03487 (2019) (vid. pág. 36).

[Nak09] Satoshi Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System". En: *Cryptography Mailing list at https://metzdowd.com* (mar. de 2009) (vid. págs. 7, 8, 45).

[Nas+20] Mohamed Nassar, Khaled Salah, Muhammad Habib ur Rehman y Davor Svetinovic. "Blockchain for explainable and trustworthy artificial intelligence". En: *WIREs Data Mining and Knowledge Discovery* 10.1 (2020), e1340. DOI: 10.1002/widm.1340 (vid. pág. 47).

[NC17] Arvind Narayanan y Jeremy Clark. "Bitcoin's Academic Pedigree". En: *ACM Web Applications* (2017). URL: https://queue.acm.org/detail.cfm?id=3136559 (vid. págs. 8, 9).

[Net+19] G. Neto, W. Santos, P. Endo y R. Fagundes. "Multivocal literature reviews in software engineering: Preliminary findings from a tertiary study". En: *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. 2019, págs. 1-6 (vid. pág. 51).

[Nov+18] Petr Novotny *et al.* "Permissioned blockchain technologies for academic publishing". En: *Information Services & Use* 38 (ene. de 2018), págs. 1-13. DOI: 10.3233/ISU-180020 (vid. pág. 46).

[Pan+18] Alfonso Panarello, Nachiket Tapas, Giovanni Merlino, Francesco Longo y Antonio Puliafito. "Blockchain and IoT Integration: A Systematic Survey". En: *Sensors* 18 (ago. de 2018), pág. 2575. DOI: 10.3390/s18082575 (vid. págs. 53, 56).

[Ped92] Torben Pryds Pedersen. "Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing". En: *Advances in Cryptology — CRYPTO '91*. Ed. por Joan Feigenbaum. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, págs. 129-140 (vid. pág. 58).

[PK16] Namje Park y Namhi Kang. "Mutual authentication scheme in secure internet of things technology for comfortable lifestyle". En: *Sensors* 16.1 (2016), pág. 20. DOI: 10.3390/s16010020 (vid. pág. 62).

[PMR18] G. Perboli, S. Musso y M. Rosano. "Blockchain in Logistics and Supply Chain: A Lean Approach for Designing Real-World Use Cases". En: *IEEE Access* 6 (2018), págs. 62018-62028 (vid. pág. 46).

[Rab17] Kefa Rabah. "Challenges & opportunities for blockchain powered healthcare systems: A review". En: *Mara Research Journal of Medicine and Health Sciences* 1.1 (2017), págs. 45-52 (vid. pág. 40).

[Raj19] K. Raj. *Foundations of Blockchain*. Packt Publishing, 2019. ISBN: 9781789139396 (vid. pág. 1).

[RL18] Igor Radanović y Robert Likić. "Opportunities for Use of Blockchain Technology in Medicine". En: *Applied Health Economics and Health Policy* 16.5 (2018), págs. 583-590. DOI: 10.1007/s40258-018-0412-8 (vid. pág. 40).

[RST01] Ronald L. Rivest, Adi Shamir y Yael Tauman. "How to Leak a Secret". En: *Advances in Cryptology — ASIACRYPT 2001*. Ed. por Colin Boyd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, págs. 552-565 (vid. pág. 58).

[Sal+19] K. Salah, M. H. U. Rehman, N. Nizamuddin y A. Al-Fuqaha. "Blockchain for AI: Review and Open Research Challenges". En: *IEEE Access* 7 (2019), págs. 10127-10149 (vid. pág. 47).

[San10] Julio Sanchez-Meca. "Cómo realizar una revisión sistemática y un meta-análisis". En: *Aula abierta, ISSN 0210-2773, Vol. 38, Nº 2, 2010, pags. 53-64* 38 (ene. de 2010) (vid. pág. 51).

[Sid17] J. Sidhu. "Syscoin: A Peer-to-Peer Electronic Cash System with Blockchain-Based Services for E-Business". En: *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. 2017, págs. 1-6 (vid. pág. 45).

[SK17] Madhusudan Singh y Shiho Kim. "Blockchain Based Intelligent Vehicle Data sharing Framework". En: *ArXiv* abs/1708.09721 (2017) (vid. pág. 44).

[SP16] Siamak Solat y Maria Potop-Butucaru. *ZeroBlock: Timestamp-Free Prevention of Block-WithholdingAttack in Bitcoin*. Inf. téc. Sorbonne Universites, UPMC University of Paris 6, 2016. URL: https://hal.archives-ouvertes.fr/hal-01310088v2/document (vid. pág. 57).

[SRD20] Jayasree Sengupta, Sushmita Ruj y Sipra Das Bit. "A Comprehensive Survey on Attacks, Security Issues and Blockchain Solutions for IoT and IIoT". En: *Journal of Network and Computer Applications* 149 (2020), pág. 102481. ISSN: 1084-8045. DOI: 10.1016/j.jnca.2019.102481 (vid. págs. 53, 62).

[Sza97] Nick Szabo. "Formalizing and Securing Relationships on Public Networks". En: *First Monday* (1997). URL: https://journals.uic.edu/ojs/index.php/fm/article/view/548 (vid. pág. 26).

[Tah+17] Rashid Tahir *et al.* "Mining on Someone Else's Dime: Mitigating Covert Mining Operations in Clouds and Enterprises". En: *Research in Attacks, Intrusions, and Defenses*. Ed. por Marc Dacier, Michael Bailey, Michalis Polychronakis y Manos Antonakakis. Cham: Springer International Publishing, 2017, págs. 287-310. ISBN: 978-3-319-66332-6 (vid. pág. 39).

[TS16] F. Tschorsch y B. Scheuermann. "Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies". En: *IEEE Communications Surveys Tutorials* 18.3 (2016), págs. 2084-2123 (vid. pág. 4).

[UKGov16] UK Government Chief Scientific Adviser. *Distributed Ledger Technology: beyond blockchain*. Inf. téc. Goverment Office for Science UK, 2016 (vid. pág. 1).

[VKM18] Hoang Tam Vo, Ashish Kundu y Mukesh K. Mohania. "Research Directions in Blockchain Data Management and Analytics". En: *EDBT*. 2018 (vid. pág. 46).

[Wan+18] Licheng Wang, Xiaoying Shen, Jing Li, Jun Shao y Yixian Yanga. "Cryptographic primitives in blockchains". En: *Journal of Network and Computer Applications* (nov. de 2018). URL: https://www.sciencedirect.com/science/article/pii/S108480451830362X (vid. pág. 13).

[Wan+19] Wenbo Wang *et al.* "A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks". En: *IEEE Access* 7 (2019), págs. 22328-22370. DOI: 10.1109/access.2019.2896108 (vid. págs. 53, 57).

[Wan+20c] Qin Wang, Xinqi Zhu, Yiyang Ni, Li Gu y Hongbo Zhu. "Blockchain for the IoT and industrial IoT: A review". En: *Internet of Things* 10 (2020). Special Issue of the Elsevier IoT Journal on Blockchain Applications in IoT Environments, pág. 100081. ISSN: 2542-6605. DOI: 10.1016/j.iot.2019.100081 (vid. págs. 53, 66).

[Wat16] R. Wattenhofer. *The Science of the Blockchain*. 1st. North Charleston, SC, USA: CreateSpace Independent Publishing Platform, 2016. ISBN: 1522751831 (vid. pág. 1).

[Wer18] K. Werbach. "Trust, but verify: Why the blockchain needs the law". En: *Berkeley Technology Law Journal* 33.2 (2018). Cited By :24, págs. 487-550. URL: www.scopus.com (vid. pág. 2).

[WG18] K. Wust y A. Gervais. "Do you Need a Blockchain?" En: *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. 2018, págs. 45-54 (vid. págs. 40, 41).

[XH20] Y. Xu e Y. Huang. "Segment Blockchain: A Size Reduced Storage Mechanism for Blockchain". En: *IEEE Access* 8 (2020), págs. 17434-17441 (vid. pág. 68).

[Yan+19] Lei Yang *et al.* "Prism: Scaling bitcoin by 10,000 x". En: *arXiv* (2019). arXiv: 1909.11261 (vid. pág. 68).

[Yli+16] Jesse Yli-Huumo, Deokyoon Ko, Sujin Choi, Sooyong Park y Kari Smolander. "Where Is Current Research on Blockchain Technology?—A Systematic Review". En: *PLOS ONE* 11.10 (oct. de 2016), págs. 1-27. DOI: 10.1371/journal.pone.0163477 (vid. pág. 4).

[Zha+18] Peng Zhang, Douglas C. Schmidt, Jules White y Gunther Lenz. "Chapter One - Blockchain Technology Use Cases in Healthcare". En: *Blockchain Technology: Platforms, Tools and Use Cases*. Ed. por Pethuru Raj y Ganesh Chandra Deka. Vol. 111. Advances in Computers. Elsevier, 2018, págs. 1-41. DOI: 10.1016/bs.adcom.2018.03.006 (vid. pág. 40).

[Zhe+18] Zibin Zheng, Shaoan Xie, Hong Ning Dai, Xiangping Chen y Huaimin Wang. "Blockchain challenges and opportunities: a survey". En: *International Journal of Web and Grid Services* 14.4 (2018), pág. 352. DOI: 10.1504/ijwgs.2018.095647 (vid. págs. 53, 56).

[ZXL19] Rui Zhang, Rui Xue y Ling Liu. *Security and Privacy on Blockchain*. 2019. arXiv: 1903.07602 [cs.CR] (vid. págs. 1, 20).

## Otras fuentes en línea consultadas

[Bon15] J. Bonneau. *How long does it take for a Bitcoin transaction to be confirmed?* Coin Center. 2015. URL: https://www.coincenter.org/education/crypto-regulation-faq/how-long-does-it-take-for-a-bitcoin-transaction-to-be-confirmed/ (en línea, visitado el 15-03-2020) (vid. págs. 1, 20).

[Bro20] Julian Browne. *Brewer's CAP Theorem*. Julianbrowne.com. 2020. URL: http://www.julianbrowne.com/article/brewers-cap-theorem (vid. pág. 32).

[Bru14] J. D. Bruce. *The Mini-Blockchain Scheme*. Cryptonite.info. 2014. URL: http://cryptochainuni.com/wp-content/uploads/The-Mini-Blockchain-Scheme.pdf (en línea, visitado el 30-07-2020) (vid. pág. 55).

[Corda15] Corda. *Corda — Open Source Blockchain Platform for Business*. Corda. 2015. URL: https://www.corda.net/ (en línea, visitado el 30-08-2020) (vid. pág. 30).

[CTIC17] CTIC Centro Tecnológico. *Blockchain for Lift Maintenance and Security Audit*. CTIC Centro Tecnológico. 2017. URL: https://www.fundacionctic.org/en/news/blockchain-lift-maintenance-and-security-audit (en línea, visitado el 23-07-2020) (vid. pág. 1).

[CTIC18] CTIC Centro Tecnológico. *CarTrustChain: tecnología blockchain para acabar con el fraude en los cuentakilómetros*. Blockchain Services. 2018. URL: http://www.blockchainservices.es/novedades/cartrustchain-tecnologia-blockchain-para-acabar-con-el-fraude-en-los-cuentakilometros/ (en línea, visitado el 23-07-2020) (vid. pág. 1).

[Dai98] Wei Dai. *B-Money*. Wei Dai's Home Page. 1998. URL: http://www.weidai.com/bmoney.txt (en línea, visitado el 08-09-2019) (vid. pág. 7).

[Dan17] Dantheman. *DPOS Consensus Algorithm - The Missing White Paper*. 2017. URL: https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper (en línea, visitado el 15-07-2020) (vid. pág. 24).

[Dob18] D. Dobson. *The 4 Types of Blockchain Networks Explained*. Iltanet. 2018. URL: https://iltanet.org/blogs/deborah-dobson/2018/02/13/the-4-types-of-blockchain-networks-explained?ssopc=1 (en línea, visitado el 27-06-2020) (vid. pág. 1).

[Dol16] Doloto. *What are Blockchain Confirmations?* Reddit. 2016. URL: https://www.reddit.com/r/ethereum/comments/4eplsv/how_many_confirms_is_considered_safe_in_ethereum/ (en línea, visitado el 15-03-2020) (vid. pág. 21).

[Ethos18] Ethos. *What are Blockchain Confirmations?* Ethos. 2018. URL: https://www.ethos.io/what-are-blockchain-confirmations/ (en línea, visitado el 15-03-2020) (vid. págs. 1, 20).

[Gartner19] K. Panetta. *Hyperautomation, blockchain, AI security, distributed cloud and autonomous things drive disruption and create opportunities in this year's strategic technology trends*. Gartner. 2019. URL: https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2020/ (en línea, visitado el 20-08-2020) (vid. pág. 108).

[Hos17] C. Hoskinson. *Why we are building Cardano*. 2017. URL: https://cardano.org/why/assets/WhyCardanoEN.pdf (en línea, visitado el 20-01-2020) (vid. pág. 8).

[IPFS15] *IPFS is the Distributed Web*. IPFS, Protocol. 2015. URL: https://ipfs.io/ (en línea, visitado el 14-09-2019) (vid. págs. 91, 112).

[MAB17] L. Meijueiro, R. Alonso y A. Berdasco. *El potencial de las tecnologías Blockchain*. CTIC Centro Tecnológico. 2017. URL: https://www2.fundacionctic.org/sites/default/files/files/CTIC-Jornada23M%5C%20AplicarBlckChn.pdf (en línea, visitado el 25-07-2020) (vid. pág. 1).

[Mei18] L. Meijueiro. *Luis Meijueiro: "Si aceptas las grandes empresas sabes que usarán tus datos"*. La Nueva España. 2018. URL: https://www.lne.es/centro/2018/12/14/luis-meijueiro-aceptas-grandes-empresas/2396388.html (en línea, visitado el 25-07-2020) (vid. pág. 1).

[Mei19a] L. Meijueiro. *Asturias traza la sidra con Blockchain*. Blockchain Economia. 2019. URL: https://www.blockchaineconomia.es/luis-meijueiro-asturias-sidra-blockchain/ (en línea, visitado el 25-07-2020) (vid. pág. 1).

[Mei19b] L. Meijueiro. *Servicios innovadores en Blockchain*. I Congreso Asturias Blockchain. 2019. URL: https://bitcoinorg.info/%5C#our-team-2 (en línea, visitado el 25-07-2020) (vid. pág. 1).

[ML20] L. Meijueiro y R. Luque. *Blockchain veo, Blockchain quiero. Blockchain en el sector público ¿una posibilidad real?* Novatica - Revista de la Asociación de Técnicos en Informática. 2020. URL: https://www.novatica.es/blockchain-veo-blockchain-quiero-blockchain-en-el-sector-publico-una-posibilidad-real/ (en línea, visitado el 25-07-2020) (vid. pág. 1).

[Multichain15] Multichain. *Multichain — Open source blockchain platform*. Multichain. 2015. URL: https://www.multichain.com/ (en línea, visitado el 30-08-2020) (vid. pág. 30).

[Pre17] A. Preukschat. *Los tipos de Blockchain: públicas, privadas e híbridas (y II)*. Iecisa. 2017. URL: https://www.iecisa.com/es/blog/Post/Los-tipos-de-Blockchain-publicas-privadas-e-hibridas-y-II/ (en línea, visitado el 28-07-2020) (vid. pág. 1).

[StackExchange20] *Why is it important that the hash algorithm is puzzle friendly?* Bitcoin Stack Exchange. 2020. URL: https://bitcoin.stackexchange.com/questions/76411/why-is-it-important-that-the-hash-algorithm-is-puzzle-friendly (en línea, visitado el 20-06-2020) (vid. pág. 12).

[Sza05] N. Szabo. *Bit Gold*. 2005. URL: https://nakamotoinstitute.org/bit-gold/ (en línea, visitado el 15-01-2020) (vid. pág. 7).

[Teloscope19] The Teloscope. *The Teloscope*. The Teloscope. 2019. URL: https://theteloscope.io/ (en línea, visitado el 23-07-2020) (vid. pág. 1).

[Tender18] *Tendermint*. 2018. URL: https://tendermint.com/ (en línea, visitado el 04-08-2019) (vid. pág. 26).

[Thi19] M. Thibodeau. *3 Types of Blockchain Explained*. Hedgetrade. 2019. URL: https://hedgetrade.com/3-types-of-blockchain-explained/ (en línea, visitado el 27-07-2020) (vid. pág. 1).

[Vos19a] S. Voshmgir. *Blockchains & Distributed Ledger Technologies*. Blockchainhub Berlin. 2019. URL: https://blockchainhub.net/blockchains-and-distributed-ledger-technologies-in-general/ (en línea, visitado el 28-07-2020) (vid. pág. 1).

[Vos19b] S. Voshmgir. *Smart Contracts*. Blockchainhub Berlin. 2019. URL: http://blockchainhub.net/smart-contracts/ (en línea, visitado el 28-07-2020) (vid. pág. 1).

[VS18] S. Viswanathan y A Shah. *The Scalability Trilemma in Blockchain*. NeonVest. 2018. URL: https://medium.com/@aakash_13214/the-scalability-trilemma-in-blockchain-75fb57f646df (en línea, visitado el 27-07-2020) (vid. pág. 1).

[Zil17] *Zilliqa*. 2017. URL: https://zilliqa.com/ (en línea, visitado el 04-08-2019) (vid. pág. 26).