ISSN: 0123-3033 – ISSN-e: 2027-8284

INGENIERIA Y COMPETITIVIDAD
REVISTA CIENTÍFICA Y TECNOLÓGICA

Vol. 24 No. 1-2022 - DOI: 10.25100/iyc.24i1.11027

SYSTEMS ENGINEERING

A Comparison of Two Blockchain Platforms: Bitcoin and Hyperledger Fabric

INGENIERÍA DE SISTEMAS

Comparación de Dos Plataformas de Blockchain: Bitcoin y Hyperledger Fabric

Francisco J. Moreno-Arboleda[1§], Johan S. Rodríguez-Camacho[1], Daniel Giraldo-Muñoz[1]

[1]Universidad Nacional de Colombia, Sede Medellín, Facultad de Minas, Departamento de Ciencias de la Computación y de la Decisión, Medellín, Colombia.

[§]*fjmoreno@unal.edu.co, josrodriguezca@unal.edu.co, dagiraldomu@unal.edu.co*

Recibido: 18 de febrero de 2021 – **Aceptado**: 17 de junio de 2021

Abstract

In this paper we review several works that compare blockchain platforms. For each one, we present the comparison criteria and its emphasis. Additionally, we compared two of the most popular blockchain platforms: Bitcoin and Hyperledger Fabric. With regard to the analyzed works, none of them emphasize on the similarities and differences regarding the flow of a transaction, a complex process in Hyperledger Fabric due to the number of concepts and components involved. To facilitate the comparison, we show the flow of a transaction in both platforms. The analyzed works also do not compare the functions of the components of each platform, for example, of the nodes and their different types, nor do they detail the structure of the blocks. Finally, an effort was made to establish a common vocabulary between the two platforms.

Keywords: *blockchain, bitcoin, hyperledger fabric, transactions, distributed databases*.

Resumen

En este artículo se presenta una recopilación de varios trabajos de comparación de plataformas de *blockchain*. Para cada uno se describen los criterios de comparación usados y su énfasis. Además, se compararon dos de las plataformas de *blockchain* más populares: Bitcoin y Hyperledger Fabric. Con respecto a los trabajos analizados, ninguno enfatiza en las similitudes y diferencias en cuanto al flujo de una transacción, un proceso complejo en Hyperledger Fabric

Como citar:
Moreno-Arboleda F, Rodríguez-Camacho JS, Giraldo-Muñoz D. Comparación de Dos Plataformas de Blockchain: Bitcoin y Hyperledger Fabric. INGENIERÍA Y COMPETITIVIDAD. 2022;24(1):e30311027. https://doi.org/10.25100/iyc.v24i1.11027

debido a la cantidad de conceptos y de componentes involucrados. Para facilitar la comparación, se ilustra el flujo de una transacción en las dos plataformas. Los trabajos analizados tampoco comparan las funciones de los componentes de cada plataforma, por ejemplo, de los nodos y sus diferentes tipos, ni detallan la estructura de los bloques. Además, se trató de establecer un vocabulario común entre las dos plataformas.

Palabras clave: *cadena de bloques, bitcoin, hyperledger fabric, transacciones, bases de datos distribuidas*.

1. Introducción

Una *blockchain* es una base de datos descentralizada y distribuida (replicada) formada por una lista enlazada de bloques, es decir, una cadena de bloques, donde cada bloque contiene un conjunto de transacciones, por ejemplo, una transferencia de dinero entre dos cuentas, un voto de una persona, un cambio en el inventario de un producto, entre otras. En este tipo de bases de datos los datos de las transacciones confirmadas son casi imposibles de modificar [(1)] y las transacciones son aprobadas de manera autónoma por un conjunto de participantes, es decir, sin una autoridad central. Precisamente, la motivación inicial de *blockchain* fue permitir que los usuarios intercambiasen datos de manera confiable sin necesidad de un intermediario. Por ejemplo, que intercambiasen dinero sin necesidad de un banco. Lo anterior originó las criptomonedas como Bitcoin en 2008 y Ethereum en 2015. Una criptomoneda es según el Banco Central Europeo: "*un tipo de dinero no regulado, digital, que se emite y por lo general es controlado por sus desarrolladores, utilizado y aceptado entre los miembros de una comunidad virtual específica*". Desde entonces, la popularidad de *blockchain* ha crecido, así como la variedad de casos de uso en los que se puede aplicar, por ejemplo, para la gestión de votaciones [(2)], cadenas de suministro [(3)], bienes raíces [(4)], tiquetes [(5)], programación de pagos [(6)], entre muchos otros.

En este artículo se presenta una recopilación de diversos trabajos de comparación de plataformas de *blockchain*. Para cada uno se describen los criterios de comparación usados y su énfasis. Como contribución adicional, se comparan dos de las plataformas de *blockchain* más populares: la *blockchain* de Bitcoin, la primera plataforma de *blockchain* (aunque los fundamentos teóricos se habían planteado desde 1991 [(7)]) y Hyperledger Fabric. Hyperledger Fabric se ha posicionado como una de las principales plataformas para la gestión de *blockchains*; empresas como Amazon, IBM y Oracle ofrecen productos que permiten crear y administrar una *blockchain* apoyados en esta plataforma. La *blockchain* de Bitcoin es de acceso público (cualquier persona u organización se puede vincular a ella) mientras que la de Hyperledger Fabric es de acceso privado (solo pueden acceder los miembros autorizados).

Ninguno de los trabajos analizados de comparación enfatiza en las similitudes y diferencias en cuanto al flujo de una transacción, un proceso complejo en Hyperledger Fabric, debido a la cantidad de conceptos y de componentes involucrados. Aquí, se describe el flujo de una transacción en las dos plataformas, las fases por las que pasa y los componentes que actúan en cada una. Además, para facilitar la comparación se presenta de forma gráfica el diagrama de flujo de una transacción en cada plataforma.

Los trabajos analizados tampoco comparan las funciones de los componentes de cada plataforma, por ejemplo, de los nodos y sus diferentes tipos, ni detallan la estructura de los bloques. Finalmente, se trató de establecer un vocabulario común entre las dos plataformas. Esta labor implicó unificar términos extraídos de diferentes fuentes, ya que se suelen usar diferentes términos para referirse a un mismo concepto o se suele usar

el mismo término para referirse a conceptos diferentes.

El artículo se estructura así: en la Sección 2 se presentan los trabajos de comparación de plataformas de *blockchain*. En la Sección 3 se presenta la *blockchain* de Bitcoin y en la Sección 4 Hyperledger Fabric. En la Sección 5 se comparan las dos plataformas. Se concluye en la Sección 6, donde se describe además el trabajo futuro.

2. Comparación de plataformas de *blockchain*

En esta sección se presentan trabajos de comparación de plataformas de *blockchain*. Primero se presentan algunas definiciones.

2.1 Definiciones

a) Tasa de confirmación (del inglés *throughput*): número de transacciones confirmadas por segundo.

b) Latencia: tiempo promedio que transcurre entre dos bloques añadidos a la cadena.

c) Escalabilidad: medida que indica cuánto se afecta la tasa de confirmación y la latencia a medida que se incrementa el número de nodos de la red.

d) Contrato inteligente (CI): conjunto de funciones que representan la lógica del negocio y que se pueden ejecutar en la plataforma *blockchain*. En la Figura 1 se muestra en pseudocódigo un CI para la gestión de automóviles (contrato auto). El contrato tiene tres funciones: i) consultar: permite consultar los datos de un automóvil, ii) transferir: permite cambiar el dueño de un automóvil y iii) actualizarPrecio: permite cambiar el precio de un automóvil. Los CI se suelen programar en lenguajes como Go, Java, JavaScript, Solidity y Vyper, entre otros.

e) Gobernanza: conjunto de reglas, acciones y responsables que gestionan el desarrollo de la plataforma considerando las necesidades y demandas del entorno.

```
Contrato auto:

  consultar(auto):
    buscar(auto);
    retornar auto;

  transferir(auto, comprador, vendedor):
    consultar(auto);
    IF auto.dueño = vendedor THEN
      auto.dueño = comprador;
      guardar(auto);
    END IF;
    retornar auto;

  actualizarPrecio(auto, precio):
    consultar(auto);
    auto.precio = precio;
    guardar(auto);
```

Figura 1. *Pseudocódigo de un CI para la gestión de automóviles. Fuente: elaboración propia*

2.2. Trabajos de comparación

En [8] se propone un marco de evaluación para plataformas de *blockchains* privadas enfocado en tasa de confirmación, latencia, escalabilidad y tolerancia a fallos. Con este marco se evalúan tres plataformas: Ethereum, Parity y Hyperledger Fabric. Se presenta una tabla comparativa donde se incluyen otras ocho plataformas: Eris, Ripple, ScalableBFT, Stellar, Dfinity, Tezos, Corda y Sawtooth Lake. Se usaron cinco criterios: a) propósito (CI o criptomoneda), b) plataforma para la ejecución del CI (por ejemplo, Docker, Java Virtual Machine, Ethereum Virtual Machine), c) lenguajes de programación de los CIs, d) modelo de verificación de saldos [9]: modelo UTXO, donde se calcula el saldo de una cuenta a partir del histórico de sus transacciones y modelo basado en cuentas, donde se lleva el saldo actual de cada cuenta y e) protocolo de consenso [10] como

prueba de trabajo (PoW por su sigla en inglés), tolerancia práctica a fallas bizantinas (PBFT por su sigla en inglés) y prueba de participación (PoS por su sigla en inglés).

En [10] se discuten y analizan seis plataformas de *blockchain*: Hyperledger Fabric, Hyperledger Sawtooth, Bitcoin, Ethereum, Corda e IOTA. Se analizan los protocolos de consenso y se presenta una tabla comparativa para las seis plataformas con trece criterios: a) acceso (público o privado), b) gestión de permisos, c) descentralización, d) y e) consumo de recursos de cómputo y de la red, f) escalabilidad, g) tasa de confirmación, h) latencia, i) inmutabilidad de los datos, j) tolerancia a ataques, k) seguridad de los datos, l) gestión de CIs y m) criptomonedas soportadas.

En [11] se analizan las principales diferencias entre Hyperledger Fabric, Corda y Ethereum. Se presenta una tabla comparativa con cinco criterios: a) tipo (genérica o especializada, por ejemplo, para finanzas como Corda), b) gobernanza, c) acceso (público o privado), d) protocolo de consenso, e) gestión de CIs y f) criptomonedas soportadas.

En [12] se comparan cinco plataformas de *blockchain*: Ethereum, IBM Open Blockchain, Intel Sawtooth Lake, Sidechain Elements y Eris. Se definen ocho criterios para la comparación: a) facilidad de uso y aprendizaje, b) soporte y documentación, c) madurez de la plataforma y tamaño de la comunidad de desarrolladores, d) casos de uso, e) escalabilidad y tasa de confirmación, f) protocolo de consenso e incentivo, g) criptomonedas soportadas y h) seguridad de los datos y privacidad de los usuarios.

Posiblemente el trabajo más cercano al presente artículo es [13]. Allí se analizan tres plataformas de *blockchain*: Bitcoin, Ethereum y Hyperledger Fabric. Se describe brevemente su funcionamiento. El trabajo se enfoca en el análisis de rendimiento de cada una. Se presenta una tabla comparativa con 14 criterios: a) propósito (CI o criptomoneda), b) tipos de datos que almacena, c) lenguajes de programación de los CIs, d), e) y f) aspectos de acceso: público o privado, vinculación de usuarios y autoridades para el ingreso, g) criptomonedas soportadas, h) transparencia de las decisiones, i) y j) seguridad y gestión de claves, k) latencia, l) y m) tiempo de aprobación y tamaño de una transacción y n) protocolo de consenso.

En [14] se comparan dos versiones de Hyperledger Fabric, la 0.6 y la 1.0. El énfasis es el rendimiento. Adicionalmente, se comparan Ethereum y Hyperledger Fabric con respecto al tiempo de aprobación de una transacción y a la tasa de confirmación. Finalmente, se comparan Hyperledger Fabric, Ethereum y Parity con respecto a la escalabilidad.

En [15] se comparan Hyperledger Fabric y Ethereum. Se describe su funcionamiento básico y sus componentes. Se hace una comparación con seis criterios: a) seguridad y permisos, b) protocolo de consenso, c) ecosistema, es decir, herramientas de desarrollo, d) casos de uso, e) lenguajes de programación de los CIs y f) facilidad de despliegue.

En [16] se explica la tecnología *blockchain*. Se analizan ocho plataformas de *blockchain*: Bitcoin, Ethereum, Cardano, IOTA, Hyperledger Fabric, Hyperledger Indy, MultiChain y Corda. Se enfatiza en la aplicación de estas plataformas para el internet de las cosas y en aplicaciones ("*Oracles*") que interactúan con estas plataformas.

En [17] se comparan nueve plataformas de *blockchain*: Bitcoin, Corda, Multichain, Ethereum, Hyperledger Fabric, Sawtooth, Neo, Quorum y NXT. Se definieron tres categorías para la comparación, cada una con sus criterios: a) técnica. Incluye seguridad, protocolo de consenso, lenguajes de programación de los CIs, latencia y tasa de confirmación, b) empresarial.

Incluye gobernanza, licencias, soporte, costos y consumo de recursos y c) indicadores de estabilidad. Incluye desarrollo de nuevas funcionalidades, popularidad, solidez de su criptomoneda y de la red, inversores, calificación de los expertos y principales clientes que usan la plataforma.

Finalmente, en [18] se analizan tres plataformas de *blockchain*: Bitcoin, Ethereum y Hyperledger Fabric. El análisis está enfocado en tres criterios: escalabilidad, latencia y tasa de confirmación. En las Tablas 1 y 2 se resumen los trabajos comparados.

3. La *Blockchain* de Bitcoin

Bitcoin fue la primera *blockchain*. Fue creada en 2008 por Satoshi Nakamoto (un individuo o grupo de individuos que permanece anónimo hasta la fecha) y está orientada a la transferencia de dinero sin intermediarios. Su criptomoneda se denomina *bitcoin* la cual se suele abreviar como BTC o XTC. El precio de un *bitcoin* está determinado por la oferta y la demanda, de forma similar a una acción de la bolsa de valores. Bitcoin no le pertenece a un individuo o a una compañía. La plataforma se mantiene en funcionamiento gracias a sus usuarios.

3.1. Elementos básicos

Los elementos básicos de la *blockchain* de Bitcoin son:

- *Transacción:* es el intercambio de *bitcoins* entre dos partes interesadas. Por ejemplo, un usuario envía dos *bitcoins* a otro usuario.
- *Bloque:* es un conjunto de transacciones (cada una con sus datos) y de datos adicionales. Un bloque incluye [19]: a) el número identificador único del bloque, b) la fecha de creación del bloque, c) los datos de las transacciones, d) el *hash* actual: es el número *hash* del bloque, e) el *hash* anterior: es el número *hash* del bloque anterior. Cada bloque de la cadena, excepto el primero, tiene el número *hash* del bloque anterior; de esta forma, los bloques se enlazan y forman una cadena y f) *nonce*: es un número arbitrario. Su nombre proviene del inglés, "*number only used once*" es decir, número que se usa una sola vez. Su propósito se explica más adelante. En la Figura 2 se muestran dos bloques.
- *Red:* es un conjunto de computadores (nodos) interconectados que participan en la gestión de la *blockchain*.
- *Nodo:* es un computador de la red. Cada nodo tiene una copia de la *blockchain*. Un nodo puede a) validar las transacciones, b) participar en el proceso de aprobación que autoriza que se añada un bloque a la cadena y c) armar un bloque e intentar que este sea añadido a la cadena (acá el nodo actúa como un minero, véase a continuación PoW), entre otras funciones.
- *PoW:* es un problema matemático que tiene que resolver el minero para lograr que un bloque que él ha armado sea añadido a la cadena. Los pasos para resolver la PoW son [1]:

    1. Generar un número arbitrario, es decir, el *nonce*.
    2. Añadir el *nonce* al final de todos los datos del bloque.
    3. Generar un número *hash* mediante el método SHA256 [20]. Para generarlo se consideran los datos de las transacciones del bloque, la fecha, el *hash* anterior y el *nonce*. Si el número *hash* generado comienza por un determinado número de ceros (condición impuesta por el sistema, esta es en sí la PoW) entonces el minero ha resuelto la PoW y este número *hash* es el *hash* actual. Si no, el minero deberá empezar desde el paso 1.

Tabla 1. *Comparación de los trabajos analizados: criterios*

Referencia / Criterio	(8)	(10)	(11)	(12)	(13)	(14)	(15)	(17)	(18)
Acceso		✓	✓		✓				
Casos de uso				✓			✓		
Consumo de recursos		✓						✓	
Criptomonedas	✓	✓	✓	✓	✓				
Escalabilidad	✓	✓		✓		✓			✓
Gestión de CI	✓	✓	✓		✓		✓	✓	
Gobernanza			✓					✓	
Latencia	✓	✓			✓	✓		✓	✓
Protocolo de consenso	✓		✓	✓	✓		✓	✓	
Seguridad		✓		✓	✓		✓	✓	
Soporte				✓				✓	
Tasa de confirmación	✓	✓		✓		✓		✓	✓

Fuente: elaboración propia

Tabla 2. *Comparación de los trabajos analizados: criterios adicionales y plataformas.*

Referencia	Criterios adicionales	Plataformas de *blockchain* analizadas
(8)	Modelo de verificación de saldos, tolerancia a fallos.	Ethereum, Parity, Hyperledger Fabric, Eris, Ripple, ScalableBFT, Stellar, Dfinity, Tezos, Corda y Sawtooth Lake.
(10)	Descentralización, inmutabilidad de los datos, tolerancia a ataques.	Hyperledger Fabric, Hyperledger Sawtooth, Bitcoin, Ethereum, Corda e IOTA.
(11)	Tipo de *blockchain* (genérica o especializada).	Hyperledger Fabric, Corda y Ethereum.
(12)	Facilidad de uso, madurez.	Ethereum, IBM Open Blockchain, Intel Sawtooth Lake, Sidechain Elements y Eris.
(13)	Propósito, tamaño de una transacción, tipos de datos, transparencia en las decisiones.	Bitcoin, Ethereum y Hyperledger Fabric.
(14)	Ninguno.	Ethereum y Hyperledger Fabric.
(15)	Ecosistema, facilidad de despliegue.	Hyperledger Fabric y Ethereum.
(16)	Aplicación de estas plataformas para el internet de las cosas.	Bitcoin, Ethereum, Cardano, IOTA, Hyperledger Fabric, Hyperledger Indy, MultiChain y Corda.
(17)	Estabilidad.	Bitcoin, Corda, Multichain, Ethereum, Hyperledger Fabric, Sawtooth, Neo, Quorum y NXT.
(18)	Ninguno.	Bitcoin, Ethereum y Hyperledger Fabric.

Fuente: elaboración propia

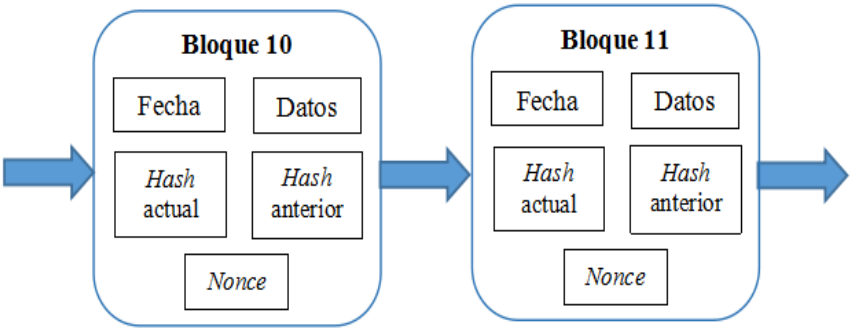

Figura 2. *Dos bloques y su estructura. Fuente: adaptado a partir de [(1)].*

- *Incentivo:* es un premio, en *bitcoins*, que se concede a un minero cuando logra añadir a la cadena el bloque que él armó. Se creó para compensar el consumo de recursos (procesamiento computacional) que supone la solución de la PoW. Así, se espera que más personas u organizaciones, que aportan sus computadores como mineros, se unan a la red.
- *Fondo (del inglés pool):* es un área donde se encuentran las transacciones pendientes por ser incluidas en la *blockchain*. Cada nodo tiene una copia del fondo. El fondo se actualiza cuando: a) llegan nuevas transacciones y b) se añade un bloque a la cadena, las transacciones de dicho bloque son borradas del fondo [21].

3.2. Flujo de una transacción

A grandes rasgos, la *blockchain* de Bitcoin funciona como se describe a continuación [1]:

1. Los usuarios del sistema emiten transacciones.
2. Las transacciones del paso 1 son puestas en el fondo al cual acceden los mineros (nodos).
3. Cada minero puede acceder al fondo y puede armar un bloque con transacciones. En este punto, hay una competencia entre los mineros ya que cada uno desea añadir a la cadena el bloque que él armó para ganarse el incentivo. Por ello, el sistema plantea una PoW que cada minero debe resolver. El primero que la resuelva, será quien tiene la oportunidad de añadir su bloque a la cadena.
4. Cuando un minero resuelve la PoW, este informa a los otros mineros. Estos validan el bloque: se verifican las transacciones y la solución de la PoW; si es aceptado (al menos por el 51% de los mineros), el bloque se añade a la cadena, el minero que resolvió la PoW obtiene el incentivo y comienza de nuevo la competencia para generar el próximo bloque.

El proceso se muestra en la Figura 3. Allí se muestran en el paso 3 tres mineros (nodos 281, 345 y 433) que tratan de resolver la PoW para generar el próximo bloque de la cadena, este será el bloque 23. Para ello, cada minero selecciona a partir del fondo un conjunto de transacciones (denotadas Trans. en la figura) y con esta arma un bloque. Nótese que las transacciones de los bloques de dos mineros no son necesariamente las mismas. Luego cada minero intenta resolver la PoW. Si un minero la resuelve (en la figura se representa con un par de dados que muestran dos seis, ver el nodo 433), los otros mineros detienen la búsqueda de la solución para su PoW, proceden a validar las transacciones y la solución de la PoW del nodo 433 y si al menos el 51% de los mineros aprueba, el bloque del nodo 433 se añade a la cadena como se muestra en la figura en el paso 4.

Cuando un minero va a armar un bloque, él puede seleccionar cualquier transacción del fondo. La elección depende de la comisión que cada transacción otorgue; así, los mineros procuran armar un bloque con las transacciones que mayor comisión ofrezcan, de tal forma que, si el minero resuelve la PoW de su bloque, él ganará todas las comisiones (de las transacciones de su bloque), además del incentivo.

A continuación, se presentan los conceptos básicos de la *blockchain* de Hyperledger Fabric.

4. Hyperledger Fabric

Hyperledger Fabric es un proyecto de código abierto para gestionar plataformas *blockchain* empresariales. Fue creada en 2015 por la Linux Foundation, un consorcio sin ánimo de lucro dedicado al crecimiento de Linux y al desarrollo colaborativo de *software*. Hyperledger Fabric está orientada a un amplio rango de casos de uso. Al ser un proyecto de código abierto, se puede descargar y usar sin costo. De esta forma, cualquier individuo o compañía puede gestionar sus *blockchains* por medio de esta plataforma.

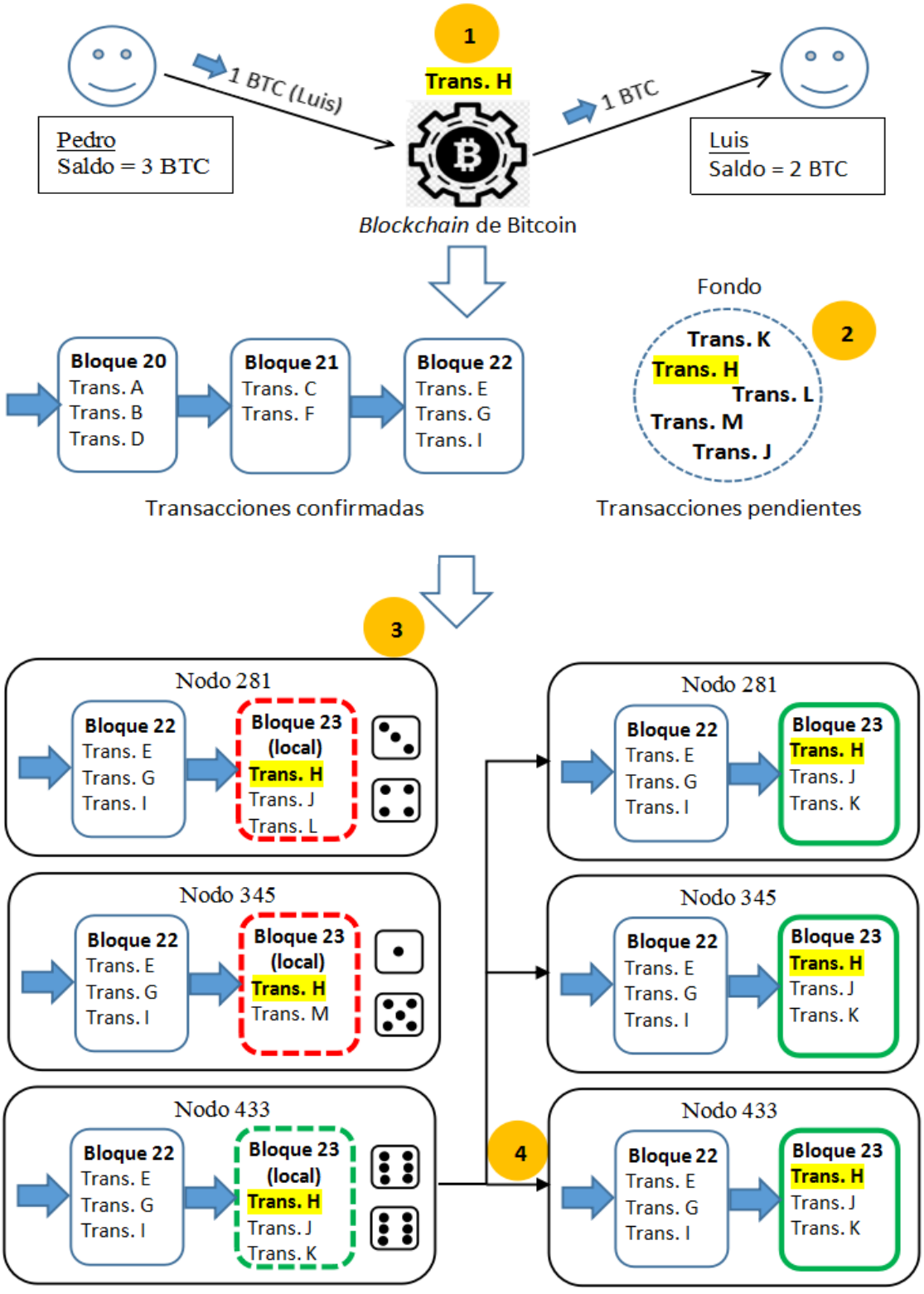

Figura 3. *Flujo de una transacción en Bitcoin. Fuente: adaptado a partir de* [(1)].

4.1 Elementos básicos

Los elementos básicos de *Hyperledger Fabric* son:

- *Transacción:* es el registro que se genera a partir de la invocación de una función de un CI y de su posterior aceptación en la *blockchain*. Por ejemplo, una transferencia de un activo. En el ejemplo del CI de la Figura 1, el traspaso de un automóvil (función transferir).
- *Libro (del inglés ledger: libro mayor de contabilidad):* es la cadena de bloques. Es la base de datos que contiene todas las transacciones, las cuales se agrupan en bloques de manera similar a la *blockchain* de Bitcoin.
- *Bloque:* es un conjunto de transacciones y otros datos adicionales. Es similar a un bloque de la *blockchain* de Bitcoin pero no tiene *nonce*.
- *Red:* es igual que en la *blockchain* de Bitcoin.
- *Nodo par u homólogo (del inglés peer):* es una aplicación que se ejecuta en Docker (Docker es una herramienta que facilita la ejecución de aplicaciones en Linux [22]). Un par alberga uno o más libros (copias de estos, es decir, una o más *blockchains*). Un par está pendiente de determinados procesos, por ejemplo, cuando se van a añadir bloques a un libro. Hay dos tipos esenciales de par, el avalador y el confirmador:
  - Un par avalador ejecuta una función del CI, lo que genera un aval. Más adelante se detalla este proceso.
  - Un par confirmador añade los bloques al libro.
- *Nodo ordenador:* es una aplicación que se ejecuta en Docker. Un conjunto de nodos ordenadores conforma el servicio de ordenamiento, el cual se explica más abajo.
- *Organización:* es un miembro de la red que gestiona una o varias *blockchains*. Posee uno o varios nodos pares pero un par pertenece a una sola organización. Una *blockchain* suele ser gestionada por varias organizaciones [23]. Por ejemplo, sean dos organizaciones Org1 y Org2. Cada organización interactúa con la *blockchain* por medio de una aplicación. Supóngase que la Org1 desea transferir el Auto1 (un automóvil) a la Org2 y esta última desea transferir el Auto2 (otro automóvil) a la Org1. Para ello, la Org1 invoca la función transferir (Auto1, vendedor, comprador), véase la Figura 1, donde vendedor = Org1 y comprador = Org2, y la Org2 invoca la función transferir(Auto2, vendedor, comprador), donde vendedor = Org2 y comprador = Org1.
- *Consorcio:* es un conjunto de organizaciones.
- *Canal:* es un mecanismo mediante el cual se comunican los miembros de un consorcio, es decir, las organizaciones. Puede haber varios canales en una red. Es útil cuando hay diferentes libros: cada canal tiene un único libro; de esta forma, se puede aislar a las organizaciones haciendo que estas solo tengan acceso a determinados libros.

En la Figura 4 se pueden ver los distintos participantes en una *blockchain* de Hyperledger Fabric. Allí hay tres organizaciones (Org1, Org2 y Org3) y ocho pares (P1 a P8) en una red llamada N. El canal C conecta cinco de estos pares: P1, P3, P5, P7 y P8. Los otros pares P2, P4 y P6, no están conectados a este canal, pero están unidos al menos a otro canal (un par puede estar conectado a varios canales). En la Figura 5 se presenta otro esquema de una *blockchain* de Hyperledger Fabric. Allí, hay tres organizaciones Org1, Org2 y Org3; y dos canales Canal_Todos y Canal_Org1_Org2. Cada organización tiene asociado un par: Par0.Org1, Par0.Org2 y Par0.Org3. Nótese que un par puede estar asociado con varios canales como el Par0.Org1 y el Par0.Org2. Cada par tiene una copia del libro y un CI por cada canal al que tiene acceso

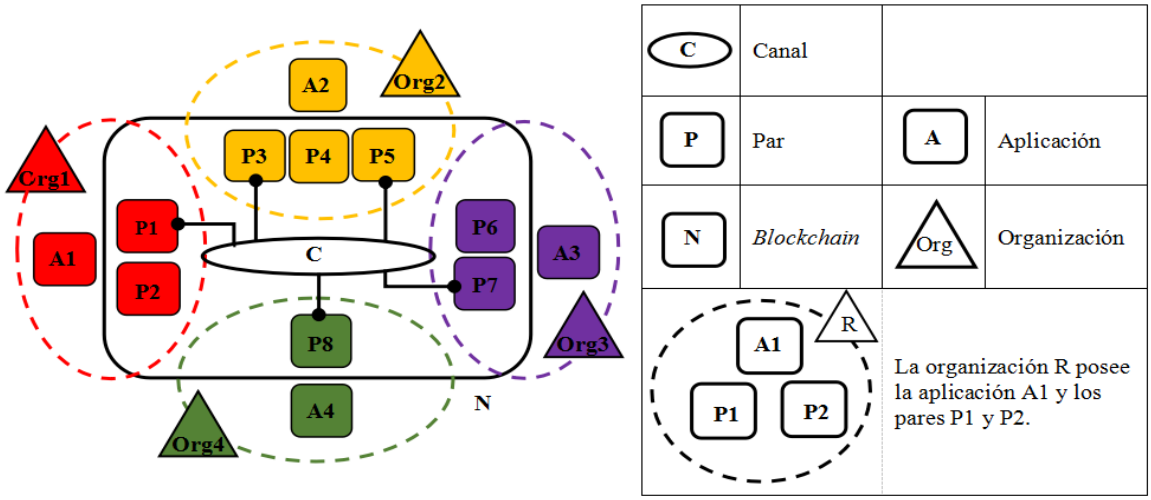

Figura. 4. *Esquema de una blockchain de Hyperledger Fabric. Fuente: adaptado a partir de* [23]

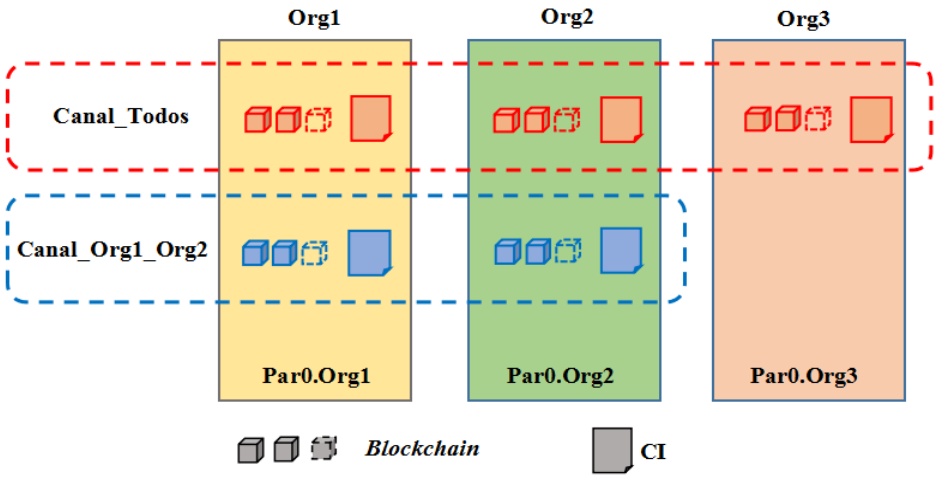

Figura 5. *Otro esquema de una blockchain de Hyperledger Fabric. Fuente: adaptado a partir de* [24].

- *Política de aval:* es una política que establece cuáles o cuantas organizaciones se requieren para avalar una transacción. Por ejemplo, una política puede establecer que al menos un par avalador de cada organización debe avalar la transacción para que este pase a la fase de confirmación.
- *Servicio de ordenamiento:* es un servicio de s*oftware* que arma los bloques. Este servicio recibe las transacciones avaladas, es decir, los avales hechos por los pares avaladores, las ordena, ver explicación más abajo, las agrupa en un bloque y lo envía a los pares confirmadores para que estos lo añadan al libro. Nótese que en la *blockchain* de Bitcoin el orden de las transacciones queda determinado por el minero: el minero armó el bloque y; por lo tanto, les dio un orden a las transacciones en este.
- *Política de consenso:* es una política que establece las reglas para llegar a un acuerdo sobre cuál será el siguiente bloque que se

añadirá al libro y sobre el orden de las transacciones en cada bloque.

4.2. Flujo de una transacción

A grandes rasgos, una *blockchain* en Hyperledger funciona como se describe a continuación [23], [25] y [26]:

1. Un usuario mediante una aplicación emite una transacción (invocación de una función de un CI) y esta es enviada a los pares avaladores.

2. Cada par avalador ejecuta la función del CI, pero sin actualizar su libro, y devuelve un aval a la aplicación del paso 1. Específicamente, cada par avalador ejecuta la función del CI y devuelve en el aval en una variable llamada *estadoModificado* los datos como *quedarían* actualizados por la función y en otra variable llamada *estadoActual* los datos *sin* actualizar. Así, el *estadoModificado* y el *estadoActual* conforman el aval, véase la Figura 6.

3. La aplicación del paso 1 recoge todos los avales de una transacción (enviados por los pares avaladores del paso 2) y los reúne en un paquete llamado *transacción avalada* y la envía al servicio de ordenamiento.

4. El servicio de ordenamiento establece el orden de todas las transacciones avaladas que recibe y las agrupa en bloques [25] y [26]. En este punto entra en juego la política de consenso: se llega a un acuerdo por parte de los nodos ordenadores sobre el orden de los bloques y de las transacciones en cada bloque. Por ejemplo, supóngase que dos transacciones avaladas (un retiro y una consignación) van a afectar a una misma cuenta. Debe entonces quedar determinado cuál de las dos se confirmará primero en el libro.

5. Los bloques del paso 4 son enviados a los pares confirmadores.

6. Cada par confirmador valida las transacciones de cada bloque (recibido del paso 5) y actualiza en su libro local los cambios generados por las transacciones avaladas válidas. Específicamente, cada par confirmador valida para cada transacción: i) la política de aval y ii) que no haya habido cambios en el estado del libro con respecto al estado actual que fue obtenido en los avales del paso 2, es decir, que los datos de la variable *estadoActual* coincidan con el estado actual del libro.

7. Si la transacción avalada es válida, entonces los cambios que aparecen en la variable *estadoModificado* son aplicados al libro local de cada par. Cada transacción avalada del bloque es etiquetada como válida o inválida y solo para las transacciones válidas se confirma el *estadoModificado*. Cada par confirmador añade el bloque a su libro. Finalmente, se emite un evento para notificar a la aplicación del paso 1 si la transacción fue válida o inválida. El proceso se muestra en la Figura 7.

5. Comparación de Bitcoin y Hyperledger Fabric

En la Tabla 3 se presenta un resumen de la comparación de las dos plataformas.

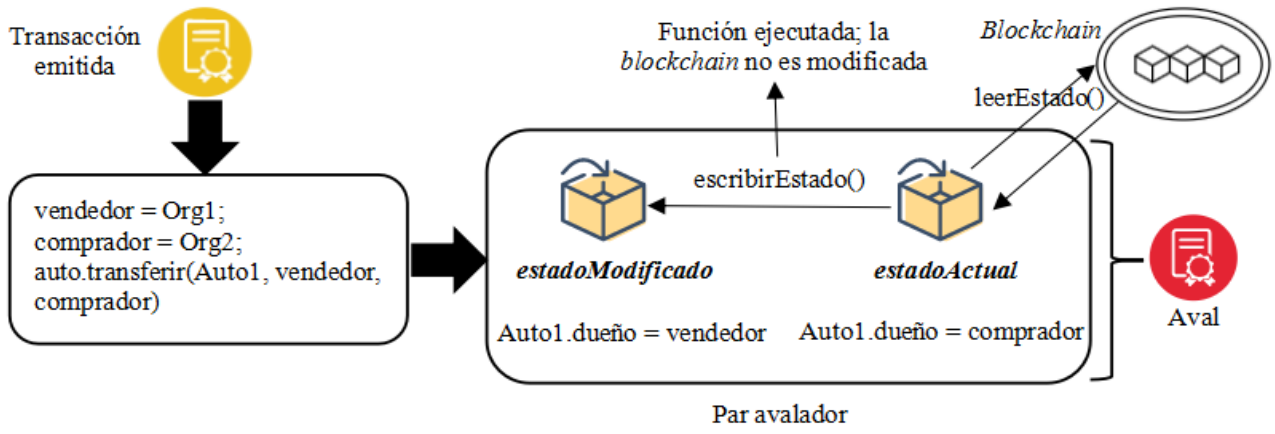

Figura 6. *Estructura de un aval. Fuente: elaboración propia.*

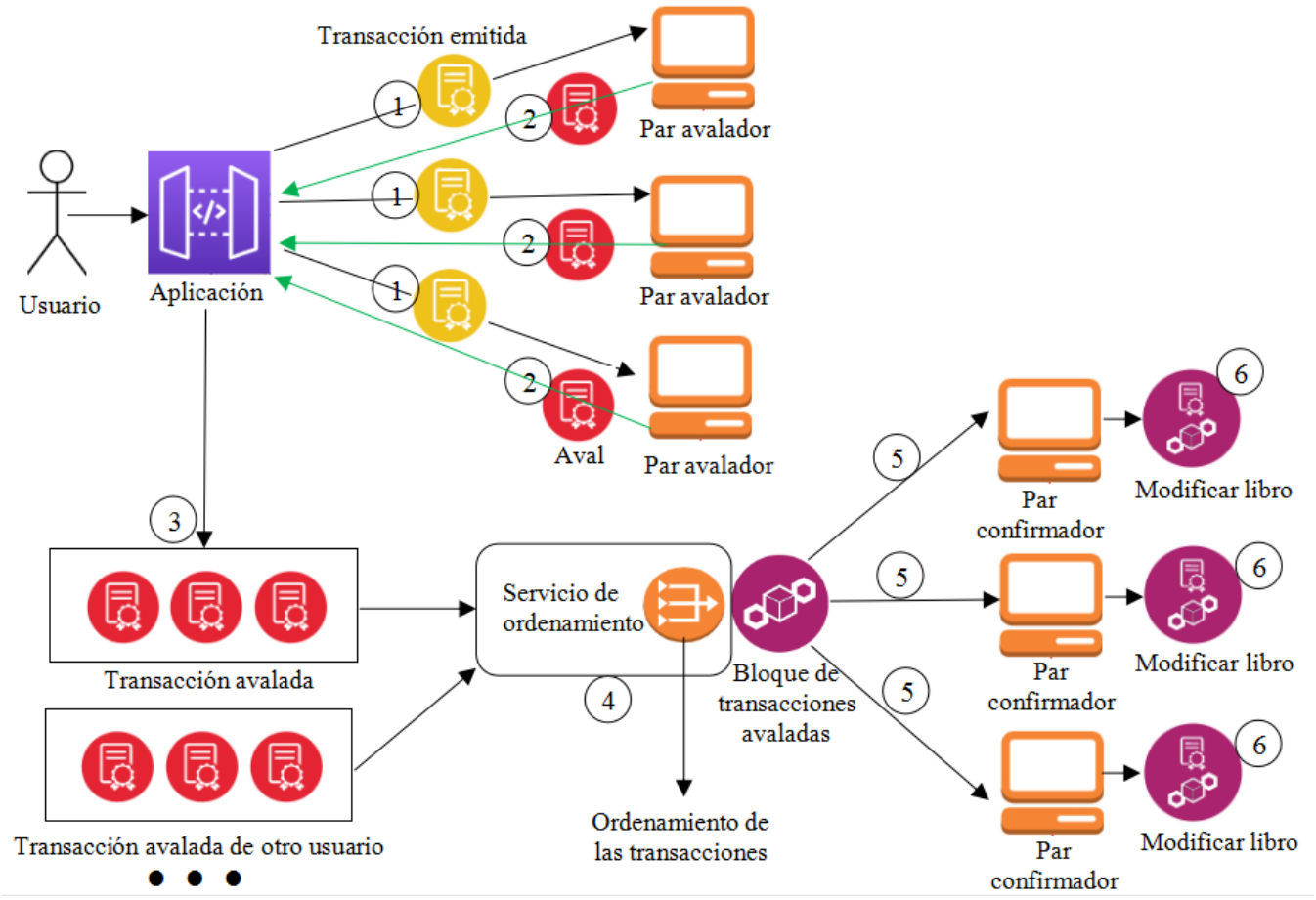

Figura 7. *Flujo de una transacción Hyperledger Fabric. Fuente: elaboración propia.*

Tabla 3. *Comparación de las dos plataformas.*

Aspecto	Bitcoin	Hyperledger Fabric
	Nodo	
Tipo	Es un computador de la red.	Es una aplicación que se ejecuta en Docker.
Función	Validar las transacciones y autorizar que se añada un bloque a la cadena. También puede actuar como un minero: arma un bloque e intenta resolver la PoW.	Según su función se tiene: a) Par avalador: avala las transacciones emitidas. b) Par confirmador: añade bloques al libro. c) Ordenador: arma los bloques y determina el orden de las transacciones en cada uno.
	Transacción	
Tipo	Transferencias de *bitcoins*	Pueden ser de cualquier tipo, es decir, dependen del negocio.
Zona de acopio (de las transacciones emitidas por los usuarios)	Fondo	No hay
Fases de ejecución	Ordenar y ejecutar [27]	Ejecutar, ordenar y validar [27]
Tiempo promedio de aprobación	30 minutos [28]	Depende del número de nodos y del protocolo de consenso.
	Bloque	
Tamaño	1.000.000 de *bytes* [29]	Configurable
Número promedio de transacciones	2700 [30]	Varía según el tamaño máximo que puede tener un bloque, el cual es configurable.
Responsable de su armado	Cada nodo selecciona a partir del fondo las transacciones para armar un bloque.	Existe un servicio de ordenamiento que arma los bloques y determina el orden de las transacciones en cada uno.
Latencia	10 min [10]	100 ms [10]
	Consenso	
Política	Existe una política para establecer el porcentaje de nodos requerido para aprobar un bloque.	Existen políticas para establecer: a) Cuáles o cuántas organizaciones se requieren para avalar una transacción emitida.

		b) Cuál es el siguiente bloque a añadir al libro y el orden de las transacciones en cada bloque.
Protocolo	PoW	Soporta diferentes protocolos, por ejemplo, PBFT y Apache Kafka. La elección del protocolo depende de tres factores: velocidad, escalabilidad y finalidad (es decir, confiabilidad de que un bloque una vez comprometido no será revocado) [31].
	Seguridad	
Acceso	Público	Privado
Autenticación	Por medio de una billetera virtual los usuarios se autentican para hacer transacciones.	Existe un proveedor de servicios de membresía [23] que se encarga de generar certificados para la autenticación de los miembros de la red.
Inmutabilidad de los datos	Alta. Es casi imposible modificar los datos debido al número de nodos (miles) que tienen copia de la cadena.	Baja. Al ser privado tiende a haber pocas (decenas) copias del libro; por lo tanto, es más vulnerable.
	Otros	
CIs	Poco personalizables: están enfocados a las transferencias de *bitcoins*. Se programan en Bitcoin Script.	Personalizables: se pueden crear según las necesidades del negocio. Se pueden programar en Go, Java y JavaScript.

Fuente: elaboración propia

6. Conclusiones y trabajos futuros

En este artículo se presentaron varios trabajos que comparan plataformas de *blockchain*. Destaca la variedad de criterios usados para su evaluación. En general, se evalúan principalmente aspectos de rendimiento, protocolos de consenso y seguridad.

También se compararon dos plataformas de *blockchain*: Bitcoin y Hyperledger Fabric. La diferencia principal es el enfoque: la *blockchain* de Bitcoin es pública y está orientada a transferencias de *bitcoins*. Por su parte, Hyperledger Fabric es una plataforma de *blockchains* privadas de propósito general ya que las transacciones pueden ser de cualquier tipo. Se enfatizó en el flujo de una transacción en las dos plataformas para observar sus similitudes y diferencias a nivel de componentes y las funciones de cada uno.

Como trabajo futuro se espera comparar diferentes plataformas de *blockchain* considerando el aspecto forense. Es decir, analizar los datos generados por cada plataforma y descubrir, por ejemplo, en cuál de ellas hay más indicios de fraudes, de lavado de activos, de transacciones periódicas sospechosas entre dos usuarios, entre otras. Lo anterior requiere el diseño de herramientas y de lenguajes de consulta

especializados que faciliten la identificación de estos aspectos. Finalmente, la llegada de Ethereum 2.0, planeada para 2022, plantea una serie de innovaciones que deberán ser analizadas.

7. Declaración de financiación

Los autores declaran que no se recibió financiación para la realización de este artículo de ninguna institución.

8. Referencias

(1) Bruyn AS. Blockchain an introduction. Research paper. [consultado 15 ene 2021] 2017.

(2) Casado-Vara R, Corchado JM. Blockchain for democratic voting: how blockchain could cast of voter fraud. Oriental Journal of Computer Science and Technology. 2018;11(1):1-3. http://dx.doi.org/10.13005/ojcst11.01.01.

(3) Chang J, Katehakis MN, Melamed B, Shi JJ. Blockchain design for supply chain management. SSRN Electronic Journal. 2018. Disponible en: https://dx.doi.org/10.2139/ssrn.3295440.

(4) Uzair MM, Karim E, Sultan P, Ahmed SS. The impact of blockchain technology on the real estate sector using smart contracts. Munich Personal RePEc Archive. MPRA Paper, 2018. Disponible en: https://mpra.ub.uni-muenchen.de/id/eprint/88934.

(5) Teja AM, Alekhyam S, Jeevanbabu D. Online facility of flight ticket booking using blockchain. Journal of Engineering Research and Application. 2019;9(3):79-83. Disponible en: https://www.ijera.com/papers/vol9no3/Series-5/M0903057983.pdf.

(6) Bank for Interantional Settlements (BIS), Committee on Payments and Market Infrastructures. CPMI Papers, Nro. 157. Distributed ledger technology in payment, clearing and settlement. Basel, Suiza: BIS; 2017.

(7) Haber S, Stornetta WS. How to timestamp a digital document. En: Menezes AJ, Vanstone SA, editors. Advances in Cryptology-CRYPTO' 90 CRYPTO 1990 Lecture Notes in Computer Science, vol 537. Berlin, Heidelberg: Springer; 1991. p. 437–55. Disponible en: https://doi.org/10.1007/3-540-38424-3_32.

(8) Dinh TTA. Blockbench: A framework for analyzing private blockchains. En: SIGMOD '17: Proceedings of the 2017 ACM International Conference on Management of Data. Chicago; 2017. p. 1085–1100. https://doi.org/10.1145/3035918.3064033.

(9) Zahnentferner J. Chimeric ledgers: translating and unifying UTXO-based and account-based cryptocurrencies. Cryptology ePrint Archive. Report 2018/262. 2018. Disponible en: https://eprint.iacr.org/2018/262.pdf.

(10) Salimitari M, Chatterjee M. A survey on consensus protocols in blockchain for IoT networks. Networking and Internet Architecture. 2019;arXiv:1809.05613. 2018. Disponible en: https://arxiv.org/abs/1809.05613.

(11) Valenta M, Sandner P. Comparison of Ethereum, Hyperledger Fabric and Corda. Frankfurt School, Blockchain Center. Informe Técnico. 2017. Disponible en: http://explore-

ip.com/2017_Comparison-of-Ethereum-Hyperledger-Corda.pdf.

(12) Macdonald M, Liu-Thorrold L, Julien R. The blockchain: a comparison of platforms and their uses beyond bitcoin. COMS4507-Advanced Computer and Network Security. University of Queensland; 2017. http://dx.doi.org/10.13140/RG.2.2.23274.52164.

(13) Maharjan PS. Performance analysis of blockchain platforms. [Tesis de maestría]. Las Vegas: University of Nevada; 2018. http://dx.doi.org/10.34917/14139888.

(14) Nasir Q, Qasse IA, Talib MA, Nassif AB. Performance analysis of Hyperledger Fabric platforms. Security and Communication Networks.2018:1-14. https://doi.org/10.1155/2018/3976093.

(15) Veskus K. Ethereum versus Fabric–A comparative analysis [Tesis de pregrado]. Tartu: University of Tartu; 2018. Disponible en: https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=61989&year=2018.

(16) Voulgaris S, Fotiou N, Siris VA, Polyzos GC, Jaatinen M, Oikonomidis Y. Blockchain technology for intelligent environments. Future Internet. 2019;11(10):213. http://dx.doi.org/10.3390/fi11100213.

(17) Shreves R. Block by block a comparative analysis of the leading distributed ledgers [Internet]. Mercy Corps. 2018 [consultado 10 oct 2020]. Disponible en: https://www.mercycorps.org/research-resources/block-block.

(18) Scherer M. Performance and scalability of blockchain networks and smart contracts [Tesis de doctorado]. Umeå: Umeå University, 2017. Disponible en: http://urn.kb.se/resolve?urn=urn%3Anbn%3Ase%3Aumu%3Adiva-136470.

(19) Navarro BY. Blockchain y sus aplicaciones. Universidad Católica Nuestra Señora de La Asunción. Working paper. 2017. Disponible en: https://pdfslide.tips/documents/blockchain-y-sus-aplicaciones-benjamin-yahari-navarro-universidad-cat-olica-nuestra.html

(20) Gueron S, Johnson S, Walker J. SHA-512/256. En: 8th International Conference on Information Technology: New Generations. Las Vegas; 2011. p.354-358. http://dx.doi.org/10.1109/ITNG.2011.69.

(21) Blockchain Institute of Technology. What is the Bitcoin transaction pool? [Video de internet]. Youtube. 9 ene 2018 [citado 9 nov 2020]. Disponible en: https://www.youtube.com/watch?v=m6Vaefx69_Q.

(22) Red Hat. ¿Qué es Docker? [Internet]. Red Hat. 5 abril 2019 [consultado 10 oct 2020]. Disponible en: https://www.redhat.com/es/topics/containers/what-is-docker

(23) Hyperledger. Hyperledger a blockchain platform for the enterprise [Internet]. Hyperledger. 29 sep 2019 [consultado 3 ene 2021]. Disponible en: https://hyperledger-fabric.readthedocs.io/en/release-1.4.

(24) Tam K. Demo of three-node two-channel setup in Hyperledger Fabric [Internet]. KC Tam. 7 abr 2019 [consultado 8 ene 2021]. Disponible en: https://kctheservant.medium.com/demo-

of-three-node-two-channel-setup-in-hyperledger-fabric-54ba8a9c461f.

(25) Brandenburger M, Cachin C, Kapitza R. Blockchain and trusted computing: problems, pitfalls, and a solution for Hyperledger Fabric. 2018;arXiv:1805.08541. 2018. Disponible en: https://arxiv.org/abs/1805.08541.

(26) Androulaki E, Barger A, Bortnikov V, Cachin C, Christidis K, et al. Hyperledger Fabric: a distributed operating system for permissioned blockchains. En: EuroSys '18: Proceedings of the Thirteenth EuroSys Conference. Nueva York; 2018. p. 1-15. https://doi.org/10.1145/3190508.3190538.

(27) Nguyen TSL, Jourjon G, Potop-Butucaru M, Thai K. Impact of network delays on Hyperledger Fabric. 2019;arXiv:1903.08856. 2019. Disponible en: https://arxiv.org/abs/1903.08856.

(28) Buchko S. How long do Bitcoin transactions take? [Internet] Coincentral. 12 dic 2017 [consultado 8 ago 2020]. Disponible en: https://coincentral.com/how-long-do-bitcoin-transfers-take.

(29) BTC. Block Size [Internet]. BTC.com. 11 may 2016 [consultado 17 sep 2020]. Disponible en: https://btc.com/stats/block-size.

(30) Moos M. Bitcoin transactions per block at all-time highs [Internet]. Cryptoslate. 8 abr 2019 [consultado 3 ene 2021]. Disponible en: https://cryptoslate.com/bitcoin-transactions-per-block-at-all-time-highs.

(31) Gauba A. Finality in blockchain consensus [Internet]. Medium. 30 Ago 2018 [consultado 15 ene 2021]. Disponible en: https://medium.com/mechanism-labs/finality-in-blockchain-consensus-d1f83c120a9a.