Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

ESTUDIO DE BLOCKCHAIN DESDE LA PERSPECTIVA DE ESTRUCTURAS DE DATOS

Josué David Itzep Salvador
Asesorado por el Ing. Carlos Gustavo Alonzo

Guatemala, mayo de 2019

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESTUDIO DE BLOCKCHAIN DESDE LA PERSPECTIVA DE ESTRUCTURAS DE DATOS

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

JOSUÉ DAVID ITZEP SALVADOR
ASESORADO POR EL ING. CARLOS GUSTAVO ALONZO

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, MAYO DE 2019

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martinez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Luis Diego Aguilar Ralón
VOCAL V	Br. Christian Daniel Estrada Santizo
SECRETARIA	Inga. Lesbia Magalí Herrera López

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. Luis Fernando Espino Barrios
EXAMINADOR	Ing. Sergio Arnaldo Méndez Aguilar
EXAMINADOR	Ing. William Estuardo Escobar Argueta
SECRETARIA	Inga. Lesbia Magalí Herrera López

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

ESTUDIO DE BLOCKCHAIN DESDE LA PERSPECTIVA DE ESTRUCTURAS DE DATOS

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 5 de septiembre de 2018.

Josué David Itzep Salvador

Guatemala 02 de Febrero de 2019.

Ing. Marlon Antonio Pérez Türk
Director de Escuela
Escuela de Ingeniería en Ciencias y Sistemas

Por este medio yo Carlos Gustavo Alonzo hago constar que el estudiante Josué David Itzep Salvador que se identifica con CUI No. 2987 94136 0101 y código estudiantil No. 2013 18613 ha concluido el trabajo de graduación que lleva por título "Estudio de Blockchain desde la perspectiva de Estructuras de Datos" bajo mi asesoría y expreso mi aprobación del mismo.

Sin otro particular, me es grato suscribirme.

F:_____

Ing. Carlos Gustavo Alonzo
Col. 6358
carlosalonzo795@hotmail.com

Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 28 de marzo de 2019

Ingeniero
Marlon Antonio Pérez Türk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **JOSUÉ DAVID ITZEP SALVADOR** con carné **201318613 y CUI 2987 94136 0101** titulado **ESTUDIO DE BLOCKCHAIN DESDE LA PERSPECTIVA DE ESTRUCTURAS DE DATOS** y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo aprobado.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación

ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERÌA
ESCUELA DE INGENIERÍA EN
CIENCIAS Y SISTEMAS
TEL: 24767644

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación ***"ESTUDIO DE BLOCKCHAIN DESDE LA PERSPECTIVA DE ESTRUCTURAS DE DATOS",*** *realizado por el estudiante, JOSUÉ DAVID ITZEP SALVADOR aprueba el presente trabajo y solicita la autorización del mismo.*

"ID Y ENSEÑAD A TODOS"

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DIRECCION DE INGENIERIA EN CIENCIAS Y SISTEMAS

Ing. Marlon Antonio Pérez Türk
Director
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 15 de mayo de 2019

Universidad de San Carlos
de Guatemala

Facultad de Ingeniería
Decanato

DTG. 236.2019

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **ESTUDIO DE BLOCKCHAIN DESDE LA PERSPECTIVA DE ESTRUCTURAS DE DATOS,** presentado por el estudiante universitario: **Josué David Itzep Salvador,** y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

Ing. Pedro Antonio Aguilar Polanco
Decano

Guatemala, mayo de 2019

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DECANO
FACULTAD DE INGENIERIA

/gdech

ACTO QUE DEDICO A:

Dios	Por darme la fortaleza mental, disciplina y responsabilidad necesaria para alcanzar mis objetivos.
Mis padres	Letty Salvador, por ayudarme a conseguir mis metas y objetivos a lo largo de los años, ser la persona que me motiva a seguir adelante en los momentos en que no encuentro las fuerzas necesarias, por esforzarse de sobremanera para brindar lo necesario a mi hermano y a mí y por amarme de una manera incondicional; y Santos Itzep por mostrar su apoyo a lo largo de los años, por mostrarme la realidad de la vida a través de anécdotas y experiencias, por transmitir conocimiento y por ser un ejemplo de superación y lucha.
Mi abuela	Por ser una gran influencia en mi vida, haber ayudado en mi crianza, mantener siempre un espíritu vivaz y mostrar su afecto incondicional a mi hermano y a mí.
Mi hermano	Por ser una de las mayores influencias en mi vida.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por abrirme las puertas para mi desarrollo como profesional, por mostrarme la realidad que se vive en Guatemala y permitirme sentir orgulloso de pertenecer a esta casa de estudios.
Facultad de Ingeniería	Por brindarme las herramientas y el conocimiento necesario para mi carrera profesional y ser parte en mi formación como persona.
Mis amigos de la Facultad	Por haber sido una fuente de apoyo, por brindar un ambiente agradable y permitirme distraer de las dificultades encontradas a lo largo de la carrera y aportar conocimiento a mi persona.
Mis catedráticos	Por haber compartido sus conocimientos y experiencias y formar parte de mi vida personal y profesional.
Ing. Carlos Gustavo Alonzo	Por mostrarme su apoyo tanto en el periodo de práctica final como en este trabajo de investigación y por compartir y abrir las puertas a nuevos conocimientos.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

FIGURAS

TABLAS

GLOSARIO

Algoritmo	Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.
Altcoin	Variante creada a partir de una criptomoneda.
Blockchain	Tecnología objeto de estudio de este trabajo de graduación, que permite la transparencia, seguridad e inmutabilidad de la información.
Bitcoin	Criptomoneda descentralizada.
Complejidad algorítmica	Cantidad de recursos utilizados por el algoritmo.
Criptología	Estudio de los sistemas, las claves y los lenguajes ocultos o secretos.
Criptomoneda	Activo digital, utilizado como medio de intercambio de bienes, que hace uso de la criptografía para asegurar las transacciones y transferencias de bienes.
Didáctica	Arte de enseñar.
Disrupción	Rotura o interrupción brusca.

Distributed ledger technology	Tecnología de tipo distribuida, que consiste en un conjunto de miembros que comparten información entre sí a través de distintos algoritmos.
Ether	Forma de pago en la red de Ethereum.
Ethereum	Plataforma descentralizada que ejecuta contratos inteligentes.
Framework	Conjunto de herramientas de software utilizado para el desarrollo de software.
Fundacional	Perteneciente o relativo a la fundación.
Ganache	Blockchain personal para el desarrollo en Ethereum.
Hash	Término utilizado para denotar una transformación y mezcla de valores que no necesariamente poseen un sentido lógico.
Hyperledger	Esfuerzo colaborativo de código abierto creado para el desarrollo de tecnologías Blockchain.
Inmutable	No mudable, que no puede ni se puede cambiar.
Java	Lenguaje de programación multipropósito orientado a objetos.

Java JDK	Conjunto de herramientas de desarrollo para el lenguaje de programación Java.
Javascript	Lenguaje de programación interpretado.
MedChain	Aplicación de la tecnología Blockchain en la industria de la salud.
Merkle tree	Estructura de datos no lineal compuesta de valores criptográficos.
Node.js	Ambiente de ejecución para el lenguaje Javascript
Nonce	Valor perteneciente a la estructura de Blockchain, utilizada para aumentar el nivel de seguridad.
npm	Administrador de paquetes para el lenguaje de programación Javascript.
Openssl	Librería de uso general de criptografía.
Open-source	Software en el que el código es liberado bajo distintas licencias para su uso.
Paradigma	Teoría o conjunto de teorías cuyo núcleo central se acepta sin cuestionar y que suministra la base y modelo para resolver problemas y avanzar en el conocimiento.

Peer-to-Peer	Tipo de arquitectura de red, que conecta a cada uno de los miembros de la red entre sí.
PEPS	Método que se basa en que el primero que entra a un sistema es el primero que sale de él.
Replicar	Repetir lo que se ha dicho.
Solidity	Lenguaje de programación de alto nivel orientado a contratos.
Struct	Tipo de dato de programación que permite crear una estructura de mayor complejidad.
Timestamp	Marca de tiempo digital, que posee valores como fecha y hora.
TCP	Protocolo de comunicación a través de una red de computadoras.
Truffle	Ambiente de desarrollo para aplicaciones Ethereum.
UEPS	Método que se basa en que el ultimo en entrar es el primero en salir.

RESUMEN

Las estructuras de datos han sido objeto de estudio de la computación desde los inicios de esta, y el manejo de la información ha sido sin lugar a duda su principal objetivo. Blockchain es una tecnología que desde sus inicios causó una revolución en la computación dando paso así a nuevos paradigmas, soluciones, y a lo largo de su ciclo de vida ha ido evolucionando y su alcance se ha ido expandiendo a distintos mercados y nuevos casos de uso han sido descubiertos, potenciando así los beneficios que esta tecnología ofrece.

La relación que existe entre la tecnología Blockchain y las estructuras de datos es muy estrecha; y está formada por distintas estructuras, por lo que es posible establecer la tecnología como objeto de estudio de las estructuras de datos.

Se ha desarrollado un marco de trabajo orientado a la didáctica de las estructuras de datos, así como distintas herramientas didácticas y tecnologías que ejemplifican sus bases teórica y técnica.

OBJETIVOS

General

Presentar los lineamientos generales de la tecnología Blockchain, incorporado a la enseñanza de las estructuras de datos.

Específicos

1. Describir la relación que existe entre distintos componentes de la tecnología Blockchain con el estudio de las estructuras de datos.

2. Examinar distintas aplicaciones de Blockchain que permita crear una idea de la evolución de la tecnología.

3. Determinar una secuencia lógica para el proceso de enseñanza de la tecnología Blockchain.

4. Comparar distintos tipos de herramientas Blockchain para determinar la mejor alternativa en la docencia.

INTRODUCCIÓN

Las estructuras de datos representan, sin lugar a duda, uno de los fundamentos de las ciencias de la computación; su estudio es de gran importancia en la computación y sus aplicaciones pueden verse en distintas ramas de esta ciencia; Blockchain es una tecnología que hace uso de distintas estructuras de datos y hace uso de distintos algoritmos pertenecientes a esta rama de la computación.

Se ha tomado esta tecnología como objeto de estudio por las distintas aplicaciones de las estructuras de datos que utiliza, el potencial que este a lo largo de los años ha mostrado y los beneficios que proporciona.

Este trabajo describe como punto inicial el estudio de las estructuras de datos; los distintos componentes de la tecnología Blockchain; las distintas formas en que se puede clasificar y su funcionamiento general. Además, se establecen distintas herramientas didácticas para el estudio de esta tecnología como estructura de datos.

1. ESTRUCTURAS DE DATOS

Desde el inicio de la computación en el año 1642, con la primera calculadora de Blaise Pascal, hasta la actualidad con las supercomputadoras, las cuales son capaces de procesar más de un billón de operaciones por segundo con facilidad, el objetivo de estas a lo largo del tiempo ha sido sin lugar a duda el manejo de la información.

Las estructuras de datos son conjuntos de datos atómicos o compuestos, que delimitan un dominio de valores con características similares y al mismo tiempo son delimitadas por dichas características, las cuales también poseen una serie de reglas por medio de las cuales se asocian o relacionan entre sí.

Así como la computación, el objetivo de las estructuras de datos es manipular de una forma correcta la información, buscando la eficacia y eficiencia en dicho manejo, así como su correcta organización; para lo cual se hace uso de distintos algoritmos y formas de organización que se encargan de cumplir con dichos objetivos. Entre las formas de organización se pueden encontrar dos grandes categorías:

- Estructuras de datos lineales
- Estructuras de datos no lineales

1.1. Tipos de estructuras de datos

Las estructuras de datos se dividen en dos grandes ramas: lineales y no lineales; la diferencia entre estas es, como se verá a continuación, la manera en que son recorridas y organizadas.

1.1.1. Lineales

La característica principal de este tipo de estructuras, es el hecho de que se organizan de manera secuencial a nivel lógico, aunque a nivel físico pueden no estarlo, esta secuencialidad significa que los elementos se encuentran uno seguido de otro.

Otra característica de las estructuras lineales es de que los recorridos sobre ellas se realizan consecutivamente uno detrás del otro visitando a cada uno de los elementos de la estructura uno por uno; además, se caracterizan por el hecho de que dicho recorrido puede ser realizado en una sola ejecución. Así mismo, estas presentan un solo nivel de profundidad y suelen ser de fácil implementación. Entre las estructuras de datos lineales están:

- Arreglo
- Lista enlazada
- Pila
- Cola

1.1.2. No lineales

En contraste a las estructuras lineales, la organización de estas no es secuencial y normalmente es organizada con base en algoritmos de

ordenamiento; además, los elementos de estas estructuras pueden estar relacionados a uno, dos o más elementos del mismo tipo; dicha relación es de tipo especial y generalmente constituyen nuevos niveles de profundidad. Por lo general, estas suelen tener un grado mayor de complejidad en su implementación, aunque su tiempo de respuesta suele ser más rápido; comparado con las estructuras lineales. Entre las estructuras de datos no lineales están:

- Árboles
- Grafos

1.2. Tablas de dispersión

Las tablas de dispersión, o tablas *hash*, son estructuras compuestas de estructuras lineales y en algunas ocasiones de estructuras no lineales que asocian un valor denominado llave a un índice dentro de la estructura. El objetivo principal de esta estructura es tener una eficiencia óptima, la cual es lograda mediante la asociación de un valor representativo a la información almacenada en la estructura, a través de las denominadas funciones *hash*.

1.2.1. Funciones *hash*

Estas funciones cumplen con el objetivo de transformar una llave en un valor equivalente a una posición dentro de la tabla de dispersión; idealmente estas funciones generarán una posición única, para cualquier valor de llave que se presente; sin embargo, ya sea que se genere un valor único o un valor para los índices repetido, de igual manera se alcanzará una gran eficiencia. Estas funciones normalmente consisten en operaciones aritméticas y algoritmos con

una baja necesidad de cómputo. Existen distintos tipos de funciones *hash*, entre las cuales están:

- Funciones triviales
- Funciones multiplicativas
- Funciones criptográficas
- Funciones perfectas

De las cuales, debido al alcance de este documento, interesan las funciones perfectas y criptográficas; las funciones perfectas como se menciona anteriormente, y como bien lo describe su nombre, son aquellas que permiten transformar cada una de las llaves a un resultado único e ir repetido, por lo que pueden ser llamadas también funciones uno a uno.

1.3. Criptografía

La criptografía es un campo de la criptología, que desde la antigüedad se ha utilizado para brindar cierto grado de seguridad y confianza a las personas sobre distintos tipos de información, para lo cual hace uso de distintos algoritmos; además, se combina con algunos conceptos y operaciones matemáticas para cumplir con su objetivo, el cual básicamente es el de ocultar el significado de los mensajes.

Sin lugar a duda la criptografía es una herramienta de gran ayuda para la protección de la información debido a la necesidad que existe de poder enviar la información y compartirla con otras personas a través de distintos medios. Para lo cual han surgido distintas formas de protección de la misma, con distintos comportamientos y técnicas que se acoplan a las distintas necesidades.

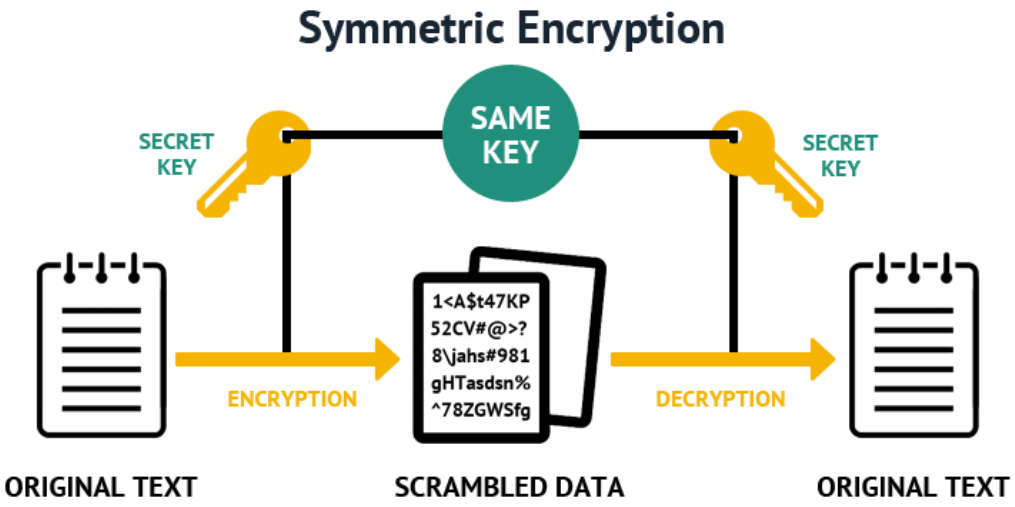
Cuando se maneja información de alta importancia, es obvio que se desea mantener el acceso a ella en un círculo de confianza limitado y que aunque personas ajenas a este círculo de confianza puedan de alguna u otra forma acceder a ella, no les sea posible su interpretación, para lo cual la criptografía se presta en distintas formas, entre las cuales en relación a este documento, se pueden mencionar las siguientes:

- Algoritmos simétricos
- Algoritmos asimétricos

1.3.1. Criptografía simétrica

También llamada criptografía de llave secreta, debido a que hace uso de una llave secreta para cifrar el mensaje o información deseada, para lo cual solo aquellos que tengan en su poder dicha llave secreta podrán realizar la labor de descifrar el mensaje o información. Aunque bastante efectiva en su labor de ocultar el mensaje presenta distintas desventajas, entre las cuales figura el hecho de que si una persona ajena obtiene de alguna forma dicha llave secreta, tiene en sus manos la capacidad de poder acceder a la información.

Figura 1. **Criptografía simétrica**

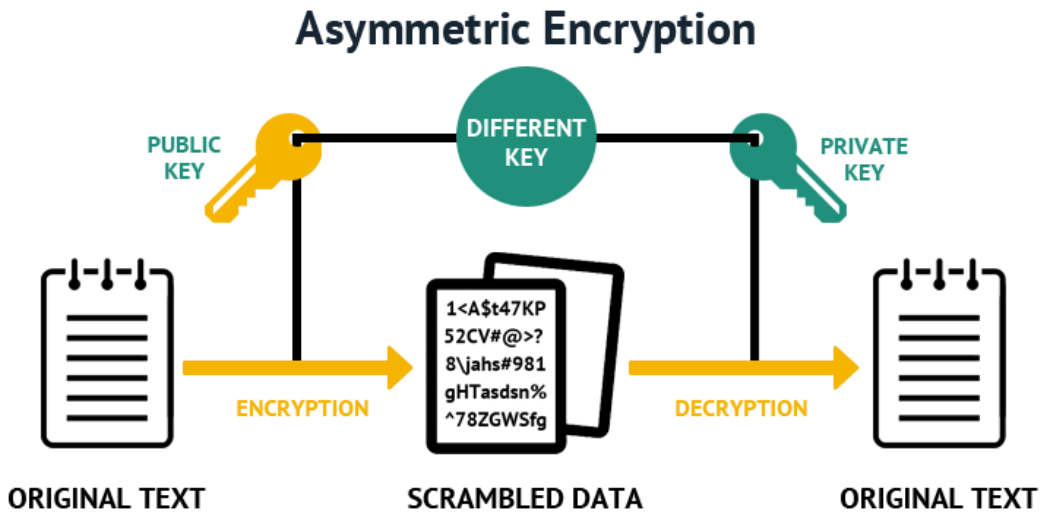


Fuente: SSL Shop. *Demystifying symmetric and asymetric methods of encryption*. www.cheapsslshop.com/blog/demystifying-symmetric-and-asymmetric-methods-of-encryption. Consulta: 5 de noviembre de 2018.

1.3.2. Criptografía asimétrica

Similar a la criptografía simétrica, esta es llamada de llave pública, se basa principalmente en la ejecución de operaciones matemáticas y consta así como la criptógrafa simétrica, tanto de una llave secreta o privada, como de una llave pública, ambas llaves complementarias entre sí. El funcionamiento general de este tipo de criptografía se basa en la distribución de la llave pública, por medio de la cual se puede cifrar un mensaje o información y el dueño único de la llave privada podrá tanto descifrar el mensaje como validar que dicho mensaje posea esa marca dejada por la llave pública, debido a la relación que existe entre ambas llaves.

Figura 2. **Criptografía asimétrica**



Fuente: SSL Shop. *Demystifying symmetric and asymetric methods of encryption.* www.cheapsslshop.com/blog/demystifying-symmetric-and-asymmetric-methods-of-encryption. Consulta: 5 de noviembre de 2018.

1.3.3. Funciones *hash* criptográficas

Las funciones *hash* a pesar de representar una herramienta utilizada en las tablas de dispersión no son limitadas únicamente a generar índices dentro de una de estas tablas, sino que su alcance y funcionalidad ha sido extendido debido a las distintas aplicaciones que han surgido. Las aplicaciones de las funciones *hash* en la criptografía son varias y normalmente consisten en la generación de pequeños segmentos de información obtenidos a partir de mensajes o información de tamaño arbitrario, dichos segmentos son llamados huellas del mensaje o valores *hash*; debido a la necesidad de tener un bajo nivel de cómputo, estas normalmente requieren que el valor *hash* generado tenga un tamaño corto y limitado, independientemente del tamaño de la

información a la que se le aplique la función. Las funciones *hash* constan de varias propiedades, las cuales deben poseer:

- Tamaño arbitrario del mensaje
- Longitud de salida fija
- Eficiencia
- Valores de salida únicos
- Resistente a las colisiones

Estas funciones y los valores que generan, normalmente, forman parte del proceso de cifrado y descifrado en la encriptación asimétrica, por lo que son llamadas también funciones *hash* criptográficas.

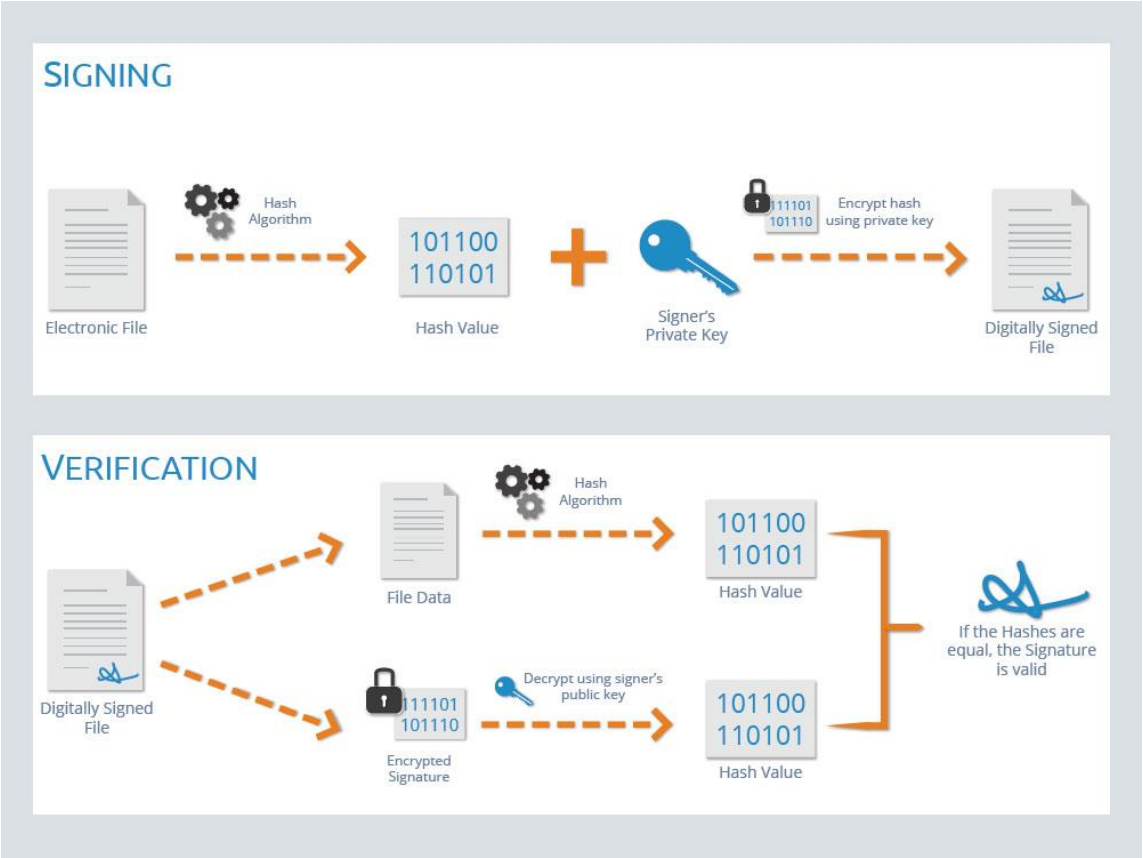
1.3.4. Firmas digitales

Las firmas digitales forman parte de la criptografía asimétrica y son una extensión de la misma, que pretenden disolver aún más los problemas de seguridad, en la cual se verifica la integridad y se valida que la persona que envía un mensaje sea aquella que lo creo.

Como la criptografía asimétrica, esta consta de llave pública y privada, la llave pública se distribuye y solo la persona que posea la llave privada puede generar una firma digital sobre la información que desee, cuando se envía la información se envía también la firma digital de dicha información, con lo cual las personas que reciban la información y la firma digital de la misma, podrán validar que de hecho la información haya sido generada por el poseedor de la llave privada.

Vale la pena resaltar, que distintos esquemas de firmas digitales, hacen uso de funciones *hash* criptográficas para la generación de llaves privadas y públicas, por la necesidad de la unicidad de las mismas y otros esquemas hacen uso de las funciones *hash*, para la generación de valores *hash* de la información el cual es el utilizado en la comparación y verificación de la información.

Figura 3. **Firma digital**



Fuente: Reva Solutions. *digital-signature-methodology.* www.revasolutions.com/expertise/process-management/digital-signatures/digital-signatures-methodology. Consulta: 5 de noviembre de 2018.

1.4. Aplicaciones

Como se ha mencionado anteriormente, el principal uso de las estructuras de datos es para almacenar y administrar la información; la principal aplicación de estas en las bases de datos; las bases de datos hacen uso principalmente de estructuras de datos no lineales para manejar su información tanto en memoria principal como en memoria secundaria, para tener un acceso eficiente a esta.

Implícitamente, el diseño de la información de las bases de datos consiste en abstracciones de la realidad en conjuntos de datos definidos. Otra aplicación de las estructuras de datos es en la inteligencia artificial; hace uso no solamente de árboles de información y árboles sintácticos, sino también de algoritmos de búsqueda utilizados para la resolución de problemas; es una de las ramas en las que las estructuras de datos y su estudio tienen un gran impacto.

Entre otras de las aplicaciones se pueden encontrar su uso en lenguajes de programación y compiladores, herramientas de todo estudioso de la computación e implícito en cualquier sistema de cómputo. Además de estas aplicaciones se pueden encontrar su uso en la criptografía y la seguridad; se ha mencionado que el uso de transformaciones y algoritmos de esta rama sirven para evitar la corrupción e intervención de terceros en nuestros sistemas; que permiten así asegurarse que la información esté segura de ataques o fallas en los sistemas. Estas son algunas de las muchas aplicaciones de las estructuras de datos en la computación que se relacionan con la tecnología objeto de estudio, Blockchain.

2. DISTRIBUTED LEDGER TECHNOLOGY

Distributed ledger technology (DLT), representa un tipo de tecnología, así como estructuras de datos, que como lo dice su nombre, hace alusión a los libros mayores utilizados en la contabilidad por cientos y cientos de años, y pretenden contener un registro de todas y cada una de las transacciones pertinentes a un sistema, de manera distribuida sobre todos y cada uno de los nodos pertenecientes al sistema conectados mediante una red, donde un nodo es un participante perteneciente a dicha red.

Entre las características principales de estos sistemas está la descentralización de la información, la cual a lo largo del tiempo ha sido objeto de creación de nuevos paradigmas y arquitecturas en el mundo de la computación.

Las DLT proponen tener una copia en cada uno de los nodos, de las distintas transacciones a las que es sometida el sistema, y que se pueda tener acceso a la información en cualquier momento; genera de esta manera una tolerancia a fallos alta, para lo cual en dado caso uno de los nodos no pueda ser accedido, o haya sido removido de la red por algún motivo, el sistema pueda seguir trabajando con normalidad.

Así mismo, esta descentralización y distribución de la información en el sistema también tiene como objeto no solo la ya mencionada anteriormente, tolerancia a fallos, sino también pretende remover a terceros, o intermediarios, que a lo largo del tiempo y en una gran variedad de transacciones comerciales o legales se han visto beneficiados en gran manera; por lo que no solo se

limitan a los involucrados en las transacciones; además, este tipo de sistemas es capaz de replicar las transacciones a todos los nodos de la red, ya sea en cuestión de minutos o incluso segundos, no sin antes haber validado y confirmado las nuevas transacciones con los distintos nodos en la red.

Es importante definir una DLT antes de adentrarse más en el capítulo, ya que es una pieza fundamental, en el objeto de estudio de este documento, la cual es Blockchain, que sin lugar a duda es en conjunto con las criptomonedas, la DLT con mayor impacto y funcionalidad.

2.1. Historia

Cuando se habla de Blockchain es inevitable hablar sobre Bitcoin, de hecho son términos que han sido intercambiados y confundidos entre sí, y no es de sorprenderse, ya que ambas tecnologías están asociadas muy íntimamente, y surgieron como tecnologías que en conjunto tenían un fin en común, el de crear un sistema de efectivo electrónico.

En el año 2008, surgió en una plataforma digital dedicada a la criptografía, un artículo llamado: *Bitcoin A Peer to Peer Electronic Cash System*, publicado bajo el nombre Satoshi Nakamoto cuya identidad a la actualidad no ha sido confirmada, dicho documento habla sobre Bitcoin una moneda digital, que hace uso de distintos protocolos y algoritmos enfocados a la validación de transacciones dentro de una red *peer-to-peer* o red entre pares; hace énfasis en la eliminación de intermediarios o terceros para la reducción de gastos, en las transacciones financieras, enfocándose también en la validez de la información y de consensos en los miembros de la red para aprobar y dar dicha validez a las transacciones.

Así mismo, este documento se enfocaba en la estructura de datos, que luego sería conocida como Blockchain. Aunque las criptomonedas ya habían sido creadas y utilizadas desde hace ya varios años, Bitcoin se presentaba como la primer criptomoneda descentralizada, y los beneficios que promovía fueron potenciadores para su éxito masivo. En el año 2009 Bitcoin fue liberada como software de código abierto y la minería en el Blockchain de Bitcoin empieza.

Alrededor del año 2014 fue que se empezó a ver los beneficios de Blockchain más allá de las criptomonedas y se inician nuevas investigaciones y aplicaciones de la tecnología. A lo largo de los años, desde su inicio hasta la actualidad, Bitcoin ha visto varias altas y bajas en su valor en el mercado oscilando desde $ 0,003 hasta $ 17 900,00. Entre los años 2013 al 2015 surgió la primer Blockchain pública, en la cual se introducían los contratos inteligentes o *smart contracts* a la tecnología. Luego de estos eventos Blockchain ha ido evolucionando e implementando nuevos protocolos de consenso, nuevos paradigmas y se ha ido expandiendo a nuevos mercados, ampliando así su alcance y encontrando nuevas inversiones en la tecnología.

2.2. Blockchain

Blockchain ha sido definida como “un libro mayor forjado por consenso entre pares, combinado con un sistema para ‘contratos inteligentes’ y otras tecnologías de asistencia”[1]. Es una *distributed ledger technology*, lo que implica que consiste de una red distribuida, sobre la cual se ejecutan distintas transacciones las cuales son validadas y agregadas al sistema por miembros de la red por medio de algoritmos de consenso; cada uno de los miembros de la

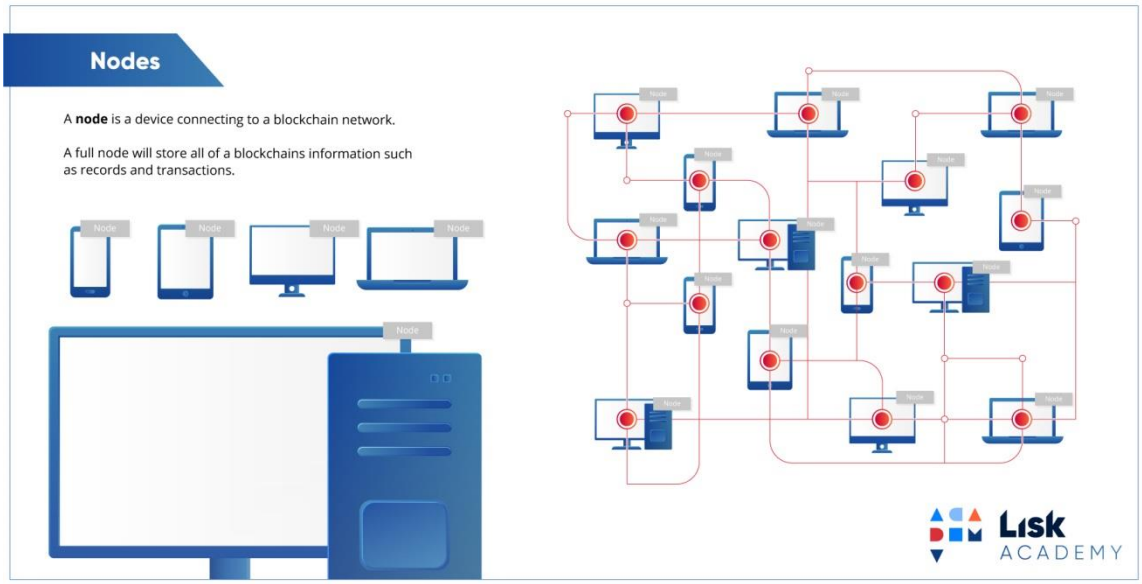
[1] Hyperledger. *About Hyperledger*. www.hyperledger.org/about. Consulta: 25 de octubre de 2018.

red puede emitir transacciones, así como interactuar e intercambiar bienes con otros miembros también pertenecientes a la red.

Blockchain cuenta también con una estructura de almacenamiento de la información la cual como hace alusión su nombre, consiste en una cadena de bloques, que son estructuras de datos con distintos atributos, la cual hace uso de funciones *hash* criptográficas y consiste en la agrupación de transacciones y generación de valores únicos e identificativos de las mismas, lo cual permite generar una secuencia en las transacciones, que aumenta la inmutabilidad de la información; esta estructura de almacenamiento se encuentra distribuida sobre todos los miembros de la red, por lo que aumenta su disponibilidad.

Otro aspecto de gran relevancia sobre esta tecnología es el uso de criptografía, que permite tanto la autenticación de las personas, como la autenticación de las transacciones realizadas sobre el sistema, permite así verificar que las transacciones sean realizadas de forma auténtica. Como se ha mencionado, Blockchain es a su vez una DLT y extiende su funcionalidad mediante el uso de contratos inteligentes.

Figura 4. **Nodos de una red DLT**



Fuente: Lisk. *Nodes*. www.lisk.io/academy/blockchain-basics/how-does-blockchain-work/nodes. Consulta: 5 de noviembre de 2018.

2.2.1. Tipos de Blockchain

A continuación, se mencionarán las distintas variantes de la tecnología Blockchain, las cuales presentan distintas categorías y pueden ser utilizadas con distintos propósitos, y se adecuan a distintos casos de uso.

2.2.1.1. Públicas

Los Blockchain públicos como bien lo dice su nombre son aquellos que están abiertos a cualquiera que desea participar en ellos, cualquiera puede consultar o escribir transacciones sobre ella; así mismo, cualquier participante puede ser parte del proceso de validación o verificación de las transacciones. Otra característica de este tipo es que no existe una entidad central que pueda

delegar o remover roles o permisos, por lo que con esta es posible alcanzar una total transparencia y democracia en la red; normalmente, los usuarios pertenecientes a esta red son anónimos.

2.2.1.2. Privadas

Estos tipos de Blockchain son aquellos que pertenece a un individuo o institución, de la cual dicho individuo o institución tiene el poder de remover distintos permisos a los miembros de la red, ya sea limitándolos únicamente a la lectura o escritura, o ya sea delegando sobre ciertos participantes de la red la tarea de validación de las transacciones. Existe mucho debate acerca del enfoque de este tipo, que choca con las creencias e ideales de conocedores del tema; sin embargo, la aplicación de la tecnología es válida, aunque pueda que no tenga los mismos principios o asegure la confiabilidad que la publica brinda.

2.2.1.3. Federadas o por consorcio

Similares a las privadas, están formadas por un grupo de participantes, de los cuales existen ciertos participantes que son los que tienen a su cargo la toma de decisiones de toda la red; por lo que las decisiones tomadas por esto son en beneficio de todos, y siguen el mismo principio de las privadas de mantener los permisos ya sea de lectura, escritura o validación bajo el control de ciertos participantes.

2.3. Blockchain como estructura de datos

Como se ha mencionado anteriormente, Blockchain es un tipo de tecnología que abarca desde la red de nodos o participantes de la red, hasta el

conjunto de transacciones o información almacenada dentro de cada nodo de la red.

En el núcleo de la tecnología reside uno de sus elementos claves, el cual es de gran importancia y es el que potencia distintos principios de la tecnología, como la seguridad y la transparencia; dicho elemento es una estructura de datos la cual es utilizada para el almacenamiento de la información, dicha estructura está formada de otros elementos, que a su vez utilizan distintos conceptos mencionados anteriormente; dichos elementos hacen uso también de otras estructuras de datos, conceptos y algoritmos relacionados a ellas.

La manera como se almacena la información en una de estas estructuras es mediante el uso de una cadena de bloques de donde radica su nombre. Dicha cadena, no es más que una lista simplemente enlazada, donde cada uno de sus elementos o nodos se les denomina bloques, este tipo de lista pertenece a las estructuras de datos lineales. Habiendo definido la manera general en que se almacena la información se procederá a desglosar y describir los componentes asociados a cada bloque.

2.3.1. Bloque

Es análogo a lo que se conoce como nodo en la teoría de Ciencias de la Computación. Esta estructura es a su vez compuesta de otros elementos, y define la cadena mediante una referencia al bloque anterior. Entre sus componentes están:

- *Hash* del bloque previo
- *Nonce*
- Marca de tiempo

- Raíz del *merkle tree*
- Transacciones

2.3.1.1. *Hash* del bloque previo

Este valor es una referencia hacia el bloque previo, dicha referencia es un valor *hash*, cada bloque dentro de un Blockchain, tiene un valor *hash* único e irrepetible. El primer nodo dentro del Blockchain tiene una referencia nula en este campo, a dicho nodo se le denomina bloque génesis. Este valor pertenece a la cabecera de un bloque.

2.3.1.2. Nonce

Es un valor numérico aleatorio que es utilizado para imponer un determinado nivel de cómputo en las inserciones de bloques dentro del Blockchain; simultáneamente, es utilizado para satisfacer ciertas condiciones propias de cada red de Blockchain; este valor puede ser generado ya sea una vez o una cantidad indeterminada de veces, hasta que el valor en conjunto de todo la cabecera del bloque, coincida con las restricciones mencionadas anteriormente.

Este es uno de los pilares de la tecnología, ya que obliga a los nodos que realizan la inserción de bloques, a realizar operaciones de cómputo de gran consumo previas a la inserción de bloques, lo que a través del tiempo, incurre que la alteración de la información requiera más y más poder de cómputo; llega a un punto en el que sea casi imposible la alteración de la estructura por uno o varios nodos de la red, este valor pertenece a la cabecera de un bloque.

2.3.1.3. Marca de tiempo

Del inglés *timestamp*, esta marca de tiempo contiene la fecha y hora de la actualidad, y se actualiza cada cierto tiempo, este segmento se encuentra en la cabecera del bloque.

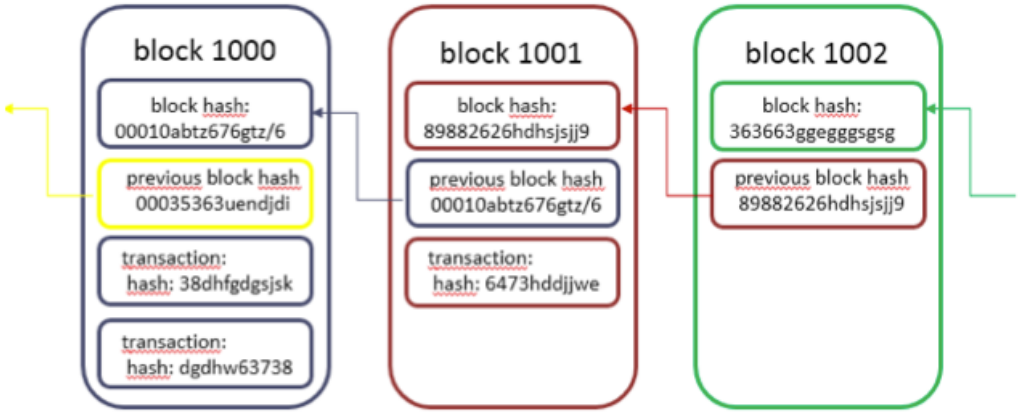
2.3.1.4. Raíz del *merkle tree*

Un *merkle tree* es una estructura de datos no lineal, es un árbol binario de valores *hash*, el cual es construido a partir de las distintas transacciones que existen dentro de un bloque. Esta hace uso de funciones de *hash* criptográficas, sobre cada una de las transacciones de un bloque; forma así un solo valor *hash*, el cual es la clave para evadir la modificación y detectar posibles cambios sobre el bloque, ya que una mínima alteración sobre una transacción alterará no solo la transacción, sino que también esta raíz, y a su vez el valor *hash* del bloque.

2.3.1.5. Transacciones

Una transacción es la unidad más pequeña dentro del Blockchain, la cual contiene información acerca de una operación o evento realizado dentro del Blockchain, como lo pueden ser emisor y receptor, valor de la transferencia, fecha y hora, etc. Un bloque tiene un número específico de transacciones, el cual varía dependiendo de la implementación del Blockchain. Este segmento pertenece al cuerpo del bloque.

Figura 5. **Estructura de almacenamiento Blockchain**



Fuente: MANHART, Sascha. *The Crypto-Guide for Beginners – What is Blockchain?* www.ico.conda.online. Consulta: 01 de noviembre de 2018.

2.4. Contratos inteligentes

Contratos inteligentes, del inglés *smart contracts*; estos son programas que se ejecutan sobre un Blockchain; contienen la lógica de negocio que se le desee dar a este y se ejecutan cuando ciertas condiciones dentro del sistema se cumplen, por lo que posteriormente cuando dichos contratos son cumplidos por las partes involucradas se registran dentro del sistema.

Estos contratos inteligentes son análogos a los contratos que se realizan en el día a día, y son acuerdos entre distintas personas o entidades y normalmente sirven para la transferencia de bienes entre los participantes; el objetivo de estos es el de agregar una capa de negocios al Blockchain y ampliar el alcance y límite que estos tenían previo a su existencia; con los cuales se puede ampliar el funcionamiento y la variedad de casos de uso que estos

tenían, las distintas tecnologías Blockchain utilizan un lenguaje propio para la modelación de estos contratos y presentan distintas limitaciones en cuanto al alcance que estos proveen, estos contratos al igual que la información son inmutables, por lo que cuando son instalados en el Blockchain, estos no pueden ser modificados por ninguna persona.

Así mismo la instalación de uno de estos contratos puede ser hecha por cualquier persona en el caso de los Blockchain de tipo público, y en el caso de los Blockchain de tipo privado los usuarios que necesitan tener permisos para realizar dicha instalación. Así como la información, estos contratos se encuentran de una manera descentralizada, por lo que residen en cada uno de los nodos pertenecientes a la red Blockchain.

2.5. Red entre pares

Red entre pares del inglés *Peer-to-Peer* (P2P), es un tipo de arquitectura que elimina la necesidad de tener un ente centralizado, donde cada nodo o elemento que pertenece a la red tiene comunicación directa con todos y cada uno de los miembros la red, que forma así una arquitectura descentralizada; esto da espacio a eliminar un solo punto de ataque y a que sea posible deshacerse de la necesidad de terceros o intermediarios, dejando solamente a los involucrados en las transacciones de la red.

Las transacciones pueden ser validadas por cualquier miembro de la red, las cuales son replicadas y confirmadas por los demás miembros mediante algoritmos de consenso. Además, cada uno de los miembros es capaz de aportar capacidad de almacenamiento o computo a los demás miembros de la red para cumplir con las tareas necesarias de esta.

3. PRINCIPIOS Y PROCESO DE BLOCKCHAIN

La tecnología Blockchain consta de distintas propiedades las cuales han permitido que se haya perfilado como una de las tecnologías con mayor capacidad para transformar la manera de realizar transacciones entre sistemas; dichas propiedades dan paso a ciertos principios los cuales serán descritos a lo largo de este capítulo, así como el flujo en que se realizan las transacciones y la manera en que se añaden dentro de uno de estos sistemas.

3.1. Inmutabilidad de la información

Una de las principales características o principios de Blockchain es la inmutabilidad de la información, de hecho esta forma parte de los pilares de la tecnología. La inmutabilidad de la información es lograda a través de la misma estructura del Blockchain, se menciona que en la estructura de un bloque cada una de las transacciones no solamente es validada por los miembros de la red, sino que estas forman parte de la raíz del *merkle tree* la cual a su vez es la unión de los valores *hash* de todas las transacciones.

Por lo que si una transacción es alterada o eliminada, esta raíz automáticamente cambia, y dado que cada bloque tiene un valor único el cual está compuesto de esta raíz, si este valor no concuerda con la referencia que posee el siguiente bloque, es posible de detectar los cambios realizados o anomalías en la información; de allí el hecho que la información dentro de esta tecnología sea considerada inmutable; así mismo, se menciona que todo bloque contiene un valor llamado *nonce*, el cual exige una cantidad de cómputo

elevada, la cual en caso de algún intento de alteración debe de ser calculada para cada uno de los bloques, haciendo casi imposible la misma.

3.2. Propiedades

A continuación, se describen algunas propiedades que son de relevancia de la tecnología y potencian su uso; y se podrá apreciar que estas propiedades forman parte de la tecnología, por el diseño y la forma en que esta se comporta.

3.2.1. Persistencia

Al inicio de este documento se menciona como las estructuras de datos fueron creadas y dan paso al manejo de la información. A lo largo del tiempo las bases de datos (constituidas por estructuras de datos) solventaron problemas de persistencia de la información, sin embargo con el paso del tiempo han surgido nuevas necesidades para el manejo de la información, y nuevas topologías de bases de datos han sido creadas, con el objetivo que la información persista de una mejor manera a lo largo del tiempo.

Cada nodo perteneciente a una red de Blockchain, posee una copia de la información; dicha información y estructura es la misma para todos los miembros de la red, por lo que si un nodo de la red falla y la información dentro de este es eliminada, o si existe un fallo de hardware o software que no permita la recuperación de la misma, este puede ser reemplazado con facilidad y la información recreada con facilidad; por lo que la única manera en que la información pueda perecer por completo es si todos y cada uno de los miembros de la red sean eliminados por completo.

3.2.2. Disponibilidad

Una propiedad de todo sistema de cómputo y de gran impacto en los mismos; hemos hablado que cada uno de los nodos pertenecientes a una red de este tipo de tecnologías, tiene acceso a la misma información que los demás, la cual es replicada en cuestión de segundos o minutos, por lo que cada usuario de la red tendrá acceso a la información cuando lo desee; se ha visto como a lo largo del tiempo la necesidad de tener alta disponibilidad ha causado la creación de nuevas arquitecturas.

La naturaleza propia de Blockchain y su arquitectura está diseñada para la alta disponibilidad ya que permite de manera rápida y segura la disponibilidad de la información para cada uno de los miembros de la red, así como para la persistencia, cuando algún nodo de la red se encuentre en un estado que no pueda proveer sus servicios otros miembros de la red pueden proveer capacidad de cómputo para validar transacciones y crear bloques, o replicar la información a sus demás pares, por lo que no es necesario la presencia de todos los nodos de la red para que se pueda seguir con las operaciones respectivas.

3.2.3. Transparencia

Al inicio del capítulo se mencionaba que la misma estructura de Blockchain daba paso a la inmutabilidad de la información, y que al momento de que exista un cambio dentro de ella los cambios eran detectados, por lo que una transacción no podía ser alterada sin poder corromper la estructura, además de este mecanismo implementado por la estructura, existen ciertos algoritmos que existen dentro de Blockchain, con los que cada transacción es validada por distintos miembros pertenecientes a la red y permite que

responsabilidad de creación de estas y de los bloques no recaiga sobre un solo miembro, dichos algoritmos se explican más a detalle posteriormente.

3.2.4. Seguridad

Se ha hablado de varios principios que sin lugar a duda hacen que las transacciones sean consideradas seguras; sin embargo, existen otros factores con los cuales se puede asegurar la información y la participación dentro de ellas, entre los cuales está la criptografía, ya sea para realizar transacciones; para identificar de manera anónima (pero válida) dentro de la red la criptografía permite asegurar que cada una de las transacciones sea válida o bien para acceder a nuestros activos dentro de la red, e identificarse de manera única ante los demás miembros; el uso de llaves públicas y privadas permitirán descifrar y cifrar cada transacción.

3.3. Transacciones

Las transacciones son la unidad mínima dentro de un Blockchain, las cuales deben pasar por un proceso luego de ser añadidas a un bloque, dicho proceso se describe a continuación.

3.3.1. Validación de bloques

El proceso de validación de nuevos bloques a la estructura se da por medio de distintos algoritmos, llamados algoritmos de consenso. Vale la pena recalcar que los bloques pueden ser agregados solamente por miembros pertenecientes a la red, para lo cual la criptografía forma parte del proceso; hace uso de llaves públicas y privadas (criptografía asimétrica), las

transacciones marcadas por las llaves públicas de los distintos miembros de la red son verificadas con las llaves privadas de los demás miembros y viceversa.

3.4. Algoritmos de consenso

En un sistema descentralizado como lo es Blockchain es necesaria la sincronización de todo el sistema; estos algoritmos se encargan de validar, recibir y replicar la información de todos los nodos, permitiendo llegar a un único estado consistente para todos los miembros de la red. Estos algoritmos permiten agregar y validar nuevas transacciones y bloques al Blockchain, dando paso así a una sinergia en el sistema. Entre los distintos algoritmos están:

- *Proof of work*
- *Proof of stake*
- *Proof of capacity*
- *Proof of elapsed time*
- *Proof of deposit*
- *Proof of activity*

Estos algoritmos varían en la implementación y cada uno permite potenciar distintos atributos del sistema, por lo que para distintos Blockchain pueden utilizarse distintos algoritmos de consenso.

4. BLOCKCHAIN COMO TECNOLOGÍA FUNDACIONAL

Aunque Blockchain surgió como una tecnología hace unos cuantos años, se ha ido desarrollando y buscando nuevos mercados, no solamente se posiciona como una tecnología disruptiva, sino como una tecnología fundacional. Aunque su uso puede ser parcial e integrado con sistemas como los que normalmente se conoce.

Se le denomina como una tecnología fundacional ya que su implementación normalmente genera cambios totalmente trascendentes en el área en el que se implemente; ya que uno de los objetivos de esta tecnología es la interacción con otras entidades que formen parte en nuestro modelo de negocio, fuerza a que el modelo de negocio tome en cuenta muchas más variables que con un sistema tradicional no se debería, por lo que el uso de esta tecnología revoluciona por completo la base de ciertos mercados, moldeando así de una nueva manera los mercados.

Aunque Blockchain es una tecnología que de acuerdo a muchos es una de las tecnologías con mayor impacto en el futuro próximo, su auge se ve aplazado para un periodo de tiempo no tan cercano, esto por el hecho de que su implementación genera nuevos planteamientos en las distintas áreas en las que es implementada; sin embargo, Blockchain ya se ha puesto en marcha en la industria y ha demostrado su potencial; a continuación, se describen las aplicaciones, mercados y casos de uso más relevantes hasta la fecha.

4.1. Aplicaciones

Los distintos beneficios de Blockchain han sido mencionados anteriormente; a continuación, se mencionan distintas aplicaciones y casos de uso más comunes.

4.1.1. Criptomonedas

Aunque como se menciona al inicio de este documento estas surgieron alrededor de los años 90, el uso de criptomonedas en los últimos años ha aumentado cada vez más. Las criptomonedas son bienes digitales utilizados en distintas transacciones como medio de pago o intercambio de valor, para lo cual utilizan criptografía para validar y autorizar dichas transacciones; además de esto cada transacción es validada por distintos miembros de la red a los cuales se les denominan mineros.

El beneficio de estas radica en el desacoplo de terceros de la transferencia de valor entre dos entidades, estos terceros pueden ser bancos, compañías de tarjetas de crédito, procesadores de pagos entre otros, los cuales al ser incluidos en las transacciones no solo demoraran los procesos, sino que generan altos intereses que son indeseables y muchas veces innecesarios.

La eliminación de intermediarios dentro de una transacción se da debido a la eliminación del problema del gasto doble o del inglés *the double spend problem* el cual es resuelto a través del uso de un Blockchain de tipo público, que permite que cada uno de los nodos de la red pueda validar y rastrear las transacciones hasta su origen.

Si bien se ha mencionado que Blockchain surge como parte de las criptomonedas, las criptomonedas no podrían ser posibles sin esta tecnología. El uso de criptomonedas se ha popularizado a lo largo del tiempo, y han surgido distintas monedas; además, el uso de este bien digital ya es aceptado como medio de pago en transacciones de bienes materiales que van desde comidas y bebidas, hasta boletos de avión, servicios de internet, servicios en computación en la nube y muchas otras más.

Entre las criptomonedas se pueden encontrar que sin lugar a duda a Bitcoin es la más popular; además, fue el punto de inicio de este medio de pago; sin embargo, desde su liberación como código abierto, han surgido variaciones de esta las cuales son llamadas *altcoins* o monedas alternativas de las cuales se pueden mencionar las más relevantes:

- Ethereum
- Ripple
- Litecoin

4.1.2. Internet de las cosas

Internet de las cosas o IoT, por sus siglas en inglés, ha sido una tecnología muy utilizada en la actualidad, no solo en la vida diaria, sino también en la industria, ya que el uso de sensores, la baja capacidad de cómputo necesaria para la transmisión de información y el análisis de información permiten brindar una capacidad de respuesta rápida y nos permiten también la automatización y facilitación de distinto tipos de tareas; sin embargo, el bajo uso de consumo de recursos que utiliza también brinda una baja capacidad de seguridad.

El uso de esta tecnología ha sido a lo largo de los años no solo beneficiosa para empresas, gobierno y personas particulares; también ha sido duramente criticada por muchas personas debido a la alta vulnerabilidad en seguridad que presentan este tipo de sistemas; además, ha sido criticada por la invasión que representa el uso de estos sistemas en la vida de las personas y también en el uso que dan las empresas a la información obtenida y generada a través de estos sistemas.

Debido a la naturaleza de Blockchain de transparencia y seguridad, y con las nuevas formas como presenta esta tecnología, como las de tipo privada o permisiva, permiten que cada uno de los componentes pertenecientes a un sistema IoT tengan capacidad limitada sobre el sistema; eleva así su grado de seguridad y el uso de criptografía de la información permiten mitigar en gran manera vulnerabilidades que el internet de las cosas presenta en la actualidad. Además, como se verá más adelante existen otros mercados o casos de uso en los cuales el IoT aplicado sobre una cadena de suministros o sobre un flujo de negocio aporta gran valor al producto final y al consumidor.

4.1.3. Energía

El uso de Blockchain ha llegado incluso a la industria de la energía, existen compañías que han iniciado la comercialización de energía renovable generada por un particular desde una instalación casera a través de sistemas de paneles solares, con el objetivo de crear una red de generación de energía escalable y autónoma. Otra clara aplicación de Blockchain y de las criptomonedas en esta área es el petro, una criptomoneda desarrollada por el gobierno de Venezuela para el comercio con extranjeros de aceite, gasolina, oro y diamantes que prometió en su momento una solución a la crisis financiera que atraviesa dicho país en los años de 2017 y 2018.

4.1.4. Medicina y salud

La medicina y la salud son de los ámbitos en los que Blockchain ha demostrado su mayor potencial y son áreas con un futuro prometedor para esta tecnología. La medicina es un tema delicado, no solo permite mejorar la salud o llegar a salvar vidas, también, un fallo en la producción de ellas puede significar la prolongación de enfermedades o empeoramiento de las mismas e incluso puede llegar a ocasionar la muerte de cientos o bien miles de personas.

El uso de Blockchain en la medicina abarca desde la producción del producto en las empresas farmacéuticas hasta la entrega y consumo del cliente final, haciendo uso de tecnologías como el internet de las cosas es posible rastrear la procedencia ya sea desde las empresas farmacéuticas hasta los proveedores que surten a estas empresas; además, es posible verificar los procesos que utilizan para su producción, que se hayan cumplido los estándares necesarios en la producción de estas, y dependiendo del nivel de procedencia que se desee es posible rastrear a las personas involucradas en dichos procesos.

También es posible involucrar el proceso de entrega de las farmacéuticas a los distribuidores para verificar que se manejen con el debido cuidado que estas necesiten y que lleguen bajo los requerimientos y los propios de las medicinas.

Blockchain en la salud también promete y se ha visto envuelta en la telemedicina; MedChain es un Blockchain global que permite el registro de historial médico para facilitar la manipulación de la información a través de distintos procedimientos médicos que permiten también el intercambio de información entre distintas entidades, públicas o privadas. El objetivo de esta es

permitir la interacción paciente-medico de una manera más sencilla, que permite almacenar resultados en un registro global, con el fin de dar continuidad a tratamientos en distintas partes del mundo, con distintos médicos y especialistas.

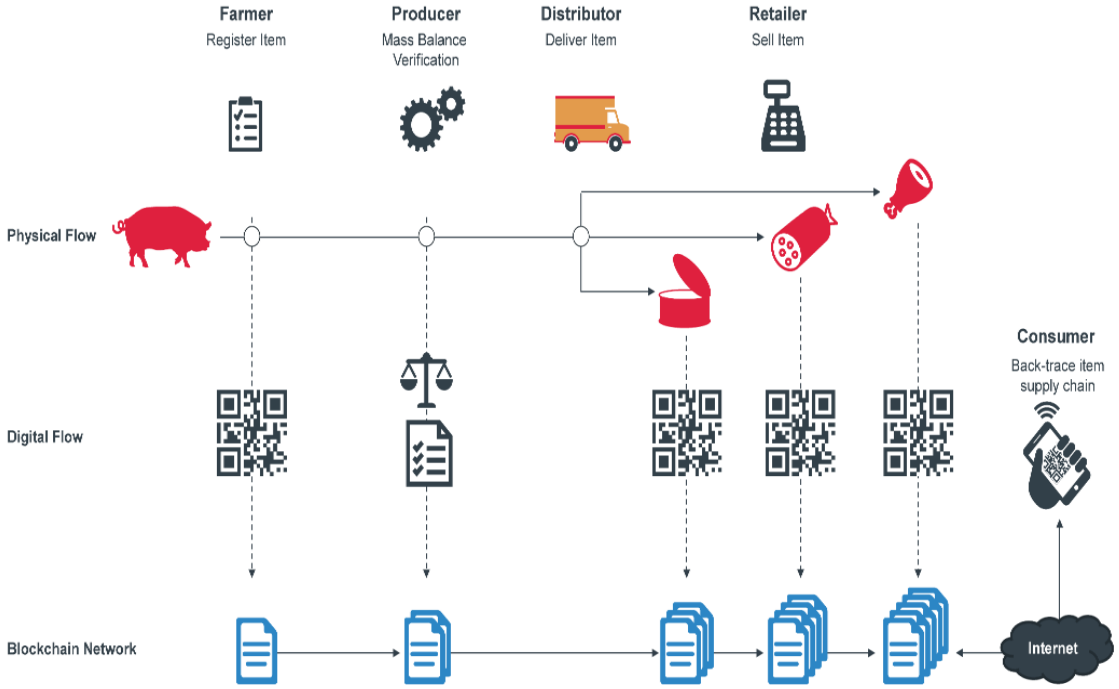
4.1.5. Cadena de suministros

Una de las aplicaciones de Blockchain con mayor potencial es sin duda la procedencia; así mismo, uno de los objetivos de esta tecnología es tener una fuente de información verdadera y confiable entre los distintos participante de una red de este tipo.

La procedencia en las cadenas de suministros es de gran relevancia ya que permite llevar un control de los distintos bienes a lo largo de todo el proceso desde la materia prima utilizada para crear o tratar los bienes, durante los distintos procesos de tratamiento de la misma, en los distintos procesos de transporte, en el proceso de almacenamiento, hasta llegar al consumidor final; en donde normalmente se involucran distintos participantes independientes los unos de los otros.

A través de esta aplicación es posible brindar al cliente final la satisfacción y confiabilidad que desea llevando el registro de todo este proceso dentro de un Blockchain; brinda detalle de quiénes estuvieron involucrados en este proceso, cómo ha sido tratado el producto y si cumple o no con las expectativas del mismo.

Figura 6. **Cadena de suministro**



Fuente: VAN ROOYEN, Jan. *Blockchains for supply chains – part II.* www.resolvesp.com. Consulta: 22 de octubre de 2018.

4.1.6. Criptoanclas

Otra aplicación de Blockchain que tiene un futuro con gran potencial son las denominadas criptoanclas; las cuales son parte de una investigación dirigida por investigadores de la empresa llamada International Business Machines Corporation (IBM); engloban todas las aplicaciones discutidas con anterioridad y no solo extienden su alcance sino que lo mejoran, incursionando en los mercados mencionados anteriormente; no solo permiten rastrear la

procedencia, sino que tienen como objeto principal comprobar la autenticidad de los objetos.

Aunque Blockchain se enfoca más en la naturaleza de las transacciones, las criptoanclas se enfocan en la autenticidad de los bienes. Estas denominadas criptoanclas pueden ser incrustadas en los productos, dentro de microcomputadoras o bien pueden ser creadas a través de códigos ópticos, apoyándose de la inteligencia artificial y la criptografía; proveen una única e irrepetible manera de identificar los objetos y reconocerlos; esta tecnología promete combatir y disminuir en gran manera el fraude y la falsificación de objetos.

5. MATERIAL DIDÁCTICO PARA EL ESTUDIO DE LA TECNOLOGÍA BLOCKCHAIN

A continuación, se muestran distintas herramientas teóricas y prácticas las cuales se proveen para el uso en la docencia de la tecnología Blockchain desde la perspectiva de las estructuras de datos.

5.1. Marco de trabajo

Como se ha mencionado en capítulos anteriores, Blockchain es una tecnología que es asistida de otras tecnologías y al mismo tiempo hace uso de distintos conceptos de las ciencias de la computación en la manera en la que realiza la gestión de la información; sin embargo, todos esos conceptos y algoritmos aunados a esos conceptos requieren del conocimiento previo y secuencial de los mismos, por lo que existe una relación de dependencia entre ellos, para lo cual se propone un flujo de enseñanza.

El siguiente flujo de enseñanza consta de cinco fases, en donde cada fase consta ya sea de uno o más conceptos pertenecientes al tema a tratar en la misma; cada fase puede ser extendida en sus conceptos o pueden ser agregadas nuevas fases que introduzcan nuevos conceptos; sin embargo, para la enseñanza de la tecnología Blockchain, es importante respetar el flujo. Las fases de las que consta son las siguientes:

- Estructuras de datos lineales
- Estructuras de datos no lineales
- Tablas de dispersión
- Criptología
- Blockchain

A continuación, se describen las distintas fases y su justificación.

5.1.1. Estructuras de datos lineales

Como se describió en capítulos anteriores la estructura de la cadena de bloques que existe dentro de la tecnología Blockchain, se comporta de la misma manera que una lista simplemente enlazada; así mismo, los bloques poseen una estructura similar a esta estructura de datos, las cuales no son las únicas razones por las que encabeza el flujo de la enseñanza, sino que a lo largo del tiempo ha encabezado dicho flujo por el bajo nivel de complejidad que esta presenta; así mismo, este tipo de estructuras sientan la base para el diseño y la comprensión de estructuras de datos de mayor complejidad. Algunos factores que pueden ser incluidos en esta fase son:

- Listas simplemente enlazadas
- Complejidad de la estructura
- Variantes en la implementación
- PEPS y UEPS

Uno de los factores que vale la pena tomar en cuenta es la complejidad que presentan los distintos algoritmos, así como las distintas maneras como este tipo de estructuras pueden ser implementadas y la diferencia que existe entre ellas tanto en estructura, implementación y complejidad algorítmica.

Otro concepto que puede ser tomado en cuenta es la distinción entre los conceptos de primero en entrar primero en salir (PEPS) y último en entrar primero en salir (UEPS), los cuales no solo forman parte de conocimientos generales de las estructuras de datos, sino que dan un panorama de como las transacciones dentro de un sistema como lo es Blockchain, pueden ser manejadas.

5.1.2. Estructuras de datos no lineales

Como segunda fase se encuentran las estructuras de datos no lineales, de las cuales sobresalen los árboles, por el hecho de ser de mayor complejidad que las estructuras de datos lineales. Un concepto introductorio en esta fase, el cual es fundamental, es el de recursividad, ya que es una forma en que los árboles son recorridos y manipulados que presenta un grado de abstracción mayo; así mismo, los arboles binarios de búsqueda, árboles AVL y árboles B, son de gran ayuda para la comprensión de estructuras compuestas de mayor complejidad como los *merkle tree* de los bloques de Blockchain, los cuales son discutidos en la fase de criptografía y que pueden ser binarios o de un mayor grado.

5.1.3. Tablas de dispersión

Las tablas de dispersión se prestan a la introducción de nuevos conceptos, como las funciones *hash*, las cuales van íntimamente relacionadas junto con la criptografía, con la generación de información dentro del Blockchain; estas permiten comprender el funcionamiento básico de la encriptación y dan paso a su estudio. Así mismo, las distintas estrategias de resolución de colisiones permiten reforzar conocimientos previos de los tipos de estructuras lineales y no lineales. Además, estas estructuras en conjunto con

las funciones *hash* sirven para iniciar la comprensión entre la transformación e indexación de la información, la cual abre paso a un funcionamiento análogo en como la información almacenada en el Blockchain es manejada y referenciada, tanto en estructura como en valor.

5.1.4. Criptología

Como cuarta fase se presenta la criptografía, una de las disciplinas que es de las más importantes o de las que aporta mayor valor en la tecnología Blockchain; es recomendable la enseñanza de los siguientes conceptos en esta fase:

- Criptología
- Criptografía
- Criptografía simétrica
- Funciones hash criptográficas
- Criptografía asimétrica
- Firmas digitales
- *Merkle tree*

Es importante la enseñanza de los distintos paradigmas de la criptografía para iniciar un conocimiento a conceptos más avanzados; de igual manera, un concepto fundamental en esta fase son las funciones hash criptográficas, las cuales son de gran aplicación en la criptografía asimétrica y las firmas digitales; y sientan la base para conceptos de Blockchain, como los *merkle tree,* las cuales son estructuras no lineales formadas meramente de valores hash o huellas digitales, todos estos conceptos relacionados a la tecnología Blockchain.

Es recomendable el uso de herramientas de software para la ejemplificación de estos conceptos e iniciar la introducción al manejo de herramientas que son necesarias dentro de una tecnología como Blockchain.

5.1.5. Blockchain

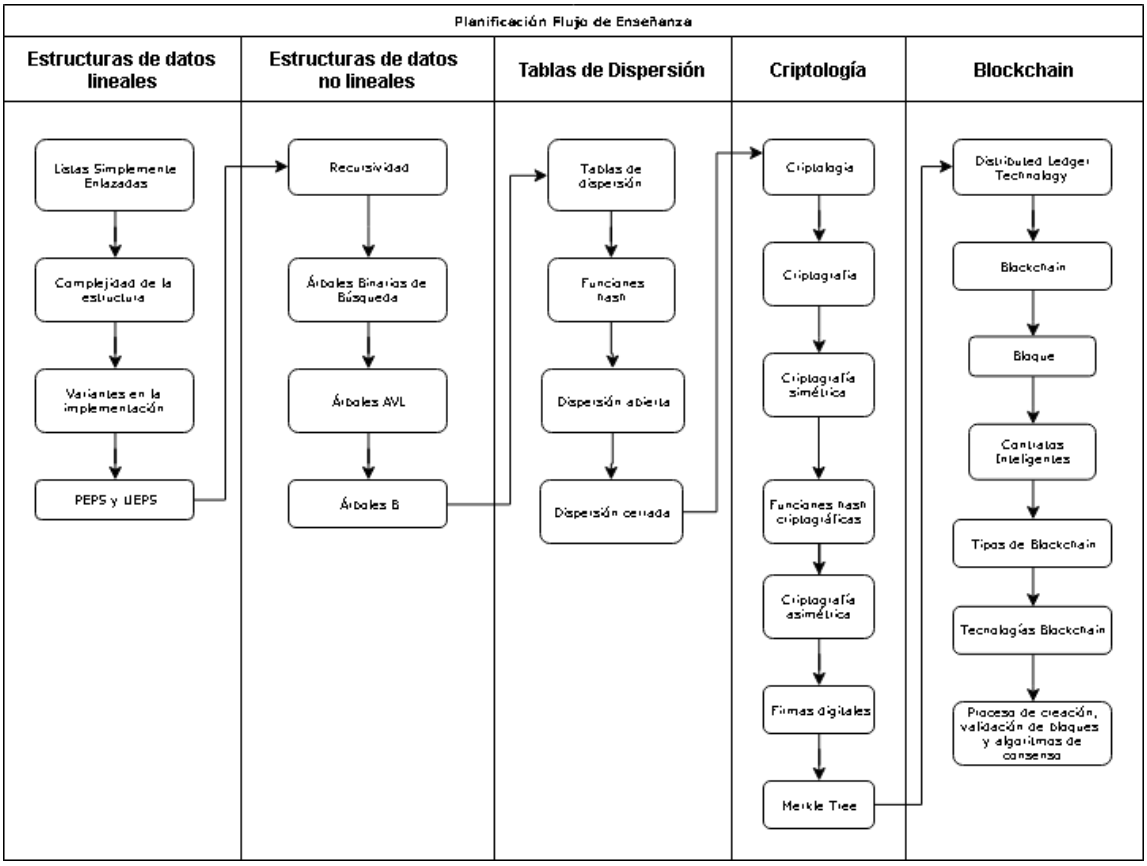
Por último, se encuentra Blockchain, la tecnología objeto de estudio, una composición de los conceptos mencionados en las cuatro fases anteriores y, sin lugar a duda, una aplicación de las estructuras de datos con mayor potencial y que muestra que su estudio puede ser aplicado en la vida real y permite crear nuevas formas de relacionar distintos mercados que mejoran la seguridad, la comunicación y la facilidad de interacción con distintos sistemas. En esta fase existe una gran cantidad de conceptos de interés que pueden ser estudiados; sin embargo, ya que la perspectiva que se propone es el de las estructuras de datos, se presentan los siguientes que son una recopilación de las estructuras de datos:

- *Distributed ledger technologies*
- Blockchain
- Bloque
- Contratos inteligentes
- Tipos de Blockchain
- Tecnologías Blockchain
- Proceso de creación, validación de bloques y algoritmos de consenso

En esta fase se finaliza con la recopilación de todos los conocimientos obtenidos en las fases previas y se basa principalmente en la aplicación de los conceptos a esta tecnología, como se relacionan e interactúan entre sí y como se construye a partir de todos esos conocimientos esta tecnología que brinda

tantos beneficios, no sin antes definir su estructura, los tipos en los que se presenta y las distintas tecnologías que existen.

Figura 7. **Planificación, flujo de enseñanza**



Fuente: elaboración propia, empleando Draw.io.

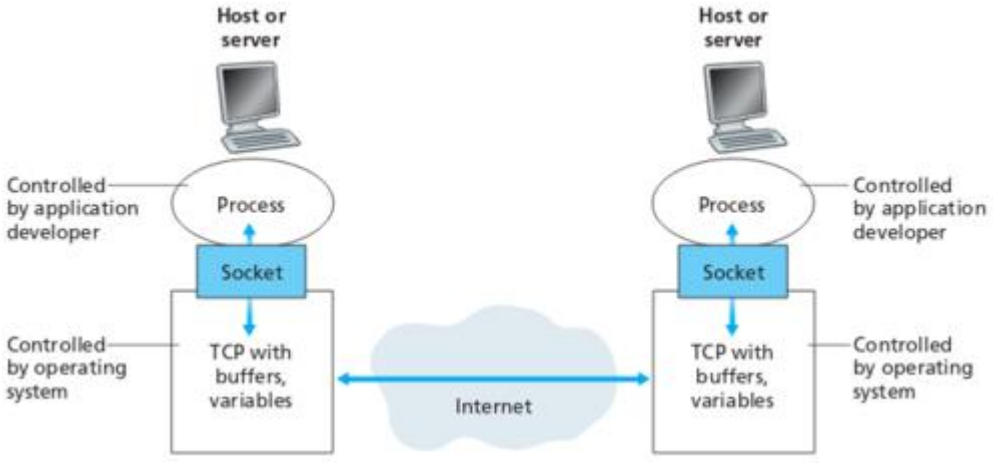
5.2. Tecnología análoga

Para demostrar un funcionamiento semejante a una red entre pares o P-2-P se recomienda el uso de clientes y servidores con *sockets* TCP los cuales son:

- Fáciles de implementar
- Amplias variedades de implementación
- Provistas por los distintos lenguajes de programación
- Provistas por librerías de los lenguajes de programación y *frameworks*

Dicha tecnología tiene el fin de construir una red de comunicación con los distintos estudiantes, con los cuales pueden poner en práctica conceptos básicos de las redes entre pares.

Figura 8. **Sockets TCP**



Fuente: Ugnes. *What is the difference between a port on a socket?* www.stackoverflow.com. Consulta: 23 de octubre de 2018.

5.3. Ejemplo de implementación

A continuación, se ejemplifica una implementación simple en el lenguaje de programación Java, la cual puede ser encontrada en la dirección web https://github.com/0722-EDD/blockchain que muestra de una manera sencilla cómo diseñar un Blockchain sencillo con sus atributos principales y relevantes; hace uso de librerías propias del lenguaje así como la integración de conceptos de estructuras de datos lineales y no lineales.

5.3.1. Blockchain implementación en Java

En los capítulos previos se menciona la estructura de Blockchain, los principios básicos y su comportamiento; a continuación, se muestra la implementación de un ejemplo básico, que ejemplifica a gran escala la composición de la estructura, y la manera como se comporta; además, se enlazan conceptos de estructuras de datos lineales en la implementación de la misma.

5.3.1.1. Requerimientos

- Java JDK 8
- Netbeans 8.2 (opcional)
- Graphviz 2.38

5.3.1.2. Estructura del proyecto

El proyecto llamado MiBlockchain consta de tres archivos principales:

- Block
- Blockchain
- Transaction

Los cuales poseen clases denominadas de igual manera para cada archivo, y de las cuales la clase Transaction posee la definición de transacciones básicas, con origen, destino y valor; la clase Block posee la definición de un bloque, y la clase Blockchain posee la definición del Blockchain o cadena de bloques implementada como una lista simplemente enlazada.

5.3.1.2.1. Transaction

La clase Transaction hace alusión a cada una de las transacciones u operaciones que se realizan dentro del sistema, ver apéndice 1. Esta consta de cuatro atributos de tipo cadena:

- Origen
- Destino
- Valor
- Timestamp

Origen y destino representan las entidades que forman parte de la transacción; valor representa el monto o cantidad la cual une al origen y destino en una transacción; el valor de timestamp es la marca de tiempo, es generada automáticamente y representa el tiempo exacto en el que se realiza la transacción.

5.3.1.2.2. Block

La clase Block almacena las transacciones y la cabecera básica de un bloque, ver apéndice 2. Esta clase posee los valores:

- Hash
- HashPrevio
- Timestamp
- MerkleTreeRoot
- Transacciones

Además, se implementa el concepto de estructuras de datos lineales, almacenando de una manera lineal consecutivas los demás bloque mediante una referencia al bloque previo llamado previo, y almacena únicamente cuatro transacciones mediante un arreglo de la clase Transaction.

Así mismo, esta estructura cuenta con el método generarHash el cual permite obtener el valor *hash* de las transacciones almacenadas para cada uno de los bloques; hace uso de la función *hash* SHA-256 para obtener el valor de las transacciones. Este método también es utilizado al momento de crear un nuevo bloque; devuelve el *hash* del bloque previo.

La función crearMerkleRoot es utilizada para crear la raíz del m*erkle tree*, hace uso de una estructura simple; combina cada una de las cuatro transacciones, retornando un único valor unificado de todas las transacciones.

5.3.1.2.3. Blockchain

La clase Blockchain se encarga de almacenar la estructura completa de bloques, ver apéndice 3. Posee como atributos:

- Bloque génesis
- Bloque último
- Número de bloques

Al crear el Blockchain se hace uso del método agregarBloque, el cual cumple con la característica principal del Blockchain y del bloque génesis, que es el de poseer un valor *hash* previo nulo o igual a 0 para el primer bloque de la cadena el cual es llamado génesis, que además indica el inicio de la cadena de bloques.

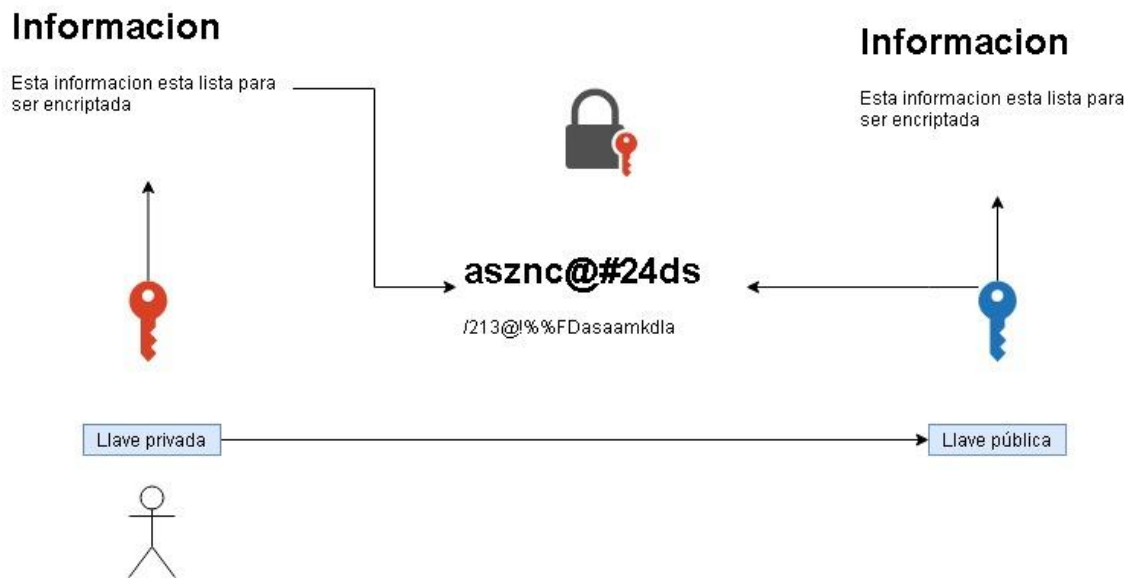
El método agregarBloque también permite agregar un nuevo bloque al final de la cadena, al momento de la creación se envía el valor *hash* del último bloque que existe en la cadena y se actualiza el último bloque de la cadena al nuevo bloque creado.

5.4. Criptografía asimétrica

Como se ha mencionado anteriormente, la principal virtud de Blockchain es la seguridad, la cual está íntimamente ligada al uso de la criptografía. Se eligió una herramienta nativa de los sistemas operativos GNU/Linux que permite ejemplificar de una manera sencilla el uso de la criptografía asimétrica, para la enseñanza de esta disciplina de una manera práctica, la cual consiste en el uso de la herramienta de software llamada openssl.

La siguiente imagen muestra el flujo de funcionamiento de esta herramienta sobre cierta información.

Figura 9. **Encriptación asimétrica**



Fuente: elaboración propia, empleando Draw.io.

El uso de la herramienta openssl es bastante sencillo; siguiendo las premisas de la encriptación asimétrica, si seguimos el siguiente flujo:

- Crear la llave privada.
- Crear la llave pública a partir de la llave privada.
- Seleccionar el archivo a asegurar mediante la encriptación.
- Utilizar la llave privada para cifrar el archivo seleccionado para descifrar.
- Verificar el contenido del archivo encriptado.
- Distribuir el archivo encriptado, así como la llave pública a los destinatarios deseados.

- Utilizar la llave pública para descifrar del archivo.

5.5. Herramientas Blockchain

En esta sección se hará uso de dos distintas tecnologías para la modelación de la lógica del caso de uso descrito más adelante; las tecnologías a utilizar son:

- Hyperledger
- Ethereum

Ambas herramientas permiten la modelación de lógica de negocio sobre sus respectivas tecnologías Blockchain; estas herramientas utilizan distintos paradigmas, por lo que la funcionalidad del caso de uso a modelar se verá limitada por las características que cada una posee.

5.5.1. Caso de uso: red de transporte público

En las siguientes implementaciones se realiza la modelación de una red simple de transporte público, que consiste en el siguiente escenario: un sistema de transporte público maneja distintos descuentos y distintos precios de acuerdo a tres categorías: estudiantes, universitarios y personas de la tercera edad; los estudiantes y personas de tercera edad gozan de transporte gratuito; los precios varían dependiendo del periodo del día, ya sea día o noche.

El sistema necesita almacenar los siguientes datos tanto de pilotos como de usuarios:

- Id
- Nombres
- Apellidos
- Edad

Así mismo, se desea llevar el control de autobuses mediante su placa, asignándole también un piloto a cada uno de ellos. Los usuarios del transporte cuentan con una tarjeta a la cual se le puede recargar un saldo cada cierto tiempo. Para los pilotos se desea almacenar también el tipo y número de licencia así como el número de transacciones, accidentes e incidentes asociados a este. Se necesita que se almacene cada una de las transacciones realizadas en el sistema; se almacena para cada transacción el usuario que realiza el pago, así como el saldo que tenía en ese momento, el autobús en el que se realiza el pago y el periodo del día.

El escenario descrito anteriormente, es un caso de uso típico, que puede ser modelado mediante sistemas tecnológicos tradicionales; a continuación, se describen las tecnologías para la modelación de dicho escenario y sus respectivas implementaciones.

5.5.2. Hyperledger

Es un proyecto que alberga distintas tecnologías Blockchain, la cual es de código abierto y reúne a distintas empresas y expertos alrededor del mundo para la construcción de distintas soluciones de este tipo. Se han seleccionado dos tecnologías, que cumplen con la labor didáctica para los estudiantes de las estructuras de datos, ya que son de fácil uso, y de corto aprendizaje, las cuales se describen a continuación.

5.5.2.1. Hyperledger Fabric

Es un *framework* perteneciente al proyecto Hyperledger de la Fundación Linux, el cual consiste en una plataforma Blokchain de tipo privada, diseñada para el uso en la industria, la cual es altamente modular y permite la personalización de muchos de sus componentes; brinda así una alta capacidad de adaptación a las distintas necesidades y mercados.

5.5.3. Hyperledger Composer

Es una herramienta perteneciente también al proyecto Hyperledger, que permite la construcción de redes de negocio, la creación de contratos inteligentes y aplicaciones Blockchain; además, permite la construcción de estas soluciones mediante el lenguaje de programación Javascript y el sistema de paquetes npm junto a node.js.

Uno de los aspectos que permite que esta herramienta sea de fácil uso es el hecho de que el lenguaje que utiliza para modelar la red y la lógica de la misma; es bastante sencilla; además, tiene una curva de aprendizaje bastante corta y posee integración con otras herramientas utilizadas en el desarrollo de software. Esta herramienta se ejecuta sobre Hyperledger Fabric, por lo que el uso de ambas es necesario.

5.5.3.1. Requerimientos

De acuerdo a la documentación en el sitio web de Hyperledger Composer, para la instalación del sistema de desarrollo es necesario el siguiente software:

- Sistema operativo: Ubuntu Linux 14.04 LTS o Ubuntu Linux 16.04 LTS o Mac OS 10.12.
- Docker Engine versión >= 17.03.
- Docker-Compose versión >= 1.8.
- Node versión 8.9.x.
- Npm versión 5.x.
- Git versión >= 2.9.
- Python versión 2.7.x.

5.5.3.2. Flujo de trabajo

Para realizar una labor educativa del funcionamiento de esta tecnología, se muestra a continuación el flujo que utiliza la herramienta, que abarca desde establecer el ambiente sobre el cual se trabajara, hasta llegar a la parte con mayor importancia que es la de trabajar la capa de negocio sobre la tecnología Blockchain.

- Cumplir con los requerimientos para Hyperledger Composer, descritos anteriormente.

- Completar la configuración de las herramientas que servirán para el desarrollo del software.

- Crear y modelar la red de negocio.

- Integrar con las demás herramientas de desarrollo de software para obtener un resultado funcional.

5.5.3.3. Configuración del ambiente de desarrollo

Además de los requerimientos de software es necesario el uso de los siguientes paquetes para el desarrollo y construcción de software:

- Composer-cli
- Composer-rest-server
- Generator-hyperledger-composer
- Yo

De los cuales todos son pertenecientes al sistema de administrador de paquetes npm del lenguaje de programación Javascript.

Para este ejemplo se hace uso de distintas características de la herramienta Hyperledger Composer para modelar dicho caso de uso, entre los cuales se encuentran:

Uso de entidades abstractas para la creación de participantes de la red, así como el uso de herencia para los participantes de la red. Uso de enumeraciones para realizar distinción entre los periodos del día. Nombre de la red: ‘org.usac.transporte’.

El proyecto se puede encontrar en la dirección web https://github.com/0722-EDD/transporte-publico y consta de tres archivos principales:

- Org.usac.transporte.cto
- Logic.js
- Permissions.acl

En el archivo org.usac.transporte.cto, ver apéndice 4, se define el modelo del negocio, sus participantes, bienes, transacciones eventos y sus atributos, en este se hace uso del lenguaje Hyperledger Composer Modelling Language, y en el archivo permissions.acl, se definen los permisos que contendrán cada participante de la red. A continuación, se verá la definición para el sistema de transporte público:

Primero se define el nombre del espacio de trabajo, que se definió como org.usac.transporte, se hace uso de una clase abstracta llamada Persona. Para la creación de los pilotos y usuarios se heredan los atributos de la clase abstracta Persona. Se declaran los autobuses como activos de la empresa, los cuales se identifican por medio del número de placa. Se hace uso de las enumeraciones definidas como día y noche, para los periodos del día. Por último, se definen las transacciones haciendo referencia a la relación entre usuarios y autobuses.

A continuación, se define la estructura del archivo logic.js, ver apéndice 5, en el cual se define la lógica del negocio a aplicar, dicha lógica se define en lenguaje Javascript. Dicho archivo en su estructura define las transacciones definidas en el modelo del archivo org.usac.transporte.cto; para este ejemplo las transacciones son definidas como pago, esta función se ejecutará para cada instancia de esta transacción. El flujo que sigue esta función es el siguiente:

- Establecer cuotas del servicio.
- Obtener los participantes involucrados en la transacción, tanto pilotos como usuarios.
- Ejecutar la lógica del negocio, verificar el tipo de usuario y aplicar los respectivos descuentos.

- Guardar los cambios en el libro de transacciones y actualizar la respectiva información.

Por último, el archivo de permisos de acceso permissions.acl, ver apéndice 6, el cual le da la característica de Blockchain privada a esta tecnología. En este se define el acceso que tendrá los distintos participantes de la red. Para este caso se utilizarán los valores por defecto.

5.5.4. Ethereum

Es un Blockchain de tipo público, que permite el desarrollo y uso de aplicaciones descentralizadas sobre su Blockchain, y facilita la creación de *Smart Contracts*; además, posee una criptomoneda nativa la cual es llamada ether que es la base de compensación en las transacciones realizadas entre los participantes de esta red.

5.5.4.1. Ethereum Virtual Machine

La Ethereum Virtual Machine, o máquina virtual de Ethereum, es la que permite la ejecución de contratos inteligentes sobre la red de Blockchain de esta tecnología; además, basa su funcionamiento en un parámetro denominado gas, el cual limita la cantidad de cómputo que puede ser realizada en una transacción o conjunto de transacciones.

5.5.4.2. Requerimientos

Para el desarrollo se hará uso de los siguientes paquetes de desarrollo, ya que brinda facilidad en la misma, y permite la conexión con otras herramientas de desarrollo de software:

- Sistema operativo: Windows, Linux o Mac OS X
- Node js versión >= 5.0
- Solidity versión 0.4.24
- Npm versión 5.x

5.5.4.3. Configuración del ambiente de desarrollo

Para el desarrollo del software sobre el Blockchain se proponen las siguientes herramientas de desarrollo:

- Truffle 4.1.14
- Ganache >= 1.2.2

De los cuales todos son pertenecientes al sistema de administrador de paquetes npm.

La herramienta Truffle permite interactuar con el Blockchain, provisto por la herramienta Ganache. Los archivos principales utilizados por esta herramienta son:

- Transporte.sol
- 2_deploy_contract.js

El archivo Transporte.sol contiene todo el código de programación del contrato inteligente o *smart contract*, y el archivo denominado 2_deploy_contract.js contiene el código de programación necesario para hacer el despliegue al Blockchain de desarrollo provisto por la herramienta Ganache, ver apéndice 6.

Del archivo Transporte.sol se pueden apreciar los siguientes aspectos de mayor relevancia. Se almacena la dirección de la empresa de transporte a la cual se le realiza él envió de activos; así mismo, el contrato almacena la cantidad de pagos o transacciones realizadas por el uso del servicio.

Otro aspecto que sobresale es el uso de un tipo de dato de tipo *struct*, el cual almacena para cada transacción, la cantidad de activos que el usuario tenía al momento de realizar la transacción, el piloto y el usuario que participaron en la transacción identificados por su dirección dentro de la red; y por último se encuentra un tipo, el cual hace referencia al periodo del día, el cual decide el tipo de cobro que se realiza al usuario.

De la modelación del caso de uso anterior se puede darse cuenta que no se cumplen todos los requisitos propuestos en el caso de uso de la red de transporte, esto se da debido a la naturaleza de esta tecnología, como punto primero para la modelación del pago se hace uso de la moneda nativa de la tecnología, la cual es el ether; como segundo punto en una red de este tipo no es posible hacer distinción entre los tipos de usuarios participantes en la red, ya que por su característica de tipo pública, permite el anonimato para la realización de transacciones dentro de la red; el código fuente de este proyecto se puede encontrar en la siguiente dirección web https://github.com/0722-EDD/transporte-publico-eth. A continuación, se muestra la comparativa en el desarrollo entre estas dos herramientas.

5.5.5. Comparativa: Hyperledger vs Ethereum

Luego de haber realizado la implementación del caso de uso con ambas tecnología se puede apreciar que existen distintas limitantes para ambas tecnologías, así como beneficios ligados a cada una de ellas, en la siguiente

tabla se hace una comparativa entre ambas tecnologías, tanto las utilizadas en este documento como lo son Fabric y Composer para Hyperledger y Ganache y Truffle para Ethereum, como las que van asociadas a ellas o ligadas internamente.

Tabla I. **Comparación Hyperledger vs Ethereum**

	Hyperledger	**Ethereum**
Seguridad	• Se puede limitar el acceso de los participantes a los recursos, mediante permisos	• Todos los participantes tienen permiso de realizar todas las operaciones
Sistema operativo	• Fabric está disponible para Mac OS X, *nix, y Windows • Composer está disponible únicamente para Ubuntu y Mac OS X	• Ganache y Truffle, están disponibles para los sistemas operativos Windows, Linux y Mac OS X • La red pública de Ethereum puede ser accedida mediante Windows, Linux y Mac
Economía	• Fabric y Composer son ambas herramientas de software libre y gratuitas	• Ganache y Truffle son ambas herramientas gratuitas • Los despliegues de contratos inteligentes y ejecución de transacciones se realizan mediante la criptomoneda ether y la cantidad de ether a utilizar depende del cómputo necesario en dichas operaciones, por lo que el costo incrementa cada vez que se realiza una transacción
Soporte	• Posee una comunidad bastante activa en Chat, así como una comunidad propia de la fundación Hyperledger	• Posee una comunidad bastante activa en Chat, Stack Exchange • Posee también una comunidad en redes sociales bastante amplia y

Continuación de la tabla I.

	• Poseen una cantidad de documentación en sus sitios oficiales bastante completa La cantidad de documentación tanto para Fabric y Composer disponible en internet es bastante pobre y muy especifica • Es posible formar parte de la organización como miembro corporativo • Está respaldada por empresas de renombre a nivel mundial	activa • Posee una documentación en sus sitios oficiales bastante completa Aunque la cantidad de documentación es bastante amplia existen demasiadas herramientas que pueden ser utilizadas para el desarrollo, así como clientes para conectar a la red lo cual resulta un poco abrumador
Compatibilidad	• Permite ampliar su funcionalidad debido a la alta capacidad de modularidad • Permite el desarrollo de aplicaciones para interactuar con los contratos inteligentes a través del uso de otros lenguajes de programación • Provee un conjunto de herramientas propias del proyecto Hyperledger que brindan capacidades adicionales sobre los distintos Blockchain • El objetivo del proyecto Hyperledger es el de crear estándares propios de tecnologías de este tipo • Brinda integración con	• No es modular • Permite el desarrollo de aplicaciones para interactuar con los contratos inteligentes a través del uso de otros lenguajes de programación • Posee una gran gama de herramientas de desarrollo distintas a las utilizadas en la implementación en este documento. • Brinda integración con software para la realización de pruebas

Continuación de la tabla I.

	software para la realización de pruebas	
Tipo de Blockchain	• Los distintos *frameworks* de Hyperledger permiten establecer blockchain de tipo permisivo, así como de tipo público	• Es un Blockchain de tipo público, por lo que permite la publicación de toda la información asociada al sistema. • Está basado en criptomoneda
Usabilidad	• Composer utiliza un lenguaje similar al lenguaje de programación Javascript, el cual es limitado y de fácil aprendizaje • Brinda aplicaciones interactivas para poder definir la lógica de negocio y probar dichas definiciones • La interacción de Fabric y Composer se realiza principalmente por medio de línea de comandos	• Utiliza un lenguaje similar a lenguaje de programación Javascript, el cuales limitado y de fácil aprendizaje • Aunque el lenguaje que utiliza es bastante simple, requiere conocimientos avanzados de programación y de su máquina virtual. • Así como Hyperledger también brinda aplicaciones interactivas para poder definir la lógica de negocio y probar dichas definiciones, además permiten determinar el consumo de ether en la aplicación • Ganache brinda una interfaz gráfica bastante simple de utilizar
Algoritmos de consenso	• Provee el uso de distintos algoritmos de consenso para las distintas tecnologías que provee	• El algoritmo de consenso principal es *proof of work* • En la actualidad está bajo desarrollo un nuevo algoritmo de consenso, el cual reemplazara al algoritmo *proof of work*

Fuente: elaboración propia.

De la comparación entre las tecnologías Hyperledger y Ethereum de la tabla I, se puede apreciar que dichas tecnologías tienen un enfoque distinto; estas pueden adaptarse a distintos tipos de escenarios; entre las principales diferencias que hay entre ellas es el hecho que Ethereum es principalmente una tecnología Blockchain de tipo público, en la cual, cualquiera puede ser parte de ella y toda la información es visible para todos; aunque dentro de esta red es utilizado el anonimato, existen casos de uso en los cuales este no sea el mejor enfoque, en comparación a las tecnologías de Hyperledger, la cual permite limitar el acceso dentro de la red, tanto a los participantes como a las transacciones.

Otro aspecto muy importante entre estas tecnologías es el uso de criptomonedas, de las cuales para Ethereum es obligatorio el uso de estas, para las tecnologías de Hyperledger no, lo cual puede limitar a ambas en distintas aplicaciones o casos de uso, para lo cual se debe de analizar la viabilidad de la implementación de ambas tecnologías.

En otro aspecto se puede apreciar la compatibilidad, usabilidad y soporte, aunque ambas tecnologías permiten la integración con otras tecnologías de software, Ethereum se ve limitada al alcance que esta puede tener sobre la propia tecnología, en comparación a las tecnologías de Hyperledger las cuales son altamente modulares; sin embargo, ambas tienen comunidades muy activas y amplias en las cuales es posible buscar una ayuda adicional.

CONCLUSIONES

1. Un complemento del estudio de las estructuras de datos es la tecnología Blockchain, debido a su íntima relación con las ya mencionadas estructuras de datos. Sin embargo, no es posible el estudio de esta tecnología sin tener previo conocimiento, de conceptos como estructuras de datos lineales y no lineales, funciones hash y de ciertos enfoques de la criptografía, por lo que se debe tener una secuencia establecida para su enseñanza.

2. Las aplicaciones de Blockchain en la actualidad son limitadas en cuanto a los mercados y casos de uso en los que se aplica, debido a que implica un gran cambio en la forma en se maneja la información y en la que trabajan las industrias; sin embargo, las ventajas que Blockchain brinda, permite que nuevos mercados sean explorados dando paso a nuevos paradigmas, tecnologías y formas de aplicación de Blockchain, siendo la principal, la veracidad de los bienes o valores que se intercambian en distintas transacciones.

3. La secuencia determinada para la enseñanza de Blockchain consta de cinco fases las cuales abarcan conceptos de estructuras de datos, aplicaciones de las estructuras de datos, sus variantes y criptografía, las cuales permiten comprender de una manera en la que toma sentido la aplicación que hace Blockchain de las estructuras de datos y criptografía.

4. De la comparación de las herramientas utilizadas que son: Hyperledger y Ethereum, se llegó a la conclusión de que la herramienta que ayuda de una mejor manera en la docencia de las estructuras de datos y Blockchain es Hyperledger, debido a la facilidad que requiere la programación, el alcance que esta provee en comparación a Ethereum; además, permite explorar más conceptos de Blockchain y en general este tipo de Blockchain que no es basado en criptomoneda, tiene un mayor alcance que aquellas que sí lo son.

RECOMENDACIONES

1. Para la enseñanza tanto de la tecnología Blockchain, como de las estructuras de datos, utilizar software para la ejemplificación de los distintos conceptos, debido a que de esta manera es posible poner en práctica los distintos conceptos aprendidos por los estudiantes.

2. Aumentar el número de conceptos del flujo de enseñanza y actualizar constantemente las tecnologías que forman parte de la labor de docencia.

3. Proponer nuevos casos de uso y aplicaciones de la tecnología Blockchain, para el estudio de su factibilidad e implementación.

4. Investigar otras herramientas Blockchain que no sean basadas en criptomonedas, que ayuden de manera similar a un aprendizaje amplio de la misma y con las mismas facilidades de uso.

BIBLIOGRAFÍA

1. BASHIR, Imran. *Mastering Blockchain*. 2a ed. Reino Unido: Packt Publishing Ltd., 2018. 1 023 p.

2. Blockchainhub. *Blockchains & Distributed Ledger Technologies.* [en línea]. <https://blockchainhub.net/blockchains-and-distributed-ledger-technologies-in-general/>. [Consulta: 7 de agosto de 2018].

3. BRASS, Peter. *Advanced Data Structures*. Estados Unidos: Cambridge University Press, 2008. 456 p.

4. Data Flair. *3 Different Types of Blockchain Technology.* [en línea]. <https://data-flair.training/blogs/types-of-blockchain/>. [Consulta: 1 de septiembre de 2018].

5. Devopedia. *Types of Blockchains.* [en línea]. <https://devopedia.org/types-of-blockchains>. [Consulta: 1 de septiembre de 2018].

6. DRESCHER, Daniel. *Blockchain Basics A Non-Technical Introduction in 25 Steps*. Alemania: Apress, 2017. 388 p.

7. EDX. *Blockchain for Business – An Introduction to Hyperledger Technologies*. [en línea]. <https://courses.edx.org/courses/course-v1:LinuxFoundationX+LFS171x+3T2017/course/>. [Consulta: 4 de agosto de 2018].

8. GILBERG F, Richard. *Data Structures A Pseudocode Approach with C.* 2da ed. Estados Unidos: Thomson Learning, 2005. 46 p.

9. Hyperledger. *About Hyperledger* [en línea]. <https://www.hyperledger.org/about>. [Consulta: 12 de agosto de 2018].

10. JAMES, Febin. *3 Popular Types of Blockchains You Need To Know.* [en línea]. <https://hackernoon.com/3-popular-types-of-blockchains-you-need-to-know-7a5b98ee545a>. [Consulta: 1 de septiembre de 2018].

11. JOHNSON, Cynthia. *5 industrias que probablemente serán sacudidas por el blockchain.* [en línea]. <https://www.entrepreneur.com/article/315367>. [Consulta: 15 de septiembre de 2018].

12. JOYANES, Luis; SANCHEZ, Lucas; ZAHONERO, Ignacio. *Estructura de datos en C++*. España: McGraw-Hill, 2007. 611 p.

13. KARUMANCHI, Narasimha. *Data Structures and Algorithms Made Easy.* 5a ed. India: Career Monk Publications, 2017. 828 p.

14. KHARIF, Olga, LEISING, Matthew. *Bitcoin and Blockchain*. Bloomberg. [en línea]. <https://www.bloomberg.com/quicktake/bitcoins>. [Consulta: 4 de noviembre de 2018].

15. KHATWANI, Sudhir. *Different Types of Blockchains In The Market and Why We Need Them.* [en línea]. <https://coinsutra.com/different-types-blockchains/>. [Consulta: 20 de septiembre de 2018].

16. MARR, Bernard. *A Very Brief History of Blockchain Technology Everyone Should Read.* [en línea]. <https://www.forbes.com/sites/bernardmarr/2018/02/16/a-very-brief-history-of-blockchain-technology-everyone-should-read/>. [Consulta: 12 de agosto de 2018].

17. _____. *A Short History of Bitcoin and Crypto Currency Everyone Should Read*. [en línea]. <https://www.forbes.com/sites/bernardmarr/2017/12/06/a-short-history-of-bitcoin-and-crypto-currency-everyone-should-read>. [Consulta: 12 de agosto de 2018].

18. MILLS, Brad. *What is Cryptocurrency: Everything You Must Need To Know!* [en línea]. <https://blockgeeks.com/guides/what-is-cryptocurrency/>. [Consulta: 20 de septiembre de 2018].

19. NAKAMOTO, Satoshi. *Bitcoin: A Peer-to-Peer Electronic Cash System*. [en línea]. <https://bitcoin.org/bitcoin.pdf>. [Consulta: 4 de agosto de 2018].

20. PAAR, Christof, PELZL, Jan. *Understanding Cryptography A Textbook for Students and Practitioners*. Alemania: Springer, 2009. 372 p.

21. SIVARAM, Varun. *Blockchain and Energy: We Sifted Hype from Reality So You Don't Have To.* [en línea]. <https://www.cfr.org/blog/blockchain-and-energy-we-sifted-hype-reality-so-you-dont-have>. [Consulta: 15 de septiembre de 2018].

22. Upfolio. *Cryptocurrency Explained* [en línea]. <https://www.upfolio.com/ultimate-cryptocurrency-guide>. [Consulta: 12 de agosto de 2018].

APÉNDICES

Apéndice 1. **Archivo Transaction.java**

```java
public class Transaction {

    private final String    origen;
    private final String    destino;
    private final String    valor;
    private final long      timestamp;

    public Transaction(String origen, String destino, String valor){
        this.origen     =   origen;
        this.destino    =   destino;
        this.valor      =   valor;
        this.timestamp  =   System.currentTimeMillis();
    }

    public String getOrigen() {
        return origen;
    }

    public String getDestino() {
        return destino;
    }

    public String getValor() {
        return valor;
    }

    public String getTimestamp() {
        SimpleDateFormat sdf = new SimpleDateFormat("MMM dd,yyyy HH:mm:ss");
        Date date = new Date(this.timestamp);
        return sdf.format(date);
    }

}
```

Fuente: elaboración propia.

Apéndice 2. **Archivo Block.java**

```
public class Block {

    private final int           id;
    private final Block         previo;

    private final Transaction   transacciones[];

    private String              merkleTreeRoot;
    private final Long          timestamp;
    private final String        hashPrevio;
    private final String        hash;

    public Block(Block previo, String hashPrevio, int id, Transaction[] transacciones) {

        this.id             =   id;
        this.previo         =   previo;

        this.transacciones  =   transacciones;

        this.merkleTreeRoot =   crearMerkleRoot();
        this.timestamp      =   System.currentTimeMillis();
        this.hashPrevio     =   hashPrevio;
        this.hash           =   obtenerHash();

    }

    public Block(int id, Transaction[] transacciones) {

        this.id             =   id;
        this.previo         =   null;

        this.transacciones  =   transacciones;

        this.merkleTreeRoot =   crearMerkleRoot();
        this.timestamp      =   System.currentTimeMillis();
        this.hashPrevio     =   "";
        this.hash           =   obtenerHash();

    }

    private String obtenerHash(){
        return generarHash(getTimestamp() + getHashPrevio() + this.getMerkleTreeRoot());
    }
```

Continuación del apéndice 2.

```
    private String crearMerkleRoot(){
        String transaccion1 =   transacciones[0].getDestino() + transacciones[0].getOrigen()
                                + transacciones[0].getValor() + transacciones[0].getTimestamp();
        String transaccion2 =   transacciones[1].getDestino() + transacciones[1].getOrigen()
                                + transacciones[1].getValor() + transacciones[1].getTimestamp();
        String transaccion3 =   transacciones[2].getDestino() + transacciones[2].getOrigen()
                                + transacciones[2].getValor() + transacciones[2].getTimestamp();
        String transaccion4 =   transacciones[3].getDestino() + transacciones[3].getOrigen()
                                + transacciones[3].getValor() + transacciones[3].getTimestamp();

        String  hash_tx1    =   generarHash(transaccion1);
        String  hash_tx2    =   generarHash(transaccion2);
        String  hash_tx3    =   generarHash(transaccion3);
        String  hash_tx4    =   generarHash(transaccion4);

        String  hash_tx1_tx2    =   generarHash( hash_tx1 + hash_tx2 );
        String  hash_tx3_tx4    =   generarHash( hash_tx3 + hash_tx4 );

        return generarHash( hash_tx1_tx2 + hash_tx3_tx4 );
    }

    public String generarHash(String str) {

        try {

            MessageDigest digest = MessageDigest.getInstance("SHA-256");
            byte[] hash = digest.digest(str.getBytes(StandardCharsets.UTF_8));
            StringBuffer hexString = new StringBuffer();
            for (int i = 0; i < hash.length; i++) {
                String hex = Integer.toHexString(0xff & hash[i]);
                if(hex.length() == 1) hexString.append('0');
                hexString.append(hex);
            }

            return hexString.toString();

        } catch (NoSuchAlgorithmException ex) {
            Logger.getLogger(Block.class.getName()).log(Level.SEVERE, null, ex);
        }

        return "";
    }
```

```
    public int getId() {
        return id;
    }

    public Block getPrevio() {
        return previo;
    }

    public Transaction[] getTransacciones() {
        return transacciones;
    }

    public String getHashPrevio() {
        return hashPrevio;
    }

    public String getHash() {
        return hash;
    }

    public long getTimestamp() {
        return timestamp;
    }

    public String getMerkleTreeRoot() {
        return merkleTreeRoot;
    }

}
```

Fuente: elaboración propia.

Apéndice 3. **Archivo Blockchain.java**

```
public class BlockChain {

    private Block   genesis;
    private Block   ultimo;
    private int     numeroDeBloques;

    public BlockChain() {
        this.genesis            =   null;
        this.numeroDeBloques    =   0;
    }

    public void agregarBloque(Transaction[] transacciones) {
        if(genesis  !=  null){
            Block nuevoBloque = new Block(this.ultimo, this.ultimo.getHash(), numeroDeBloques++, transacciones);
            this.ultimo = nuevoBloque;
        }else{
            this.genesis = new Block(numeroDeBloques++, transacciones);
            this.ultimo =  this.genesis;
        }
    }

    public int getNumeroDeBloques() {
        return numeroDeBloques;
    }

    public void obtenerBloque(int id){
        Graficas    graph   =   new Graficas(ultimo,id);
    }

    public void mostrar(){
        Graficas    graph   =   new Graficas(ultimo);
    }

}
```

Fuente: elaboración propia.

Apéndice 4. **Archivo org.usac.transporte.cto**

```
namespace org.usac.transporte

abstract participant Persona identified by id {
  o String id
  o String nombres
  o String apellidos
  o Integer edad
  o Boolean descuento optional
}

participant Usuario extends Persona {
  o Double saldo
  o String numeroTarjeta
}

participant Piloto extends Persona {
  o String tipoLicencia
  o String Licencia
  o Integer numeroTransacciones
  o Integer numeroAccidentes
}

asset Autobus identified by placa {
  o String placa
  --> Piloto piloto
}

enum PeriodoDelDia {
  o DIA
  o NOCHE
}

transaction Pago {
  --> Autobus autobus
  --> Usuario usuario
  o Double saldoActual
  o PeriodoDelDia periodo
}
```

Fuente: elaboración propia.

Apéndice 5. **Archivo logic.js**

```
/**
 * Funcion para realizar una transaccion = usuario de bus utiliza el transporte
 * @param {org.usac.transporte.Pago} pago Instancia de un pago de servicio
 * @transaction
 */
function pago(pago){
  var factory = getFactory();
  var NS = 'org.usac.transporte';

  var total = 0.0;
  var descuento = 0.0;
  var cuotaDia = 1.50;
  var cuotaNoche = 2.00;

  var usuario = pago.usuario;
  var periodo = pago.periodo;
  var piloto  = pago.autobus.piloto;

  if(usuario.edad <= 18 || usuario.edad >= 60){
    total = 0.0;
  }else{
    if(periodo == "DIA"){
      total = cuotaDia;
    }else{
      total = cuotaNoche;
    }
  }

  usuario.saldo -= total;
  piloto.numeroTransacciones += 1;

  return getParticipantRegistry(NS + '.Usuario')
      .then(function(registroUsuarios){
        return registroUsuarios.update(usuario);
      })
      .then(function(){
        return getParticipantRegistry(NS + '.Piloto')
      })
      .then(function(registroPilotos){
        return registroPilotos.update(piloto);
      })
}
```

Fuente: elaboración propia.

Apéndice 6. **Archivo permission.acl**

```
rule NetworkAdminUser {
    description: "Grant business network administrators full access to user resources"
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "**"
    action: ALLOW
}

rule NetworkAdminSystem {
    description: "Grant business network administrators full access to system resources"
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "org.hyperledger.composer.system.**"
    action: ALLOW
}
```

Fuente: elaboración propia.

Apéndice 7. **Archivo transporte.sol**

```
pragma solidity ^0.2.24

contract Transporte {

    address public empresa;
    uint public cantidadPagos;

    struct Pago {
        address piloto;
        address usuario;
        uint userBalance;
        uint tipo;   //  1   =   dia     2   =   noche
    }

    mapping(uint => Pago) public pagos;

    constructor() public {
        empresa = msg.sender;
        cantidadPagos = 0;
    }

    function pagar(address _piloto, uint _tipo) public payable {

        cantidadPagos++;
        pagos[cantidadPagos] = Pago(_piloto, msg.sender, msg.sender.balance, _tipo);

        if(_tipo == 1){
            msg.sender.transfer(msg.value);
        }else{
            empresa.transfer(msg.value);
        }
    }

    function getBalance() public view returns (uint) {
        return address(this).balance;
    }

}
```

Fuente: elaboración propia.