Componente formativo

Diseño de contratos inteligentes

Breve descripción:

Mediante el presente componente se reconocerán los conceptos más importantes para el diseño y construcción de un contrato inteligente de una red de “blockchain” y los elementos necesarios para su despliegue en producción.

Área ocupacional:

Tecnologías de la información.

Abril de 2023

Grupo de Ejecución de la Formación Virtual

Tabla de contenido

Introducción

El “blockchain” es una tecnología que permite llevar un registro seguro, descentralizado, sincronizado y distribuido de las operaciones digitales, sin necesidad de la intermediación de terceros. Se da la bienvenida al estudio del componente formativo “Diseño de contratos inteligentes”, para comenzar se invita a explorar el recurso que se presenta a continuación:

“Blockhain”: contratos inteligentes

“Blockhain”: contratos inteligentes

Síntesis del video: “Blockhain”: contratos inteligentes

El término “blockchain” en los últimos años se ha ido apropiando y posicionando en diferentes sectores, además de aumentar de forma considerable su consulta en la red. Así como su aplicación en diferentes ámbitos laborales y de la administración pública, que auguran un total éxito a este nuevo sistema.

Con el surgimiento del “blockchain” a mediados de los años 90, el criptólogo Nick Szabo buscó una forma alterna de dinamizar las transacciones entre personas o demás sistemas para el intercambio de información.

En este momento surgen los contratos inteligentes o “smart contract”, los cuales permiten establecer una lógica para adaptar protocolos y generar una comunicación entre 2 partes. Estos contratos inteligentes suelen crearse a partir de lenguajes de programación que posteriormente se despliegan sobre las redes de “blockchain”, los cuales requieren de recursos o tokens para garantizar el derecho al uso los mismos, con un bajo costo y alta confiabilidad.

1. Criptografía

Son métodos y mecanismos para mantener segura la información, haciendo uso de técnicas, algoritmos y códigos que procesan un dato entrante, limitando la lectura o interpretación por un tercero no autorizado.

Ahora bien, ya conoce que el “blockchain” es una red segura y cifrada que almacena la información en bloques y, el hecho de necesitar interactuar con esta información obliga a mantener canales como mecanismos de operación seguros implementados en los contratos inteligentes, que en primer lugar logren interactuar con los protocolos nativos de la red, así como garantizar la seguridad de la información intercambiada.

Definición

El “blockchain” es una estructura de datos que almacena información en bloques a los cuales puede agregarse información denominada metainformación, además cuenta con su propio “hash” y el “hash” del bloque

inmediatamente anterior, la cual permite mantener una estrecha relación con otros bloques de una red sosteniendo una relación y línea temporal; la información se almacena haciendo uso.

Conceptos relacionados con la criptografía

En relación a la criptografía son varios los términos de uso que se encuentran asociados, a continuación, podrá relacionar aquellos indispensables para entender mejor este concepto, véalo:

Cifrado: de acuerdo con WeLiveSecurity (2021), el cifrado es "el proceso mediante el cual se codifica algo de modo que no resulte fácil de entender para quienes no tienen acceso autorizado", también es necesario conocer sobre los procesos de criptografía mediante los cuales se aplican métodos de cifrado a una información para ser transmitida, estos procesos tienen como objetivos:

a) Garantizar la privacidad y confidencialidad, de tal manera que únicamente el destinatario interesado pueda leerla.

b) Integridad, evitando que esta sea modificada sin autorización.

c) Autenticación, garantizando que solo se pueda interpretar por los interesados.

d) No repudio, para evitar que se niegue que alguno de los interesados ha podido accederla.

Autenticidad: garantiza la legitimidad de la fuente de la transmisión o información; se busca garantizar que el emisor de un mensaje es quien dice ser.

Integridad: garantiza la persistencia y completitud de los datos o información transmitidos, garantizando que un documento no ha sido modificado por ningún agente externo a la comunicación.

No repudio: un emisor de un mensaje no puede negar haberlo enviado. En otras palabras, evita el rechazo interesado de los mensajes por parte de los comunicantes.

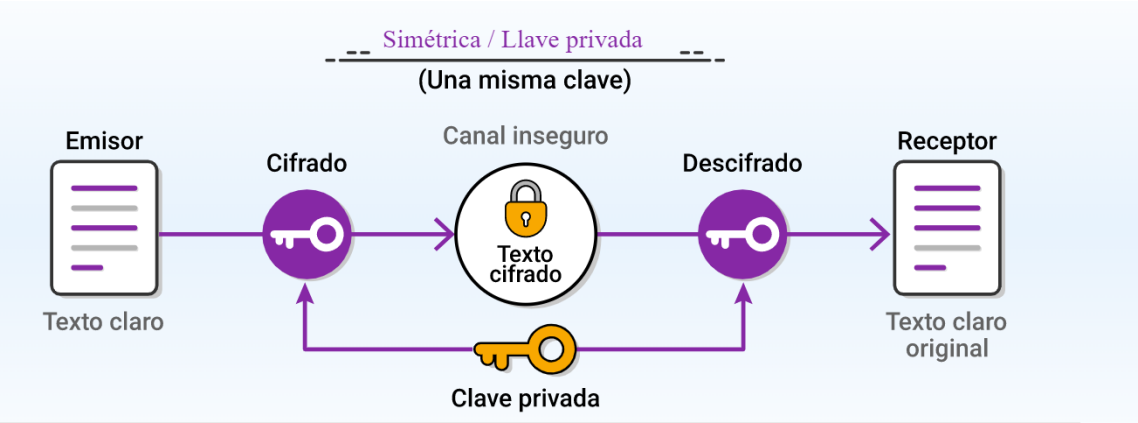
Firma digital: es un mecanismo mediante el cual permite al receptor de un mensaje garantizar que el origen es auténtico, así mismo, se puede comprobar si un mensaje ha sido modificado o se conserva intacto. Aquí están las dos fases para la realización de la firma digital:

1. Proceso de firma: el emisor cifra los datos con la clave privada y los manda al receptor.
2. Verificar la firma: el receptor descifra los datos usando la clave pública del emisor y comprueba que la información coincide con los datos originales (si coincide es que no se ha modificado).

Llave privada (simétrica)

La criptografía basada en llave privada o también conocida como criptografía simétrica consta de una sola en el proceso, esta clave se utiliza tanto para el cifrado como para el proceso de descifrado.

Criptografía de llave privada / simétrica

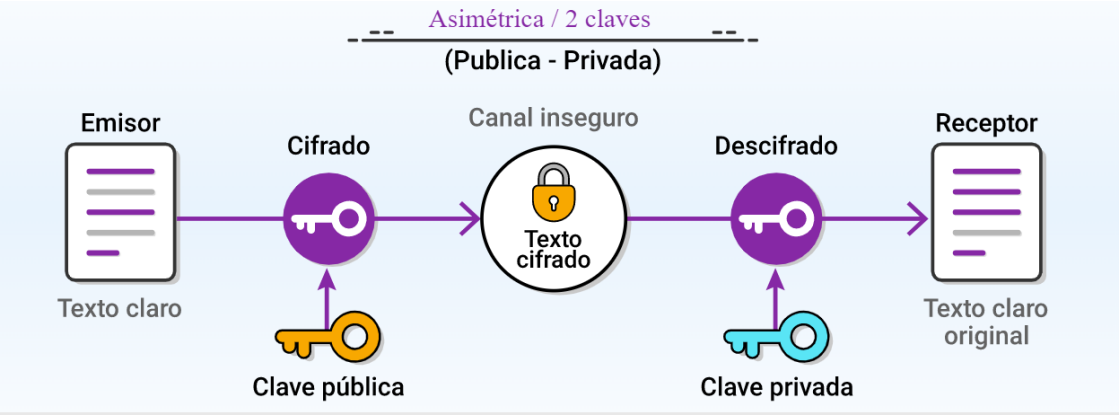


Este tipo de criptografía cuenta con ciertas desventajas, entre las cuales se puede identificar que el hecho de usar una misma clave genera problemas con las transferencias de información segura en el momento de compartir sus claves.

Llave pública (asimétrica)

La criptografía de llave pública o también conocida como asimétrica utiliza 2 claves, las cuales son: llave pública utilizada para el proceso de cifrado y la llave privada que es utilizada para el proceso de descifrado y la cual es compartida con el receptor de la información.

Criptografía de llave pública / asimétrica



Las 2 claves son generadas con algoritmos diferentes, lo que dificulta obtener la llave privada a partir de la llave pública.

Este tipo de criptografía es muy útil cuando se requiere firmar documentos o compartir información sensible.

Algoritmos de cifrado

Como pudo observar anteriormente, existen 2 tipos de criptografía como son la llave privada (simétrica) y llave pública (asimétrica), para lo cual se puede observar algunos de los siguientes algoritmos para cada uno de estos tipos de criptografía:

1. **Llave privada (simétrica):** los algoritmos utilizados en este tipo de criptografía al contar con una misma clave para el cifrado como para el descifrado cuentan con un factor importante que es su rapidez, la cual puede ser aprovechada por la nueva tecnología y realizar este proceso de manera inmediata.

Existen 2 algoritmos muy utilizados en este tipo de criptografía, que son:

AES (“Advanced Encryption Standard”) y ChaCha20.

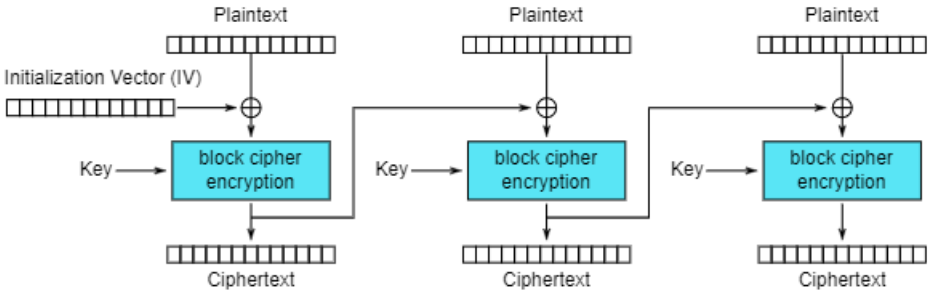
Ahora se procederá a realizar una explicación de estos algoritmos:

AES (“Advanced Encryption Standard”): este algoritmo reemplazó al algoritmo DES y se utiliza comúnmente en canales y protocolos seguros como TLS, FTPES, VPS, entre otros. Además, puede ser aprovechado tanto por “software” como por hardware; este es un algoritmo de cifrado por bloques, cada bloque cuenta con un tamaño fijo de 128 “bits”, su longitud de la clave puede ser variable entre 128, 192 y 256 “bits”.

AES cuenta con diferentes modos de cifrado o manera de gestionar sus bloques, cada uno de ellos operando de manera diferente, entre los cuales se encuentra la siguiente distribución:

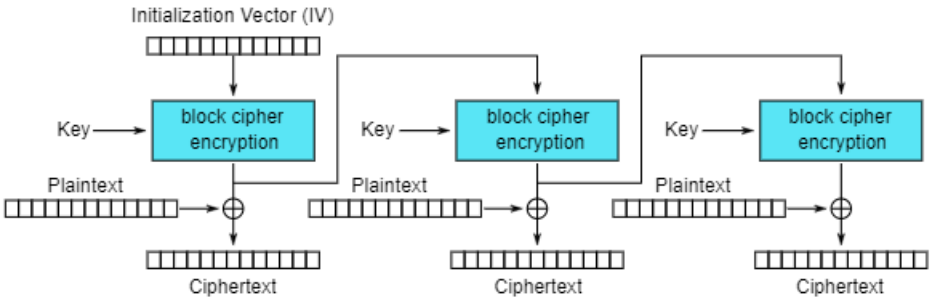
1. **CBC (“Cipher-block chaining”):** trabaja con una función hash para comprobar la autenticidad de la información, con este modo de cifrado a cada bloque de texto sin formato se le aplica una operación XOR con el bloque de cifrado anterior, cada bloque cifrado es dependiente de lo procesado hasta ese punto. Para realizar esta opción XOR con el primer bloque de texto se hace uso de un vector de inicialización IV. Este modo de cifrado se realiza de forma secuencial, no permite ser tratado de forma paralela para aumentar el rendimiento en el cifrado/descifrado de los datos.

“Cipher-block chaining”



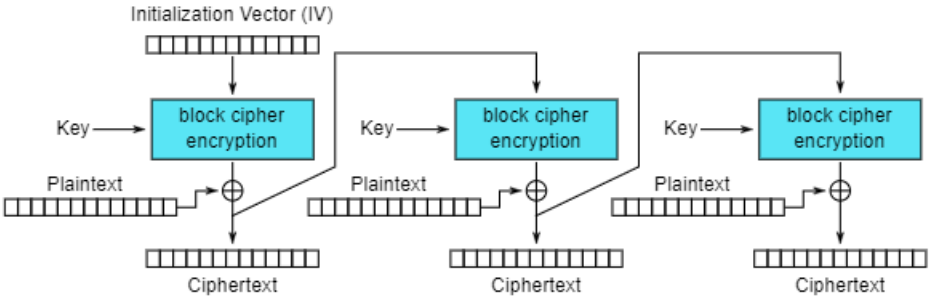
2. **OFB (“Output feedback”):** utiliza una clave secreta para crear un bloque seudo-aleatorio al que se le aplica la operación XOR con el texto sin formato para cifrar el texto, también requiere de un vector de inicialización único para cada mensaje, no se puede paralelizar.

“Output feedback”



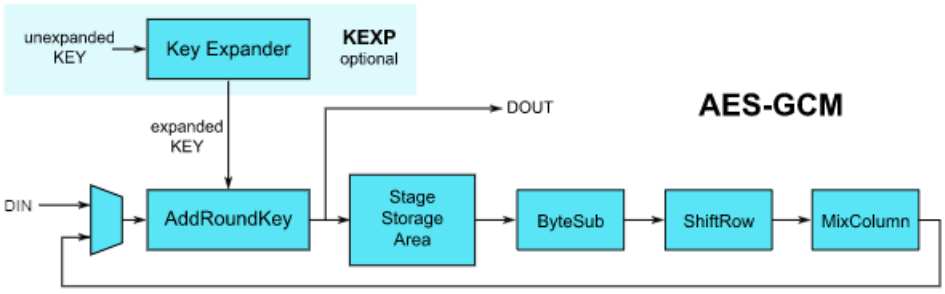
3. **CFB ("Cipher feedback"):** funciona similar al OFB, a diferencia que para producir el "keystream" cifra el último bloque de cifrado, en lugar del último bloque del "keystream" como hace OFB, no se puede paralelizar el cifrado, aunque el descifrado sí.

"Cipher feedback"



4. **GCM ("Galois/Counter Mode"):** considerado uno de los mejores por su seguridad y velocidad, permite procesar en paralelo y es compatible con procesadores AES-NI para acelerar el rendimiento en cifrado/descifrado de información. Este modo de cifrado es AEAD, además de cifrar información es capaz de autenticarla y comprobar la integridad de la misma para garantizar que no se ha modificado. Puede aceptar también vectores de inicialización aleatorios.

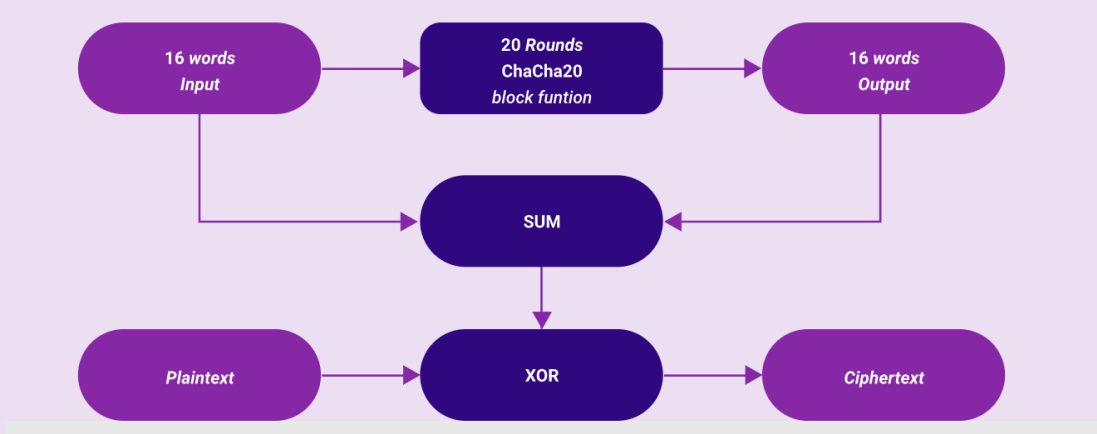
Galois/Counter Mode



AES es uno de los algoritmos más utilizados actualmente; pero el modo más recomendado es AES-GCM ya que incorpora AEAD.

ChaCha20: este algoritmo soporta claves de 128 y 256 bits y de alta velocidad, a diferencia de AES que es un cifrado por bloques, este es un cifrado de flujo, presenta características similares a su predecesor Salsa20; pero con una función primitiva de 12 o 20 rondas distintas:

Algoritmo de cifrado ChaCha20



5. **Llave pública (asimétrica):** los algoritmos utilizados en esta criptografía asimétrica están basados en funciones matemáticas, lo que dificulta su descifrado, además de contar con 2 claves como se vio anteriormente, estas 2 claves tienen funciones primordiales como cifrar información, asegurar la integridad del dato y garantizar la autenticidad del emisor.

Las funciones matemáticas utilizadas están dadas por una estructura definida de acuerdo con la función que se realice, por ejemplo:

Mensaje + clave pública = mensaje cifrado.

Mensaje encriptado + clave privada = mensaje descifrado.

Mensaje + clave privada = mensaje firmado.

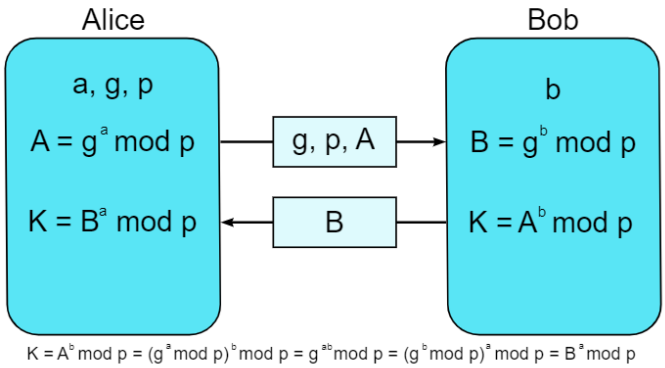
Mensaje firmado + clave pública = autenticación.

También se encuentran algunos algoritmos propios para el cifrado y descifrado de información:

Algoritmo Diffie-Hellman: este más que un algoritmo es un protocolo de establecimiento de claves, utilizado para generar una clave privada en ambos extremos de un canal de comunicación inseguro. Es utilizado para obtener una clave privada con la que posteriormente se cifrará la información junto con un algoritmo de cifrado

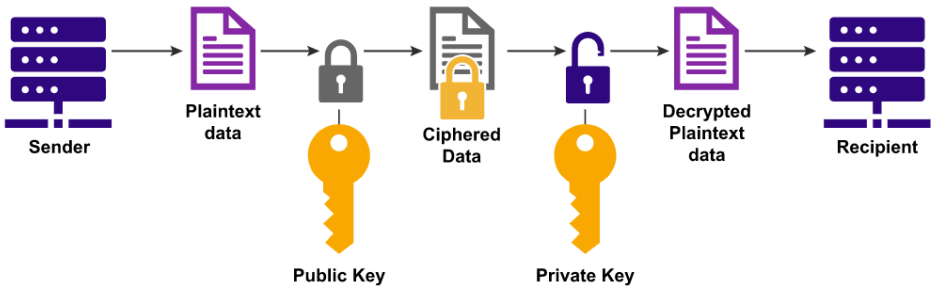
simétrico. El punto fuerte de este algoritmo es que su seguridad radica en la dificultad de calcular el logaritmo discreto de números grandes (Diffie-Hellmann también permite el uso de curvas elípticas).

Algoritmo Diffie-Hellman



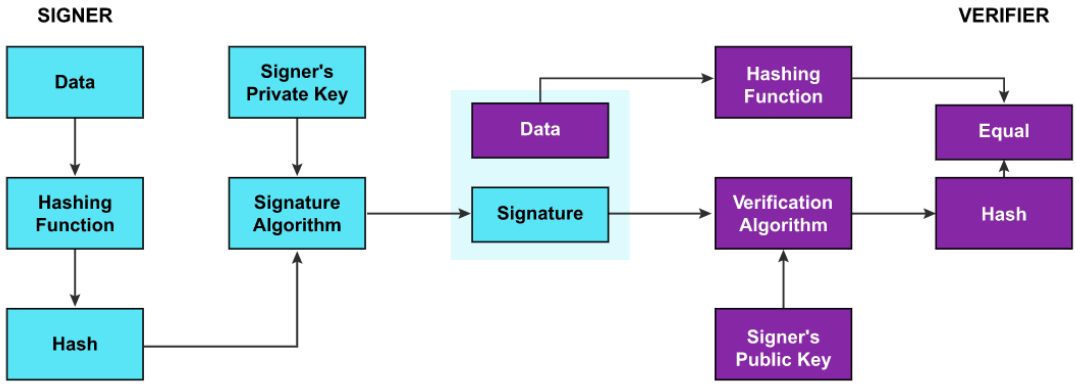
Algoritmo RSA: se basa en la pareja de claves, la pública y la privada. Su seguridad radica en el problema de la factorización de números enteros muy grandes, y en el problema RSA, porque descifra por completo un texto cifrado con RSA no es posible actualmente, aunque sí un descifrado parcial. Algunas características muy importantes de RSA es la longitud de clave, actualmente como mínimo se debe utilizar una longitud de 2048 bits, aunque es recomendable que sea de 4096 bits o superior para tener una mayor seguridad.

Algoritmo RSA



Algoritmo DSA: este algoritmo requiere de mucho más tiempo de cómputo que RSA a la igualdad de hardware. Es utilizado como algoritmo de firma digital, lo que lo convierte en un estándar, DSA no cifra datos, solamente se utiliza como firma digital. Este algoritmo se utiliza ampliamente en las conexiones SSH para comprobar la firma digital de los clientes, además, existe una variante de DSA basada en curvas elípticas (ECDSA), y está disponible en todas las librerías criptográficas actuales como OpenSSL, GnuTLS o LibreSSL. Otra característica de DSA es la longitud de clave, la mínima longitud de clave es de 512 bits, aunque lo más habitual es usar 1024 bits.

Algoritmo DSA



Estos algoritmos de cifrado pueden ser utilizados en diferentes modelos y lógicas de negocio de los contratos inteligentes que necesite diseñar, basta con identificar claramente los sistemas con los cuales se requiere interactuar y sobre cuáles componentes para tomar la decisión de uso.

Hash

Es el resultado de generar un valor de longitud fija, a partir del cifrado de un dato sin formato con la ayuda de una función o algoritmo criptográfico; este proceso por lo general es irreversible.

Puede observar un ejemplo en la tabla No. 1, en la que se ha generado la función “hash” de los algoritmos MD5 y SHA1 sobre un trozo de información, obteniendo el resultado “hash”.

Ejemplo de generación de “hash”

Información	Función	Salida
	MD5 Hash	c2010e3db97c16402d7150cf2eae8591
Saludos desde el complementario de “blockchain”: contratos inteligentes.		
Saludos desde el complementario de “blockchain”: contratos inteligentes.	SHA1 Hash	16c1ffb640cbddc213dfefe4ce8ef34627bd3c3d

Los valores resultantes del hash son muy comunes para almacenar información sensible como, por ejemplo, contraseñas, métodos de pago, entre otros.

Como se ha visto, la tecnología “blockchain” incorpora componentes seguros que garantizan la gestión y salvaguarda de la información, haciendo uso de métodos y algoritmos criptográficos, los cuales se convierten en un componente fundamental. Se invita a continuar conociendo más sobre este concepto, ya que deberá ser incluido en el proceso de construcción de los contratos inteligentes.

Contratos Inteligentes

Han surgido con la aparición del “blockchain” a mediados de los años 90, cuando el criptólogo Nick Szabo buscó cómo proporcionar mecanismos que permitieran el comercio electrónico adaptándose a los nuevos retos bajo la red de “blockchain”, a partir de ese momento han surgido iniciativas sobre cómo garantizar el desarrollo de estas actividades con la seguridad y confiabilidad necesaria y que brinde la confianza al usuario, de esta forma surgen los contratos inteligentes, los cuales permitieron programar técnicamente la lógica y controles necesarios para garantizar la seguridad y confidencialidad de la información.

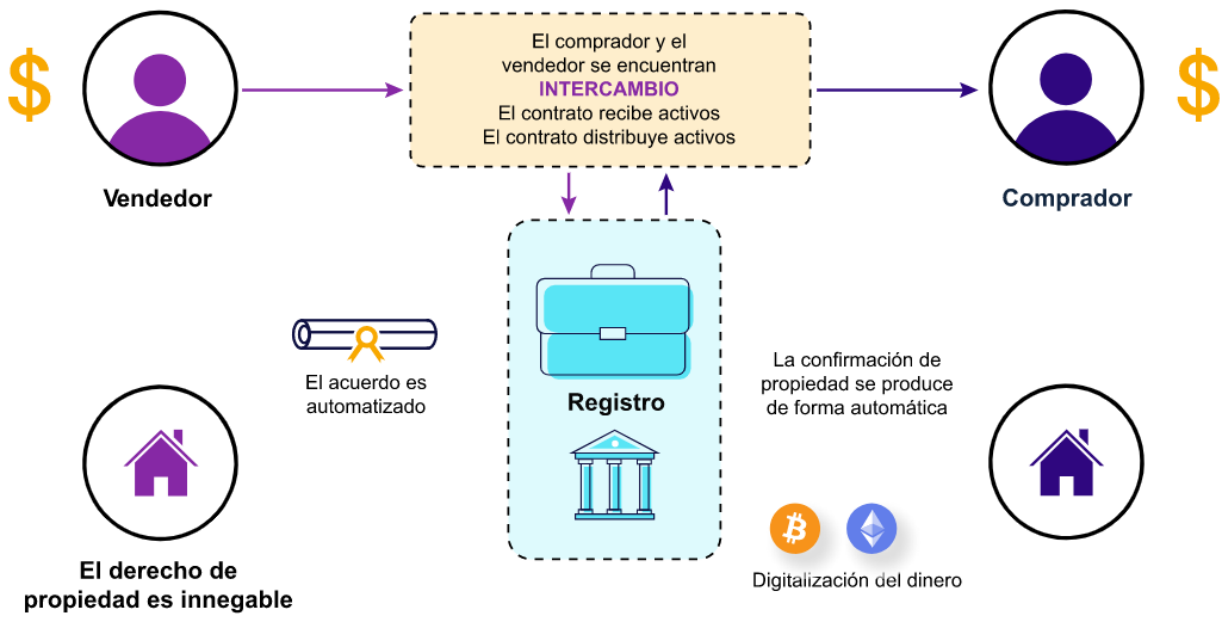
A continuación, va a conocer un poco sobre qué son los contratos inteligentes y cómo se puede determinar las características para su construcción.

Definición de contrato inteligente

“Smart contract” como su nombre lo indica, son contratos iguales a los contratos ordinarios en donde se requería participar presencialmente, firmar y/o sellar, pero ahora se cuenta con un mecanismo y agilidad que permite establecer acciones para una transacción sin estar presente, bajo algunos controles y a un bajo costo de transacción. Para la creación de un contrato inteligente se hace uso de lenguajes de programación sobre los cuales se programan las reglas, condiciones y flujo de información para finalmente registrar dicha información sobre una red de “blockchain”.

El término “smart contract” está asociado a contratos automatizados, lo cual es correcto; pero, vale la pena aclarar que, si bien los orígenes del término y las intenciones de quien lo acuñó por primera vez quizás se centraron en permitir contratos automatizados, la implementación final sobre una red de “blockchain” los ubica más próximos a programas de computación de uso más o menos genérico, con la posibilidad de acceder a dinero (“token's”) y con algunas restricciones y costos de ejecución.

Cómo funcionan los contratos inteligentes



Es así como se ha favorecido la inclusión de estos contratos inteligentes dentro de aplicaciones y soluciones de comercio electrónico basadas en "blockchain" para facilitar diferentes actividades como se puede ver a continuación:

1. **Gobierno y administraciones públicas:** seguridad en los registros y trazabilidad en las operaciones.
2. **Mercados inmobiliarios:** transparencia e inmutabilidad, además de agilidad en los negocios.
3. **Transporte**: eficacia, seguridad en inmutabilidad de los datos.
4. **Automóvil:** recordación de deudas, hasta autos que se manejan solos.
5. **Salud:** minimizar la complejidad de los métodos tradicionales.
6. **Auditoría:** campo especializado a nivel profesional, que va en aumento.
7. **Compañías de seguros:** reducción de costos, garantías de seguridad.
8. **Banca**: capacidad de autoejecutar acciones, reducción de intermediarios.
9. **Notariado y Registro:** permite proteger al ciudadano, agiliza la certificación, y disminuye la demora en el trámite.

Si se realiza una comparación con los contratos tradicionales se puede ver una marcada diferencia que invita a estudiar la posibilidad de adopción de esta tecnología, en la siguiente tabla podrá encontrar un paralelo que permite identificar la gran diferencia de los tipos de contratos.

Comparación de contratos tradicionales vs. contratos inteligentes

Contrato	Tiempo	Asignaciones	Fidecomiso	Costo	Presencia	Abogado
Tradicionales	1 a 3 días	Manuales	Necesario	Costoso	Física, se requieren firmas estampadas	Se necesita
Inteligentes	Minutos	Automáticas	Puede no ser necesario	Fracción del costo	Virtual, firma digital	Puede ser no necesario

Como se indicó anteriormente para la construcción de un contrato inteligente se debe hacer uso de lenguajes de programación, entre los más reconocidos y utilizados actualmente está Solidity, de este lenguaje de programación se puede encontrar en su documentación oficial (2022):

Solidity es un lenguaje de alto nivel orientado a contratos. Su sintaxis es similar a la de JavaScript y está enfocado específicamente a la máquina virtual de Etehreum (EVM). Además, está tipado de manera estática y acepta entre otras cosas, herencias, librerías y tipos complejos definidos por el usuario.

Puede ver a continuación un ejemplo de un contrato inteligente programado en Solidity, el cual imprime la cadena de texto “¡Hola mundo!”

Hola mundo en lenguaje de programación Solidity

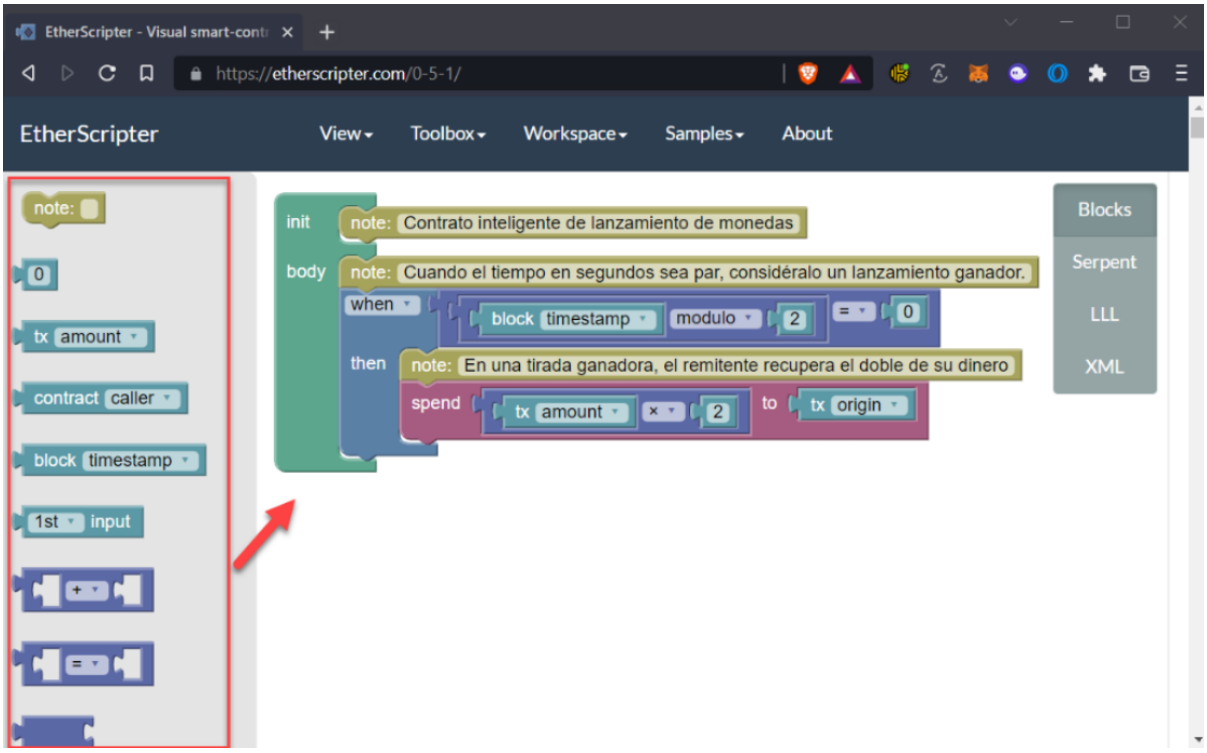
```
contract HolaMundo {
    event Escribir(string out);
    function() {
        Escribir("¡Hola mundo!");
    }
}
```

De acuerdo con lo anterior, se puede determinar que este lenguaje presenta una sintaxis amigable, con una curva de aprendizaje moderadamente compleja y que soporta el paradigma de la programación orientada a objetos, lo que permitirá reducir y reutilizar el código existente.

Puede conocer más sobre este lenguaje de programación en su sitio web oficial, al cual puede acceder mediante este enlace. Programación en su sitio web oficial.

También existen alternativas para usuarios no técnicos o aquellos que no manejan lenguajes de programación, que les ayudan a construir una lógica del contrato inteligente apoyado en una interfaz de bloques, lo que permite comprender lo que se está programando, y que posteriormente se puede exportar para complementar y publicar, una de estas es https://remix.ethereum.org/

Editor visual de contratos inteligentes EtherScripter



Esta plataforma basada en bloques permite arrastrar bloques con enganches similares a un rompecabezas y a medida que los enganchamos cada parte sirve como condicional lógico, repeticiones, y todo lo necesario para poder generar un contrato inteligente básico.

Reglas en contratos inteligentes

Sin duda, como en cualquier proceso de construcción de un programa o un proyecto de diseño de aplicaciones, este debe de cumplir con algunas fases y reglas que garantizan que al final del proceso, el contrato inteligente realice lo que se necesita sin errores, así las cosas, se va a enumerar algunas recomendaciones y reglas que se deben de tener presente en el momento de construir un contrato inteligente, como se muestra a continuación:

1. **Planificación:** es necesario conocer las necesidades que debe solucionar un contrato inteligente, determinando los recursos, lógica del programa, entradas de información, salidas de información, así como aplicaciones con las que debe interactuar e intercambiar información.

2. **Conocer los recursos disponibles y las DApps:** el incursionar en la tecnología “blockchain” lleva a estudiar y explorar las Dapps, las cuales están conformadas por contratos inteligentes, estas corren sobre una red de “blockchain”, así como diferentes recursos ya creados de las que puede hacer uso libremente.

3. **Construir el contrato con el lenguaje adecuado:** estudiar y conocer un lenguaje de programación especializado en la creación de contratos inteligentes como, por ejemplo, Solidity y su entorno de desarrollo Remix permitirá crear, testear y prepararse para desplegar sobre una red de producción de contratos inteligentes.

4. **Libre de interrupciones:** la codificación del contrato inteligente debe involucrar buenas prácticas de programación y debe garantizar estar libre de interrupciones que puedan poner en riesgo el normal funcionamiento del contrato inteligente y generar interrupción del servicio y/o pérdida de información.

5. **Seguridad:** su codificación debe incorporar controles de seguridad que garantice que solo los sistemas o interesados puedan hacer uso del contrato inteligente.

6. **Sin intermediarios:** un principio de los contratos inteligentes es que estos no deberían depender de un intermediario y deben tener la capacidad de operar sin la participación, ni aprobación de un tercero.

7. **Rapidez:** el contrato debe ser creado para ejecutarse de manera rápida, dado que este puede ejecutarse en múltiples instancias a la vez.

8. **Costo de ejecución mínimo:** el contrato inteligente deberá ejecutarse y funcionar sobre redes de “blockchain”, las cuales requieren de un recurso económico (“tokens”) para pagar el uso de la red, así que se recomienda garantizar un consumo mínimo y sin gastar grandes cantidades de recursos.

9. **Libre de errores:** se recomienda realizar una depuración o auditoría que verifique que el código del contrato inteligente no genere errores que pueda causar inconvenientes con la información transferida.

Estas son algunas reglas y recomendaciones mínimas que deben de tenerse presente en el momento de crear un contrato inteligente.

Tipos de contratos inteligentes

Teniendo en cuenta que el contrato inteligente es un código que permite la ejecución automatizada de las prestaciones pactadas dentro de las herramientas que ofrece la tecnología “blockchain” y el auge que han venido adquiriendo en los últimos tiempos, en la actualidad existe una gran variedad de tipos de contratos inteligentes, entre los más importantes se destacan los siguientes:

1. **Contratos inteligentes legales:** este tipo de contrato inteligente se encarga de verificar el funcionamiento de cualquier transacción, como contratación de productos, servicios, depósitos en garantía, compras y ventas, préstamos, entre otros.

2. **Contratos inteligentes contables:** las organizaciones autónomas descentralizadas mejor conocidas como DAO se encargan de realizar seguimientos a las interacciones financieras, y gracias a la tecnología “blockchain” que las componen son ideales para evitar falsificaciones. Por ende, estas se definen por acuerdos codificados a través de contratos inteligentes de naturaleza contable.

3. **Contratos inteligentes lógicos de aplicación:** este tipo de contrato inteligente opera en conjunto con otros de su clase y con programas automatizados, todo con el propósito de validar y facilitar la interacción entre dispositivos. Estos mayormente son comandados por un programa de gestión.

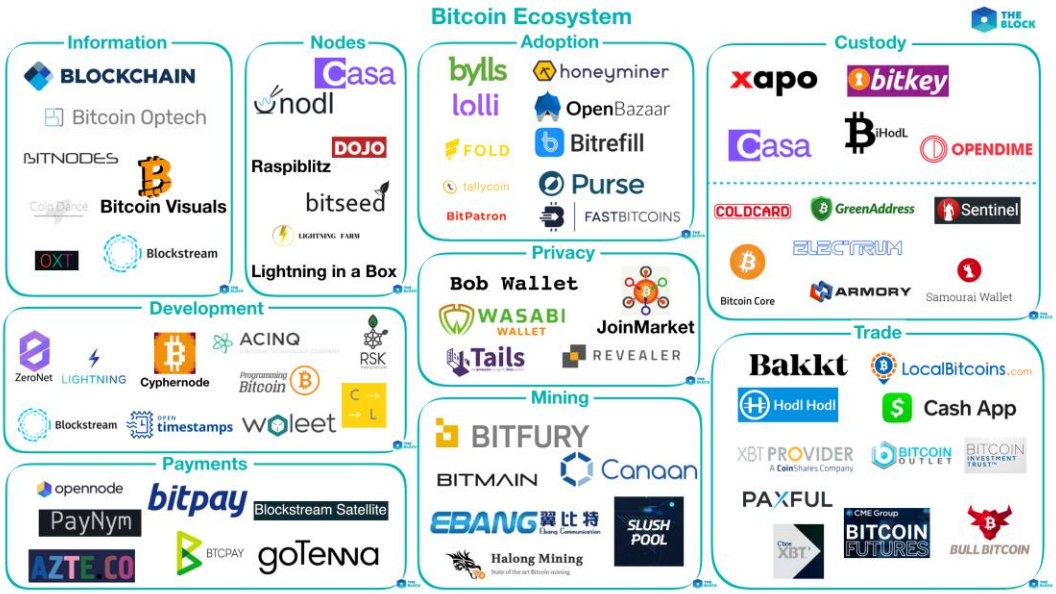
Interacción con otras aplicaciones

El “blockchain” cuenta con un gran número de proyectos que permiten interactuar con sus redes de cadenas de bloques, cada proyecto presenta una propuesta tecnológica que contempla la inclusión de servicios Dapps, Wallet, infraestructura, finanzas descentralizadas, Gaming, NFT entre otras soluciones, a continuación, va a conocer las propuestas de algunos proyectos.

Así el “blockchain” interactúa con otras aplicaciones:

1. **Ecosistema bitcoin:** Este proyecto se ha convertido en una innovadora red de pagos y una nueva clase de dinero, tal como se muestra:

Ecosistema bitcoin

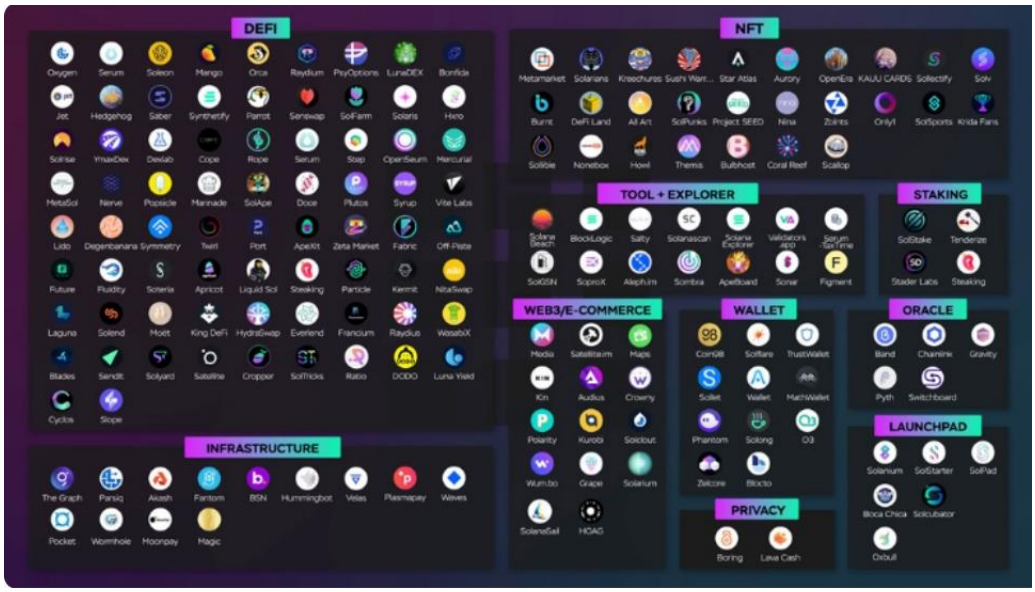


2. **Ecosistema Ethereum:** este proyecto es uno de los proyectos iniciales, y que estableció el punto de partida para los nuevos ecosistemas, cuenta con grandes soluciones para “wallets”, finanzas descentralizadas, mercados descentralizados, juegos, NFT, entre otros.

Proyectos que depende de Ethereum

3. **Ecosistema Solana**: proyecto “blockchain” que busca aprovechar al máximo los NFT y contratos inteligentes.

Ecosistema de soluciones integradas

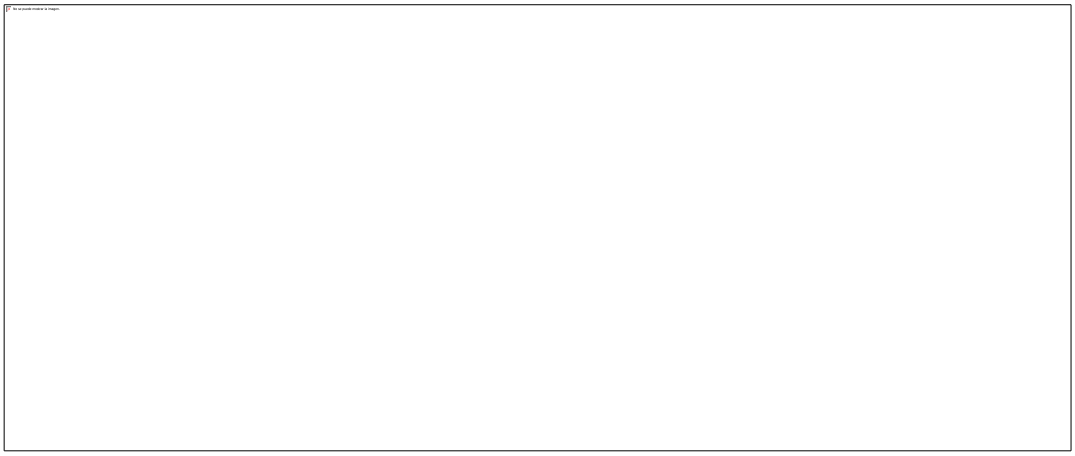


4. **Ecosistema Cardano:** este proyecto promete ser una plataforma para contratos inteligentes y su integración con grandes soluciones.

Ecosistema Cardano

5. **Ecosistema Polkadot:** este proyecto apalancado en proyectos denominados “Parachains” buscan brindar soluciones de diferentes tipos.

Ecosistema Polkadot



Estos son los proyectos de cada uno de los ecosistemas revisados anteriormente, los cuales puede consultar en los siguientes enlaces a su disposición:

Bitcoin

Ethereum

Solana

Cardano

Polkadot

Existen muchos más proyectos con infinidad de propuestas para aplicar en el mundo del “blockchain”, cerca de 10.000 proyectos activos y con infinidad de propuestas y aplicaciones con las cuales se puede implementar con contratos inteligentes, se invita a estudiarlos y tomar su propia decisión sobre con cuál trabajar en sus necesidades.

Sistema distribuido y red extendida

Los contratos inteligentes operan sobre las redes de “blockchain”, por ende, heredan uno de sus principios que es la descentralización y aquí se puede tener presente que cualquier persona que tenga un requerimiento, y que cuente con los conocimientos básicos de programación, puede crear un contrato inteligente para interactuar con las DApps y otros contratos inteligentes existentes en las redes, permitiendo descentralizar la gestión y mantenimiento de los mismos.

A continuación, va a recordar algunos principios básicos del “blockchain” y cómo los hereda los contratos inteligentes.

Conceptos, clasificación

Recuerde que los sistemas distribuidos son sistemas cuyos componentes hardware y software están en computadoras conectadas en red, se comunican y coordinan sus acciones mediante el paso de mensajes para el logro de un objetivo. Se establece la comunicación mediante un protocolo preestablecido.

Su clasificación es:

1. **DAO (Organizaciones Autónomas Descentralizadas):** son sistemas programados que representan organizaciones y que estas operan de forma autónoma. Es0tas DAO no cuentan con un ente que las dirija; pero comparten intereses y objetivos similares, y así es como se ve una organización verdaderamente descentralizada.

2. **Oráculo:** son instrumentos que permiten actualizar estados internos de un contrato inteligente a través de información del exterior (comúnmente APIs), por ejemplo, obtener la cotización de una acción o divisa o si un paquete ha sido enviado por la empresa de transporte.

3. **Red P2P**: red de dispositivos en los cuales cada una de las computadoras, dispositivos móviles IoT, entre otros, se convierten en un nodo capaz de interactuar con otros miembros de la red directamente para ofrecer y consumir servicios de una red de “blockchain”.

4. **Nodos y roles**: los nodos de una red P2P cuentan con diferentes roles dependiendo de las funciones a realizar. Cada nodo posee una colección de funciones para enrutamiento, base de datos de “blockchain”, minería y servicios de billetera.

Sistemas sin intermediarios

Un contrato inteligente busca no depender de una autoridad central, los contratos inteligentes se ejecutan sin intermediarios. En otras palabras, nadie tiene que apretar un botón, tomar una decisión, equivocarse o simplemente, cambiar de opinión en el último momento. Si las condiciones del contrato se cumplen, entonces, este se ejecuta gracias al procesamiento de datos de forma distribuida.

La seguridad y fiabilidad que esto aporta a acuerdos entre partes simplemente no tiene comparación con los métodos tradicionales.

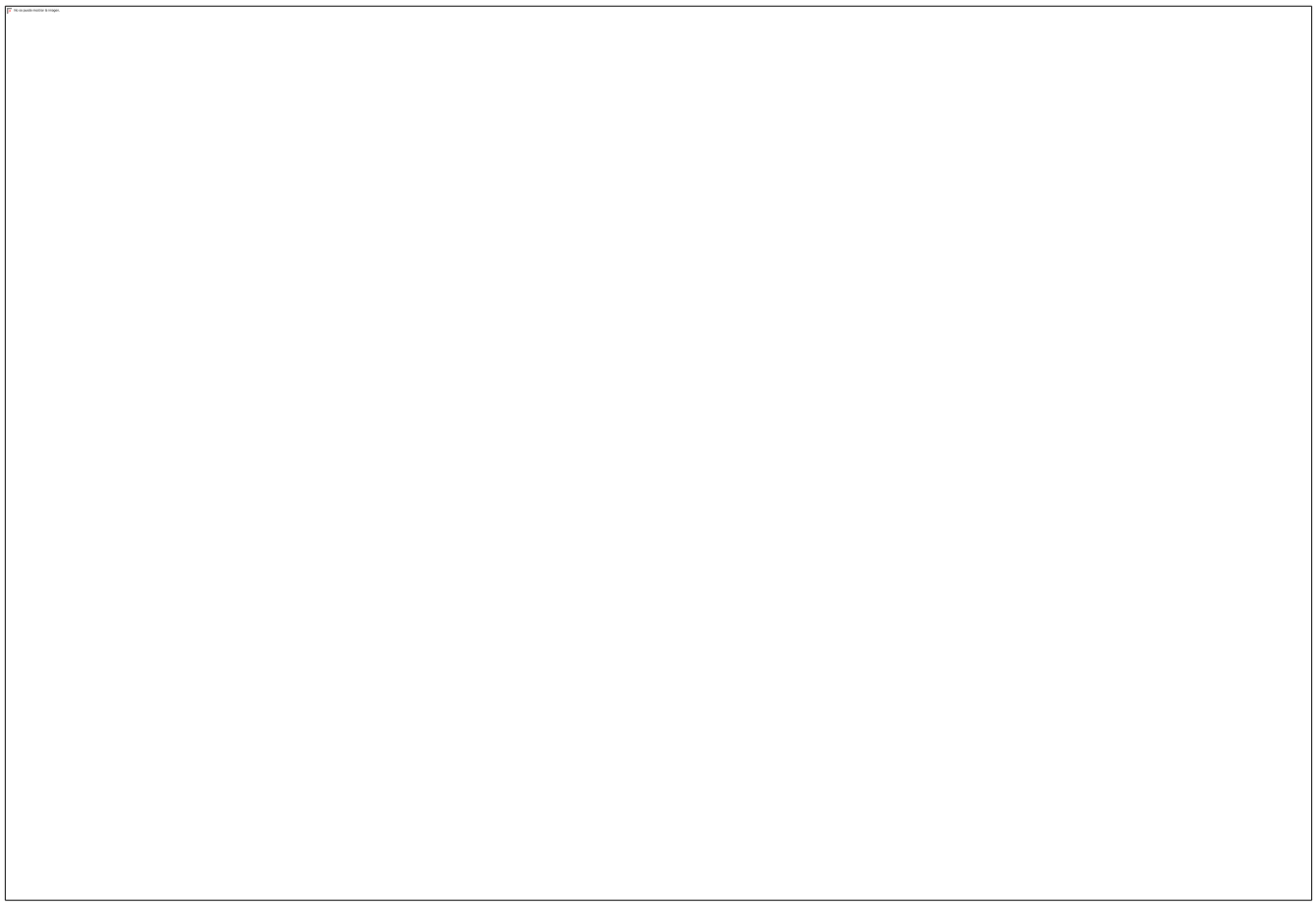
Por ejemplo, un contrato civil puede haber sido firmado por dos o más partes e incluso regulado ante notario; pero, existe la posibilidad de que no se lleve a cabo. Un contrato inteligente no puede no cumplirse, eliminando el error humano.

Con lo anterior se ha reconocido algunos conceptos y fundamentos importantes para iniciar el proceso de construcción de un contrato inteligente, se invita a continuar estudiando los componentes técnicos que le permitirán plasmar, diseñar, construir y desplegar un contrato inteligente en una red de “blockchain”.

Recuerde explorar los demás recursos que se encuentran disponibles en este componente formativo; la síntesis, la actividad didáctica para reforzar los conceptos estudiados, material complementario, entre otros.

Síntesis

A continuación, se muestra un mapa conceptual con los elementos más importantes desarrollados en este componente:



Material complementario

Tema	Referencia APA del Material	Tipo de material	Enlace del Recurso o Archivo del documento material
Criptografía - cifrado	ESET Latinoamérica. (2016). *Cifrado de datos: qué es y cómo puede ayudarte a proteger tu información en Internet [video]. YouTube.*	Video	https://www.youtube.com/watch?v=wcJBmoz6Vlk
Documentación oficial del lenguaje de programación Solidity	Solidity. (2022). *Solitidy documentation. Solidity.*	Manual electrónico	chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://docs.soliditylang.org/_/downloads/en/latest/pdf/
Definición de contrato inteligente	Remix. (s.f.). *File explorers. Remix.*	Página web	https://remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js

Glosario

Contrato: un conjunto de acuerdos o promesas entre agentes.

DAO: son sistemas programados autónomos que representan organizaciones y estas operan de forma autónoma.

Firma digital: protocolo criptográfico basado en criptografía de clave pública, que prueba que un objeto está en contacto activo con la clave privada; correspondiente a la firma, el objeto está activamente “firmado” con esa clave.

Llave: un número aleatorio extraído de un espacio de nombres tan grande que una conjetura afortunada es enormemente improbable.

Protocolo: una secuencia de mensajes entre múltiples agentes.

Referencias bibliográficas

Arroyo, D., Díaz, J. y Hernández, L. (2019). Blockchain. Editorial CSIC Consejo Superior de Investigaciones Científicas. https://elibro-net.bdigital.sena.edu.co/es/ereader/senavirtual/111431

Fuentes, E. (2022). Contratos inteligentes: un análisis teórico desde la autonomía privada en el ordenamiento jurídico colombiano. Editorial Unimagdalena.https://elibro-net.bdigital.sena.edu.co/es/ereader/senavirtual/214513

MinTIC. (2022). Guía de referencia de blockchain para la adopción e implementación de proyectos en el Estado colombiano. MinTIC.https://gobiernodigital.mintic.gov.co/692/articles-161810_pdf.pdf

Solitidy. (2022). Solidity documentation. Solitidy.https://docs.soliditylang.org/_/downloads/en/latest/pdf/

Tudela, L. (2019). Arquitectura blockchain para la securización de dispositivos IOT mediante smart contracts.

Créditos

Nombre	Cargo	Regional y Centro de Formación
Claudia Patricia Aristizábal Gutiérrez	Líder Ecosistema de Recursos Educativos Digitales (RED)	Dirección General.
Liliana Victoria Morales Gualdron	Responsable de la línea de Producción -2023	Regional Distrito Capital- Centro de Gestión de Mercados, Logística y Tecnologías de la Información.
Rafael Neftalí Lizcano Reyes	Responsable Equipo desarrollo curricular	Regional Santander - Centro Industrial del Diseño y la Manufactura
Hernando José Peña Hidalgo	Experto temático	Regional Cauca, Centro de Teleinformática y Producción Industrial
María Inés Machado López	Diseñadora instruccional	Regional Norte de Santander. Centro de la Industria, la Empresa y los Servicios
Carolina Coca Salazar	Asesora metodológica	Regional Distrito Capital- Centro de Diseño y Metrología
Julia Isabel Roberto	Correctora de estilo	Regional Distrito Capital- Centro de Diseño y Metrología
Alix Cecilia Chinchilla Rueda	Metodóloga para la formación virtual	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Eulises Orduz Amezquita	Diseñador web	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Diego Fernando Velasco Güiza	Desarrollador Fullstack	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Ernesto Navarro Jaimes	Animador y Productor Multimedia	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Lina Marcela Pérez Manchego	Validación y vinculación en plataforma LMS	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital

Castaño Pérez Leyson Fabian	Validación y vinculación en plataforma LMS	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Carolina Coca Salazar	Evaluación de contenidos inclusivos y accesibles	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital