CURSO
Introducción a Blockchain

MÓDULO 3
“Implementando una Blockchain”

Contenido

1. OBJETIVOS DEL MÓDULO

En este módulo revisaremos los pasos necesarios para definir la arquitectura y el diseño de la blockchain.

Analizaremos las diferentes herramientas disponibles para crear nuestras cadenas de bloque, realizando una comparación entre las ventajas de las mismas.

2. INTRODUCCIÓN

2.1 Determinar cómo puede ayudar blockchain en el proceso

Para descubrir si blockchain puede ayudar en nuestro proceso de negocio o introducir cambios realmente beneficiosos a las tareas que ya están automatizadas, podemos tratar de responder a las siguientes preguntas:

- ¿Mi red de negocios necesita gestionar relaciones contractuales?
- ¿Es necesario realizar un seguimiento a aquellas transacciones que involucran a más de dos partes?
- ¿El sistema actual es muy complejo y/o costoso a causa de la necesidad de intermediarios o de un punto central de control?
- ¿Se verá beneficiada la red de negocio por la confianza y transparencia en el mantenimiento de registros?
- ¿El sistema actual es propenso a errores debido a procesos manuales o esfuerzos de duplicación?
- ¿El sistema actual de transacciones es vulnerable a fraude, ciberataque o error humano?

3. IMPLEMENTANDO UNA BLOCKCHAIN

3.1 Determinar el objetivo de tu red blockchain

Si la respuesta a alguna de las preguntas anteriores es afirmativa, entonces una excelente opción es usar tecnología blockchain. También es recomendable hacer un esfuerzo más por pensar qué atributos de la blockchain son los que potenciarán la solución. Por ejemplo, si la pérdida de confianza está causando fricciones en la red de negocios, la característica de registros compartidos de blockchain incrementará la visibilidad de las transacciones, mientras que el historial de activos mejorará la confianza. Por otro lado, si las causas de fricciones en la red son las demoras en los acuerdos o reglas de negocios, los contratos inteligentes serán la solución.

Es importante tener en claro un objetivo que se pueda medir fácilmente para llevar a cabo el primer proyecto blockchain.

¿Qué es lo que estamos buscando resolver o mejorar con el uso de tecnología blockchain? ¿Qué podemos utilizar para medir el éxito de este primer proyecto en el cumplimiento de ese objetivo?

3.2 Elegir un caso de uso similar

Es recomendable seleccionar casos de uso similares. Los sitios web de plataformas blockchain albergan ejemplos que pueden usarse como punto de partida para implementar la propia blockchain.

Al seleccionar un caso de uso, podríamos preguntarnos:

Respecto del consenso: ¿El acuerdo de que cada transacción es válida en la red de negocios, proporciona algún beneficio?

¿Es importante el mantenimiento de una pista de auditoría informática?

En relación a la inmutabilidad: ¿Es importante que el tren de transacciones sea evidente? ¿Existe la necesidad de un "sistema de registro" acordado en toda la red de negocios?

3.3 Los pasos para implementar una blockchain

3.3.1 Del análisis a la implementación

Una vez definido el objetivo de la cadena de bloques que queremos construir, es aconsejable seguir una serie de pasos que conformarán la metodología aconsejada en este curso. Los pasos a seguir son:

1. Definir a los participantes.
2. Definir activos y transacciones posibles.
3. Desarrollar una lógica de transacciones.
4. Definir reglas de control de acceso.
5. Construir la red de negocios.
6. Implementar sobre una plataforma blockchain.
7. Realizar una prueba de conceptos.
8. Realizar el pase a producción.

3.3.2 Identificar la red de negocios y las dependencias

En primer lugar debemos tener en claro con quién compartiremos la blockchain, a quién le es útil el registro de las transacciones de nuestra actividad. Entonces definiremos nuestra red de negocios.

Por otro lado, en caso de no contar con un grupo de desarrollo de software (programadores) tendremos que salir a contratar personal o un socio comercial para poder realizar la implementación.

Además debemos pensar si es necesario contar con algún tipo de estructura que permita cumplir con ciertos objetivos regulatorios.

Dado que el procesamiento de transacciones se ha convertido en una tarea en equipo, una red blockchain es más aprovechable y satisfactoria cuando existen múltiples partes involucradas en el proceso. Esto aumenta el valor y la eficiencia de la blockchain.

> Las organizaciones necesitan aprender un nuevo modelo de procesos basado en "ecosistemas", y así es como funciona blockchain.

3.3.3 Actores y activos

Una vez identificada la red de negocios, listaremos los actores (participantes) de la misma y vamos a pensar qué rol tienen: pueden realizar transacciones? solo necesitan "leer" información?

También (a los ojos de los programadores) será necesario modelar los datos asociados a cada actor. Por ejemplo, para el escenario de la venta de hongos de pino, los datos sobre el vendedor serán: identificador, nombre y apellido, número de registro de habilitación municipal, domicilio, teléfono. En cambio para los compradores solo necesitaríamos datos para realizar la facturación sin importar su inscripción municipal.

De la misma manera realizaremos el modelado de datos de los activos que maneja la red. Para el ejemplo mencionado, nuestros activos o *asset*s serán los hongos de pino, ahora bien los datos que queremos registrar en relación al activo son: quién los vende, si posee certificado de semillas, cuándo fueron sembrados, cuánto cuesta el kg.

Una vez finalizados estos pasos, empezaremos a pensar en las transacciones que se registrarán en la red.

3.3.4 Definición de transacciones

Se sugiere realizar un listado de las transacciones que tendrán lugar en la red y serán registradas en la blockchain. Continuando con el escenario de la venta de hongos de pino, se listan las transacciones posibles:

1) Cargar cosecha

2) Realizar una venta

Por cada transacción, vamos a estructurar el intercambio de información necesario. Así por ejemplo la 1) requerirá del nombre de quien cosecha, que automáticamente se convierte en un potencial vendedor, la fecha de cosecha y si poseen o no certificado de autenticidad los hongos cosechados.

Mientras tanto, para la transacción 2) el proceso de venta requerirá de nombre y apellido del comprador, nombre y apellido del vendedor, fecha de venta, cantidad, precio y posiblemente, descuento.

Ahora sí, una vez plasmado el diseño de nuestra blockchain será necesario seleccionar una herramienta para su implementación, obviamente considerando todas las características estudiadas en este curso.

3.3.5 Definición de la plataforma

Una vez que tenemos en claro quiénes son los participantes de nuestra red de negocios y cuáles son las características deseadas que debería tener nuestra cadena de bloques, entonces es momento de optar por una herramienta de implementación.

Cuando hablamos de herramienta de implementación básicamente nos referimos a Ethereum, Hyperledger, Corda o Iota. Existen otras en el mercado pero por ahora, éstas son las más usadas y consolidadas.

En las páginas siguientes revisaremos los pros y contra de cada una de estas tecnologías.

3.4 Etherum

Ethereum es una plataforma blockchain descentralizada que permite construir aplicaciones, donde el "Ether" es la criptomoneda o token usada para intercambiar valor y pagar por el uso de recursos en la red.

Ethereum permite escribir contratos inteligentes y aplicaciones descentralizadas que son accesibles a todo el mundo.

Recordemos que los contratos inteligentes o smart contracts permiten definir la lógica de negocio. En Ethereum pueden ser programados en lenguajes de alto nivel tales como Solidity, Viper, LLL, Serpent o Mutant. Pero esta explicación implicaría otro curso. Así que para los curiosos o quienes tengan conocimientos más avanzados de informática, los invitamos a ingresar a esta plataforma: https://remix.ethereum.org

3.4.1 La máquina virtual de Ethereum (EVM)

En el centro de Ethereum está su máquina virtual (Ethereum Virtual achine - EVM) que podemos describir como un procesador universal en el que las transacciones se ejecutan a cambio de una pequeña comisión pagada en Ethers.

3.5 Hyperledger

Hyperledger es una plataforma blockchain de código libre creado por **The Linux Foundation** con el objetivo de crear una solución blockchain que permita reducir costos y complejidad a la hora de realizar transacciones u operaciones.

El proyecto fue lanzado en 2016 con una estructura de gobernanza técnica y organizacional definida y un total de 30 miembros fundadores. El proyecto se inició con dos frameworks: **Hyperledger Fabric** y **Hyperledger Sawtooth**, iniciados por IBM e Intel respectivamente.

Todos los frameworks de Hyperledger incluyen:

- Un registro distribuido inmutable.
- Un algoritmo de consenso para decidir el estado del registro.
- Privacidad de transacciones y acceso permisionado.
- Smart contracts para la lógica de negocio.

Hyperledger, a diferencia de redes como Bitcoin y Ethereum, es una red blockchain permisionada.

Hyperledger implementa la privacidad de transacciones a través de lo que se conoce como canales (channels), una especie de subredes de participantes dentro de la propia red.

En definitiva, Hyperledger provee características clave de una blockchain como registro único, inmutabilidad y robustez a la vez que incluye características necesarias en un entorno empresarial como escalabilidad y privacidad.

- **Frameworks de Hyperledger:** Iroha, Sawtooth, Fabric, Indy y Burrow
- **Módulos de Hyperledger:** Cello, Explorer y Composer

3.6 Hyperledger Fabric

3.6.1 ¿Qué es Hyperledger Fabric?

Es una implementación de un framework blockchain y uno de los proyectos dentro de Hyperledger. Se caracteriza por tener una arquitectura modular y altos niveles de confidencialidad, resiliencia, flexibilidad y escalabilidad. Como otras plataformas blockchain, tiene un registro compartido, transacciones y smart contracts, que reciben el nombre de chaincode en el caso de Hyperledger Fabric. A diferencia de Ethereum, Hyperledger Fabric es una blockchain permisionada y los participantes de la red deben unirse a través del Membership Service Provider (MSP) o Proveedor de Servicio de Membresía.

3.6.2 Membership Service Provider (MSP)

El MSP es un componente que define las reglas para validar, autenticar y permitir el acceso a la red a una identidad o participante. El MSP usa Certificate Authority (CA) y el interfaz por defecto es Fabric-CA API. Este componente es fácilmente reemplazable, lo que hace a Hyperledger Fabric muy flexible a la hora de usar un mecanismo de identificación u otro.

3.6.3 Roles

Hay tres tipos de roles en una red de Fabric:

- **Clients** (clientes): son aplicaciones que actúan en nombre de una persona a la hora de proponer transacciones, es decir, permite a los usuarios finales la comunicación con la blockchain.
- **Peers**: mantienen el estado de la red y una copia del registro. Dentro de este rol existen dos tipos de peers:
    - **Endorsers** (“avalista” o “patrocinador”): simulan y avalan transacciones propuestas.

- **Committers** ("grabadores"): verifican las transacciones propuestas y validan el resultado de las mismas antes de grabarlas en la blockchain.

- **Ordering Service** (servicio de ordenación): recibe las transacciones propuestas, las ordena dentro de un bloque que lo transmite a los commiters.

3.7 Corda

Corda es un producto blockchain más orientado a la industria financiera en lugar de cubrir todo tipo de aplicaciones. Pertenece al grupo empresario R3 y se dedica específicamente al sector bancario. Sin embargo, R3 Corda también se está volviendo popular en otros nichos como seguros médicos y cadenas de transporte. R3 Corda es una plataforma de código abierto (como Hyperledger) que permite implementar blockchains de tipo permisionadas

Lo que la hace única es su manera de alcanzar el consenso. A diferencia de otras DLT el consenso se hace entre las partes involucradas y un tercero denominado Notario. Las transacciones se ejecutan en dos fases:

1. Se definen los actores que intervienen en el contrato inteligente. Entre ellos se reconocen y validan las firmas digitales con las que se realizan las transacciones. Una vez comprobada la firma, se ejecuta el contrato.

2. Antes de dar por finalizado el contrato, un nodo, que no intervino en dicho contrato, hace de testigo comprobando que el contrato no se ejecutó anteriormente o, por ejemplo, que existen fondos para la transacción bancaria.

Si este nodo llamado notario, da fe de la transacción, la correcta ejecución de la misma queda grabada en la cadena de bloques.

3.7.1 Consenso

El consenso en Corda se realiza sobre una única transacción también en dos pasos diferenciados por las partes del contrato y el notario:

1. *Validity consensus*

En el primer paso, cada integrante del contrato verifica que existan las firmas digitales de los demás participantes del contrato (como ya se expresó, antes de aprobar la transacción).

2. *Uniquensess consensus*

El notario verifica que no exista un doble envío de transacción, tal como se muestra en la imagen de abajo. Si este fuera el caso, podría ocurrir que una cuenta se quedase sin fondos y no pudiese enviar el dinero. Por ejemplo: Ana está por enviar dinero a Beto y a su vez Beto a Carlos y a David. El notario verifica que el dinero que Beto promete a Carlos y David no supere al ingreso que éste tendrá cuando Ana le pague.

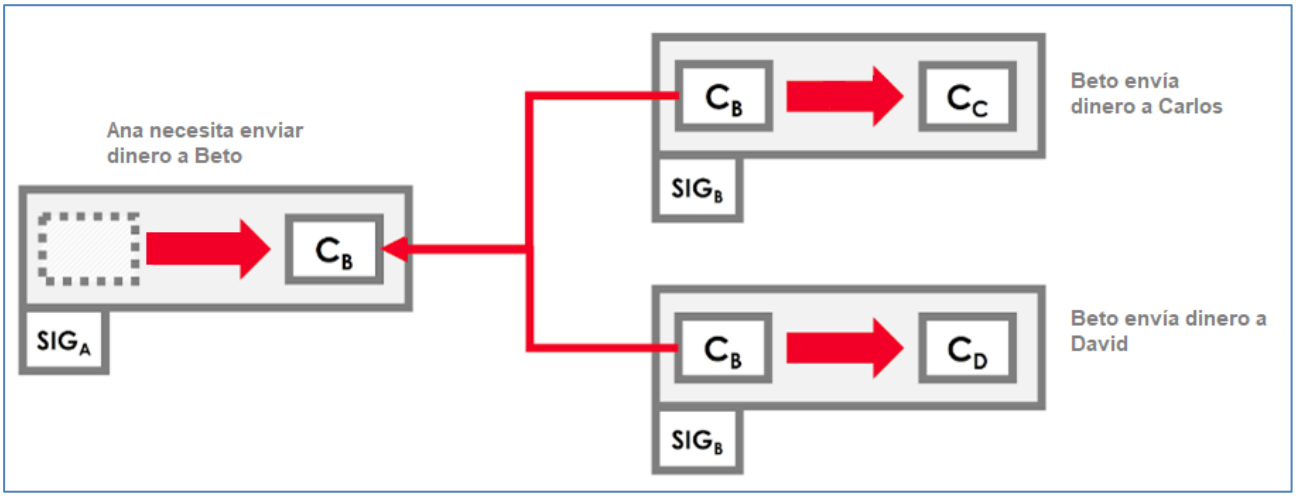

Fig. 1 Transacciones en CORDA.[1]

3.8 IOTA

La criptomoneda IOTA existe desde el año 2015. Según Martin Rosulek, "es la primera criptomoneda que proporciona todo el ecosistema basado en blockchain sin bloques" para permitir transacciones de máquina a máquina (M2M).

IOTA, sin embargo, es más que solo una criptomoneda. Esencialmente, la plataforma implica una generalización del protocolo blockchain (la tecnología llamada Tangle) que se encuentra en el backend de la plataforma IOTA.

En lugar de pagar a los mineros para validar las transacciones, la arquitectura de la red implica una validación basada en pares.

Podemos pensar en una analogía simple, la de un maestro que califica la tarea de los estudiantes: los estudiantes son los clientes/ usuarios en el protocolo de Bitcoin y el maestro es el minero/ validador. La tecnología Tangle les pide a los estudiantes (usuarios) que califiquen la tarea de los demás, haciendo superflua la necesidad de un maestro (validador externo) y evitando los gastos relacionados con el trabajo del maestro/ validador. Esto permite que la plataforma sea completamente gratuita, sin enfrentar los desafíos de escala inherentes a la primera generación de blockchain.

Además, el uso de la plataforma con dispositivos conectados o IoT (Internet de las cosas): "Permite a las empresas explorar nuevos modelos de empresa a empresa al hacer que cada recurso tecnológico sea un servicio potencial para ser comercializado en un mercado abierto en tiempo real, sin comisiones" Roger Aitken, 2017.

[1] Extraída y traducida de la página de CORDA.org

4. Comparación de herramientas

4.1 Características principales de cada blockchain

A continuación se muestra un cuadro comparativo de las principales plataformas blockchain según sus características.

	Bitcoin	Ethereum	Hyperledger Framworks
Basado en Criptomonedas	Si	Si	No
Permisionada	No	No	Si (en general)
Pseudo-anónimo	Si	No	No
Auditable	Si	Si	Si
Registro Inmutable	Si	Si	Si
Modularidad	No	No	Si
Smart Contracts	No	Si	Si
Protocolo de Consenso	PoW	PoW	Varios

Fig. 2 Cuadro comparativo Bitcoin, Etherum e Hyperledger

En las redes de Bitcoin y Ethereum podemos ver los bloques y transacciones con sus campos prácticamente en tiempo real a través de exploradores como:

Bitcoin: https://blockchain.info/

Ethereum: https://etherscan.io/

4.2 Mecanismos específicos de inmutabilidad

En Ethereum e Hyperledger, el mecanismo de inmutabilidad se basa en que cada transacción y bloque se firma digitalmente y se vinculan mediante hashes criptográficos.

Corda, por otro lado, confía en su "servicio notarial" para la inmutabilidad. Cada red Corda posee uno o más servicios notariales que verifican las transacciones por separado y, si se aprueban, son firmadas por el servicio notarial. Después de que las transacciones firmadas por un notario finalizan, ya no pueden modificarse.

4.3 Uso de criptografía

El uso de claves públicas para la gestión de identidad es una opción lógica, ya que el conocimiento de una clave pública es necesario para la verificación de firmas digitales. Tanto Ethereum como Hyperledger Fabric utilizan firmas digitales en transacciones y bloques para verificar la identidad del creador y que los datos firmados no se han modificado desde la firma. La criptografía de clave pública se usa en blockchain como un método para administrar las identidades de los usuarios sin revelar identidades del mundo real.

- En Ethereum, los usuarios se identifican por una dirección que está directamente relacionada con la clave pública del usuario. Esto proporciona verificación de identidad mientras se preserva el anonimato.
- En Hyperledger Fabric, los usuarios se identifican mediante certificados X.509. Estos certificados proporcionan varios datos sobre el usuario, pero uno de ellos también es la clave pública del usuario.

4.4 Funciones Hash

Cómo vimos en el primer módulo, las funciones hash son el núcleo de toda la tecnología blockchain. Tanto en Ethereum como en Hyperledger Fabric, los bloques incluyen el hash del bloque anterior para unir la cadena de bloques en un todo cohesivo.

Tanto Ethereum como Hyperledger Fabric son plataformas de contrato inteligentes que usan un tipo particular de árbol Merkle llamado árbol "Patricia" para almacenar el estado actual de su máquina virtual.

Las funciones hash se utilizan como el rompecabezas criptográfico en el centro del algoritmo de consenso de Prueba de trabajo (Proof of Work).

Ethereum actualmente usa Prueba de trabajo para el consenso, aunque se ha incorporado un cambio a Prueba de estaca (Proof of Stake).

Mientras que en Hyperledger Fabric solo hay dos algoritmos de consenso implementados: SOLO y Kafka. SOLO es para desarrollo y Kafka es para producción.

4.5 Incentivos o denegación de acceso

Una de las principales diferencias entre Ethereum e Hyperledger es la manera en que queda fuera de la red un participante que actuó malintencionadamente.

Dado que redes como Bitcoin y Ethereum manejan el concepto de incentivos, un actor malintencionado será descubierto dado que se penalizan las acciones que cometa en contra de la lógica de negocios de la red.

En contraposición, en Hyperledger el concepto de incentivos no tiene sentido, dado que al ser una red permisionada se reduce el riesgo dejando unirse a la red solo a participantes conocidos. Si se descubre que un participante está actuando de forma malintencionada, se le denegará el acceso mediante un consenso entre el resto de los participantes.

4.6 Desarrollo de los Smart Contracts

4.6.1 Smart contracts en Ethereum

Un ejemplo hipotético de un contrato inteligente basado en Ethereum puede involucrar la siguiente transacción: transferir el monto X del inversor a la compañía al recibir las acciones dadas de la compañía después de un incremento de capital. El monto inteligente X, que fue previamente validado por la compañía para la transacción (al igual que en una compra con tarjeta de crédito), se mantiene en custodia por el contrato inteligente, hasta que el inversionista haya recibido las acciones. La cadena de bloques Ethereum solo codifica estas "reglas de juego". Los beneficios reales se producen al interactuar con la cadena de bloques.

La siguiente imagen describe este proceso. El contrato inteligente codifica el acuerdo entre la compañía que recauda fondos y sus inversores (1). El contrato inteligente se encuentra en la cadena de bloques pública de Ethereum y se ejecuta en la Máquina virtual de Ethereum (EVM). Una vez que se alcanza un evento desencadenante, como una fecha de vencimiento o un precio de ejercicio que ha sido precodificado, el contrato inteligente se ejecuta automáticamente según la lógica comercial (2). Como beneficio adicional, los reguladores pueden analizar la actividad del mercado de manera continua, sin comprometer la identidad de jugadores específicos en una blockchain pública sin permiso, como Ethereum (3).

Fig. 3 Smart Contract en Ethereum - Fuente: Hyperledger Fundation

4.6.2 Smart contracts en Hyperledger

Los smart contracts en Hyperledger se denominan "*chaincode*" y, para quienes tienen conocimientos más avanzados, se escriben en un lenguaje de programación que puede ser Go, node.js o Java. El estado creado dentro de un *chaincode* es exclusivamente accesible desde ese contrato, es decir ningún otro *chaincode* tendrá acceso directamente a ese estado del mundo.

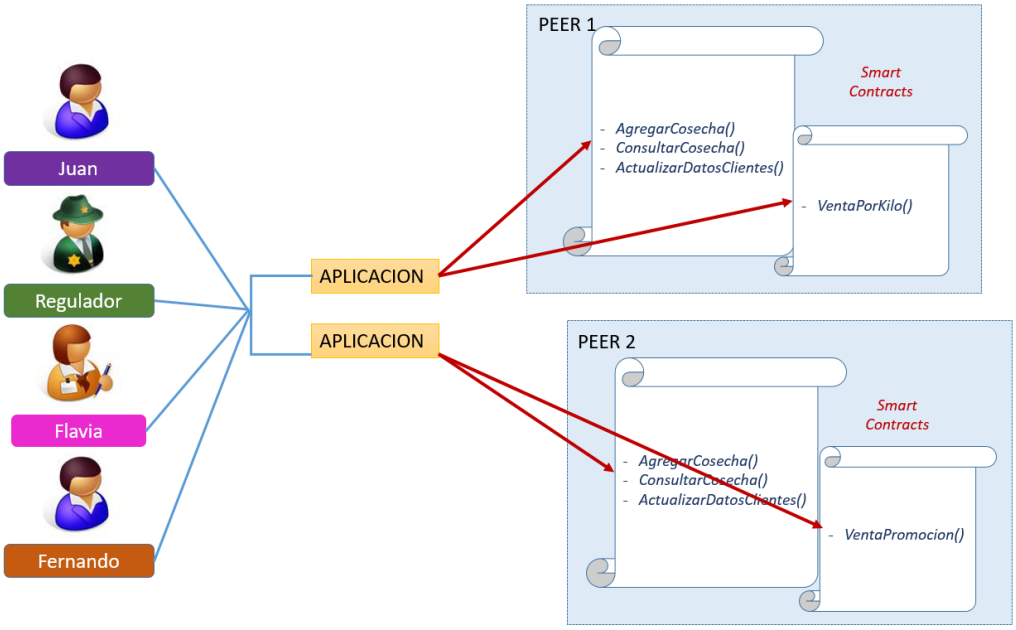

Fig. 4 Smart contracts en Hyperledger

En algunas ocasiones, dados los permisos de red correspondientes, un chaincode podrá invocar a otro chaincode.

4.7 Elección de la plataforma

Es innumerable la cantidad de plataformas y herramientas disponibles en Internet. Siempre existirá una que mejor se adapte a nuestro caso de uso.

Un consejo es adoptar una herramienta que cuente con soporte técnico, que se encuentre lo suficientemente madura y, fundamentalmente, en la que encontremos ejemplos similares a la blockchain que deseamos crear.

5. CONCLUSIONES DEL MÓDULO

En este módulo introducimos los pasos necesarios para poder implementar una blockchain desde la definición de la red de negocios hasta los smart contracts.

También abordamos el estudio de distintas herramientas para crear una blockchain: Ehtereum que nos permite crear blockchain no permisionadas, Hyperleger que nos permite construir blockchain permisionadas, y un breve resumen sobre Corda y IOTA, menos conocidas y utilizadas.

Se realizó una comparación entre las mismas abordando distintos aspectos como sus mecanismos, la forma de implementar la criptografía y generar hashes.

Finalmente vimos de forma general cómo desarrollar *smart contracts* en Ethereum e Hyperledger.