**FORMATO PARA EL DESARROLLO DE COMPONENTE FORMATIVO**

| PROGRAMA DE FORMACIÓN | *Blockchain*: contratos inteligentes |
| --- | --- |

| COMPETENCIA | 220501110. Implementar el sistema de seguridad de la información según modelo y estándares técnicos. | RESULTADOS DE APRENDIZAJE | 220501110-01 Planificar el diseño de contratos inteligentes. |
| --- | --- | --- | --- |

| NÚMERO DEL COMPONENTE FORMATIVO | 01 |
| --- | --- |
| NOMBRE DEL 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. |
| PALABRAS CLAVE | Condición, contrato, criptografía, diseño, programación. |

| ÁREA OCUPACIONAL | 6. Ventas y servicios |
| --- | --- |
| IDIOMA | Español |

**a. TABLA DE CONTENIDOS**

**Introducción**

1. **Criptografía** 
   1. Definición
   2. Conceptos relacionados con la criptografía
   3. Llave privada (simétrica)
   4. Llave pública (asimétrica)
   5. Algoritmos de cifrado
   6. *Hash*
2. **Contratos inteligentes**
   1. Definición de contrato inteligente
   2. Reglas en los contratos inteligentes
   3. Tipos de contratos inteligentes
   4. Interacción con otras aplicaciones
3. **Sistema distribuido y red extendida**
   1. Conceptos, clasificación
   2. Sistemas sin intermediarios

**b. DESARROLLO DE CONTENIDOS**

**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:



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.

| Concepto de criptografía dibujada a mano vector gratuito | 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. |
| --- | --- |

**1.1. 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.

**1.2. 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:



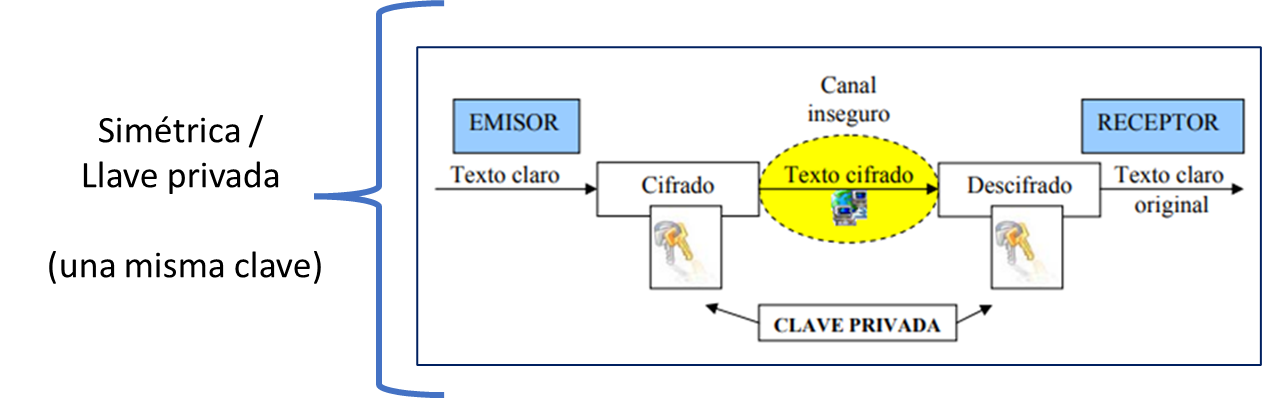


**1.3. Llave privada (simétrica)**

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, como se puede apreciar en la figura No. 1.

**Figura 1**

*Criptografía de llave privada / simétrica*



Nota. Adaptado de [la Oficina de Seguridad del Internauta (osi.es)](https://www.osi.es/es/actualidad/blog/2019/07/10/sabias-que-existen-distintos-tipos-de-cifrado-para-proteger-la-privacidad) https://www.osi.es/sites/default/files/images/concienciacion/c9-img-blog-criptografia.png

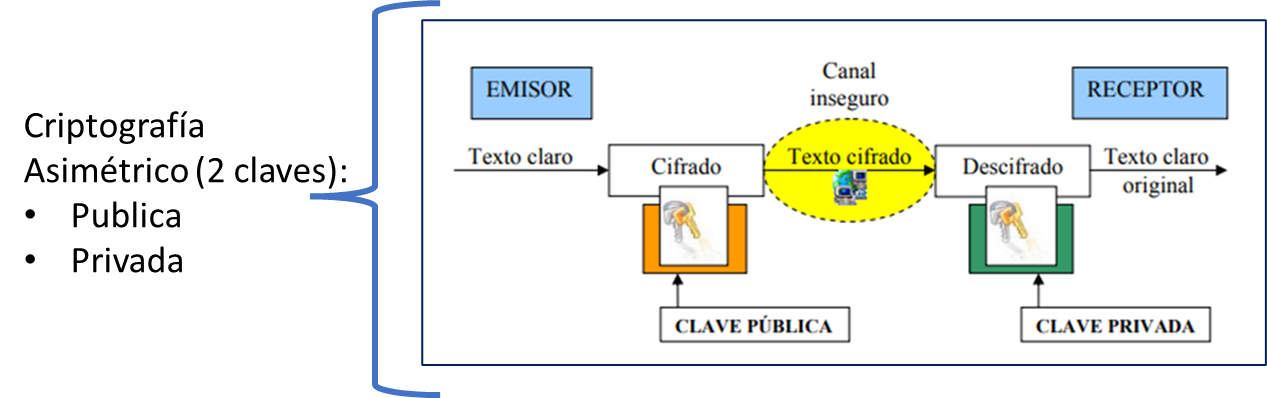
Este tipo de criptografía cuenta con ciertas desventajas, entre las cuales se puede identificar que el hecho de usar una misma clave genera problemas con las transferencias de información segura en el momento de compartir sus claves.

**1.4. 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, tal como se presenta en la figura No 2.

**Figura 2**

*Criptografía de llave pública / asimétrica*



Nota. Adaptado de la [Oficina de Seguridad del Internauta (osi.es)](https://www.osi.es/es/actualidad/blog/2019/07/10/sabias-que-existen-distintos-tipos-de-cifrado-para-proteger-la-privacidad) <https://www.osi.es/sites/default/files/images/concienciacion/c9-img-blog-criptografia.png>

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.

**1.5. 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:

* **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 puede visualizar en la imagen que se presenta a continuación:

**Figura 3**

*Algoritmos de la llave privada (simétrica)*



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, como se aprecia en el recurso educativo a continuación:

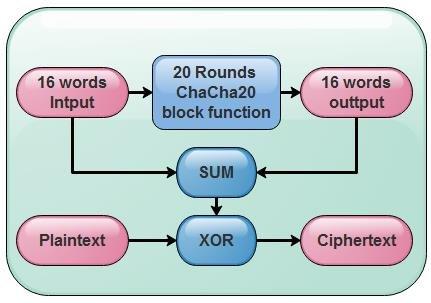


AES es uno de los algoritmos más utilizados actualmente; pero el modo más recomendado es AES-GCM ya que incorpora AEAD

* **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. Ver figura.

**Figura 4**

*Algoritmo de cifrado ChaCha20*



Nota. Tomado de  [A lightweight security scheme for advanced metering infrastructures in smart grid (researchgate.net)](https://www.researchgate.net/publication/338422244_A_Lightweight_Security_Scheme_for_Advanced_Metering_Infrastructures_in_Smart_Grid)

* **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

Aquí también se encuentran algunos algoritmos propios para el cifrado y descifrado de información, los cuales se presentan en el siguiente recurso educativo:



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.

**1.6. *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*, además puede realizar estos ejercicios de prueba y validación en sitios públicos como <http://www.sha1-online.com/> o <https://www.md5hashgenerator.com/> entre otros, y si se requiere integrar sobre algún sistema de información o programa, basta con revisar la documentación del mismo para su adopción.

**Tabla 1**

*Ejemplo de generación de hash*

| INFORMACIÓN | FUNCIÓN | SALIDA |
| --- | --- | --- |
| Saludos desde el complementario de b*lockchain*: contratos inteligentes | MD5 Hash | c2010e3db97c16402d7150cf2eae8591 |
| 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.



1. **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.

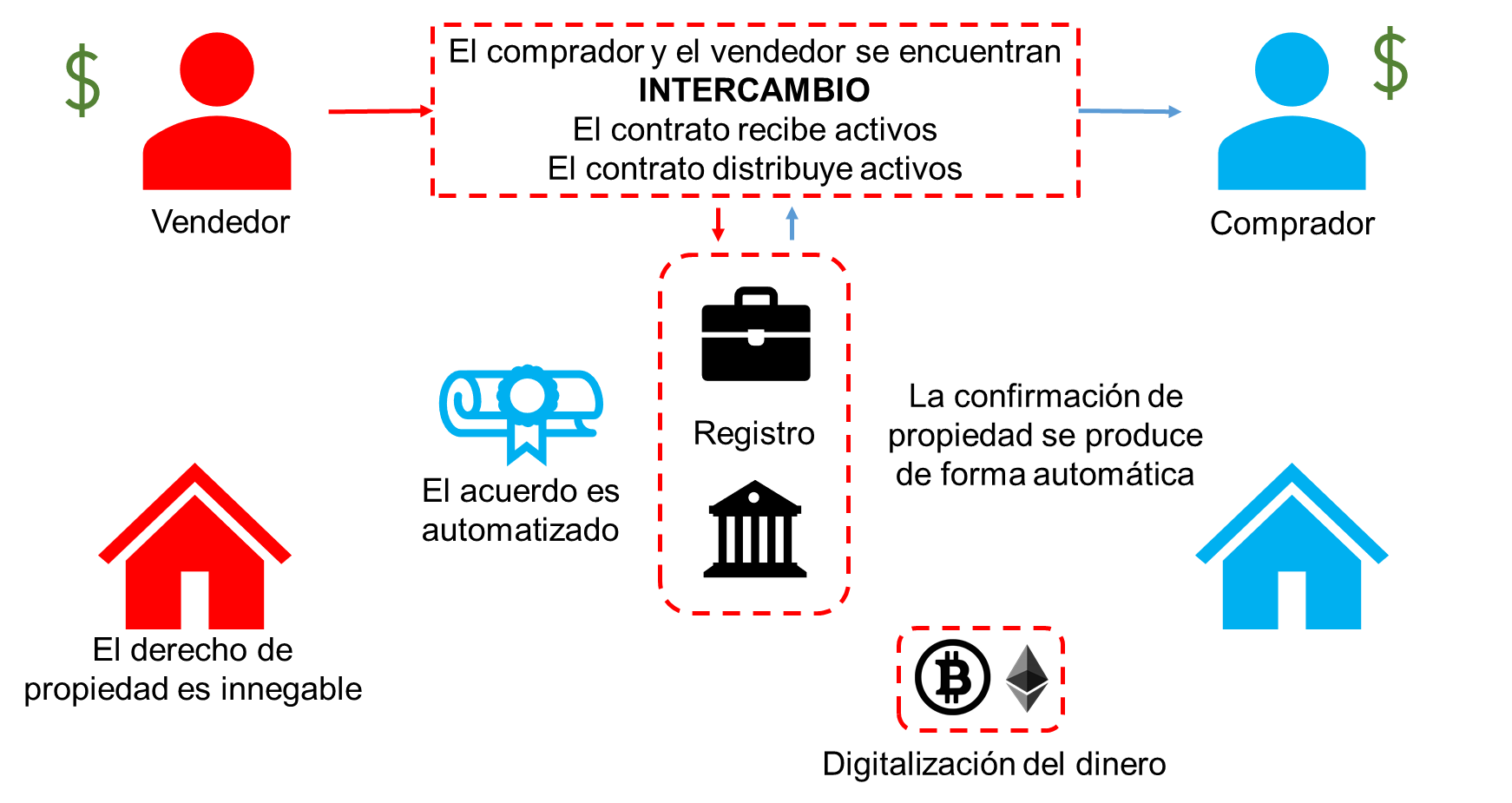
**2.1. Definición de contrato inteligente**

| Contratos inteligentes - Iconos gratis de negocios y finanzas | *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.

**Figura 5**

*Cómo funcionan los contratos inteligentes*



Nota. Adaptado de [Los 5 Beneficios de los SMART CONTRACT (Morant](https://www.alfonsomorant.com/5-beneficios-de-los-smart-contract/), s.f.).

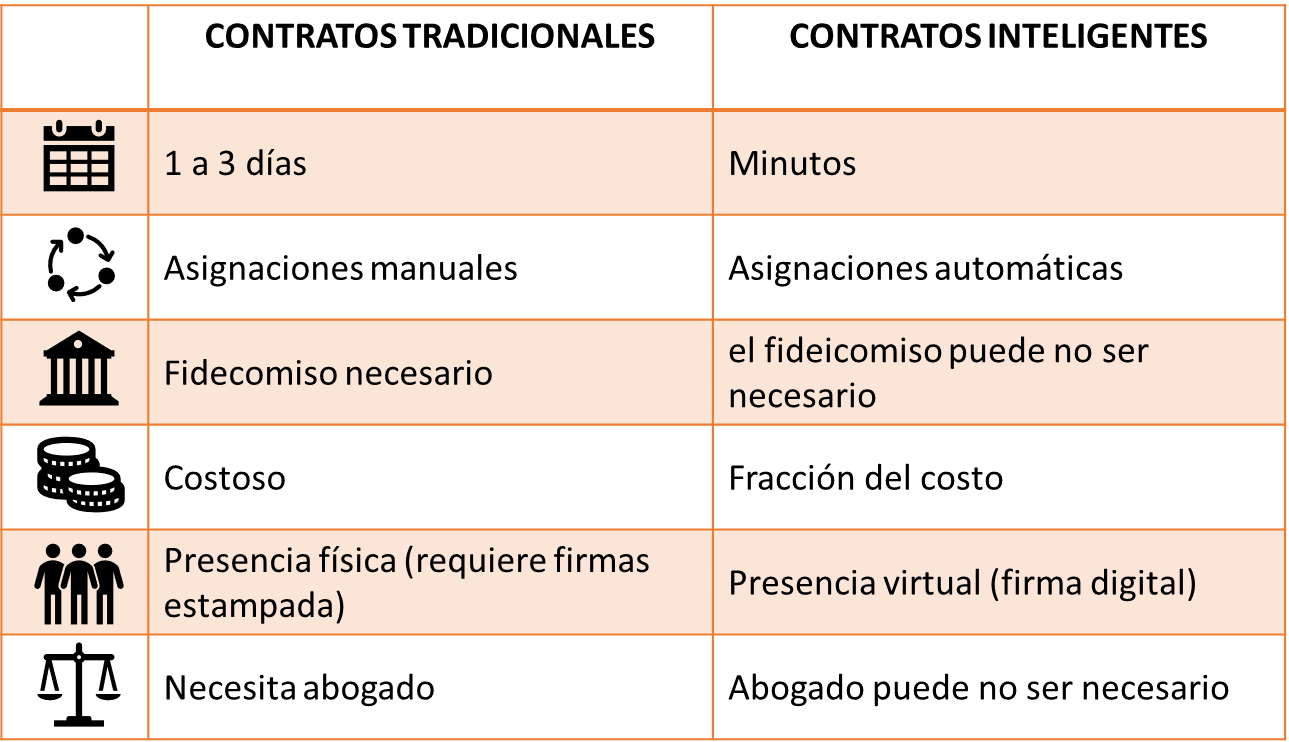
Es así como se ha favorecido la inclusión de estos contratos inteligentes dentro de aplicaciones y soluciones de comercio electrónico basadas en *blockchain* para facilitar diferentes actividades como se puede ver a continuación:



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 figura podrá encontrar un paralelo que permite identificar la gran diferencia de los tipos de contratos.

**Figura 6**

*Comparación de contratos tradicionales vs. contratos inteligentes*



Nota. Adaptado de [What are Smart Contracts. How Ethereum Smart Contract Work (Edureka](https://www.edureka.co/blog/smart-contracts/), s.f.).

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):



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!");  }  } |

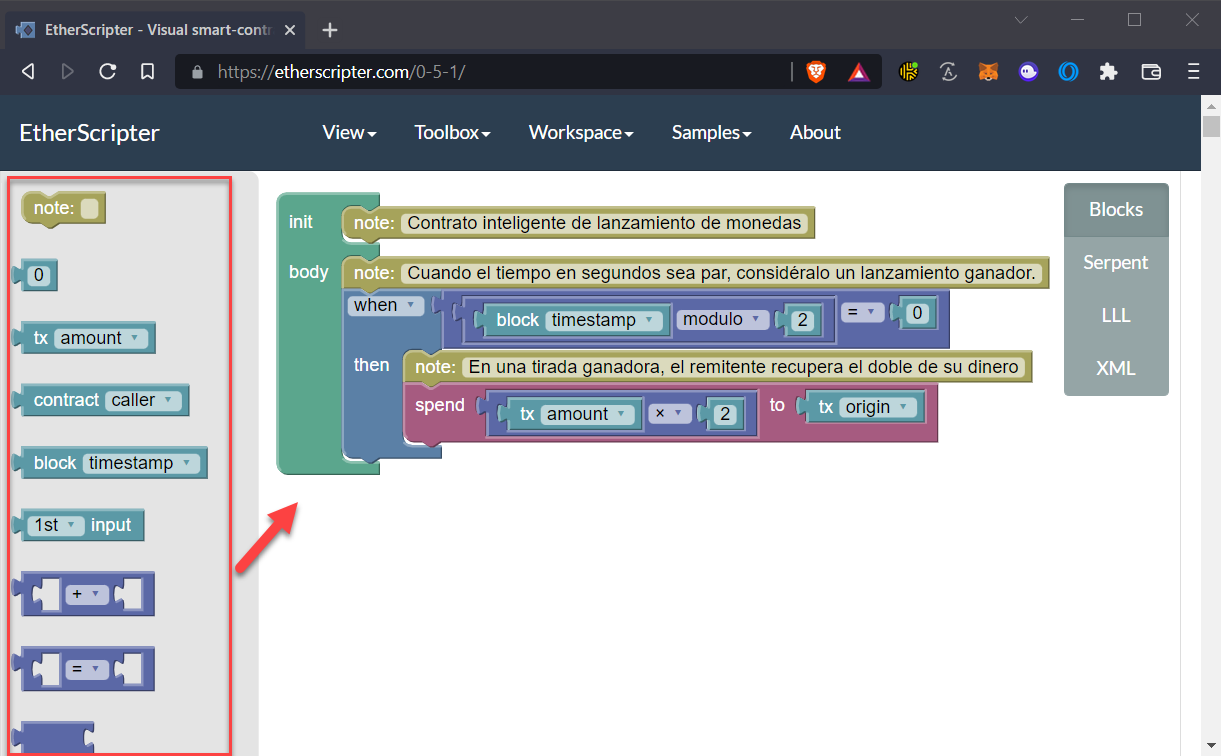
De acuerdo con lo anterior, se puede determinar que este lenguaje presenta una sintaxis amigable, con una curva de aprendizaje moderadamente compleja y que soporta el paradigma de la programación orientada a objetos, lo que permitirá reducir y reutilizar el código existente.



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 plataformas es <https://etherscripter.com>, como se puede apreciar en la siguiente figura.

**Figura 7**

*Editor visual de contratos inteligentes EtherScripter*



Nota: tomado de [EtherScripter - Visual smart-contract builder for Ethereum](https://etherscripter.com/)

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.

**2.2. 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:





**2.3. 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:



**2.4. 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.





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.

**3. 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.

**3.1. 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 puede ser consultada en el siguiente recurso educativo:



**3.2. Sistemas sin intermediarios**

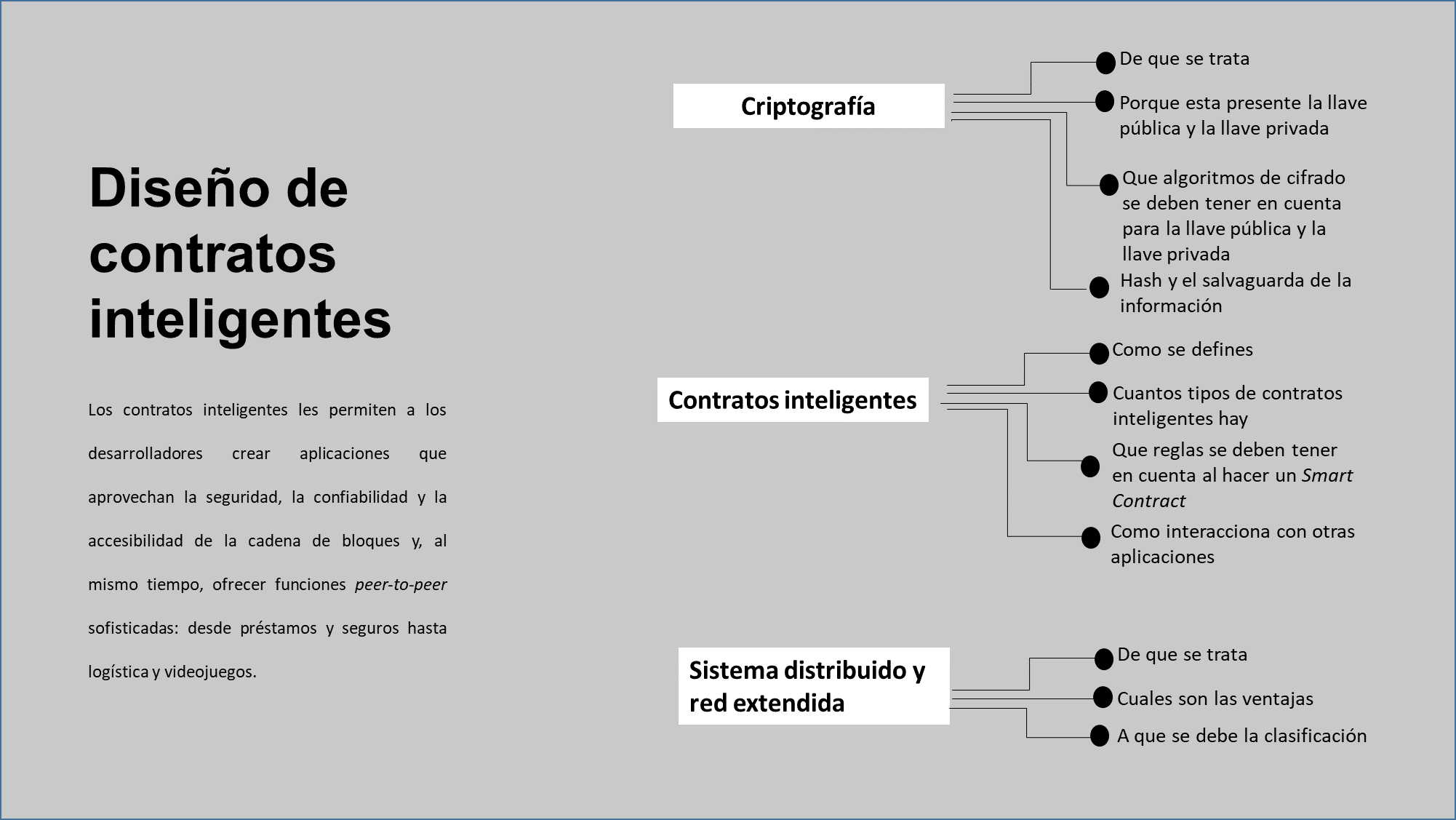
Un contrato inteligente busca no depender de una autoridad central, los contratos inteligentes se ejecutan sin intermediarios. En otras palabras, nadie tiene que apretar un botón, tomar una decisión, equivocarse o simplemente, cambiar de opinión en el último momento. Si las condiciones del contrato se 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*.

**SÍNTESIS**



**c. ACTIVIDADES DIDÁCTICAS (OPCIONALES SI SON SUGERIDAS)**

| **DESCRIPCIÓN DE LA ACTIVIDAD DIDÁCTICA** | |
| --- | --- |
| Nombre de la actividad | Sea parte de los contratos inteligentes |
| Objetivo de la actividad | Identificar conceptos, características y definiciones asociadas a los contratos inteligentes. |
| Tipo de actividad sugerida | Crucigrama |
| Archivo de la actividad  (Anexo donde se describe la actividad propuesta) | Anexo 1. Actividad didáctica |

**d. MATERIAL COMPLEMENTARIO**

| **Tema** | **Referencia APA del material** | **Tipo de material**  **(Video, capítulo de libro, artículo, otro)** | **Enlace del recurso o**  **archivo del documento o material** |
| --- | --- | --- | --- |
| Criptografía - cifrado | ESET Latinoamérica. (2016). *Cifrado de datos: qué es y cómo puede ayudarte a proteger tu información en Internet* [video]. YouTube. <https://www.youtube.com/watch?v=wcJBmoz6Vlk> | Video | <https://www.youtube.com/watch?v=wcJBmoz6Vlk> |
| Documentación oficial del lenguaje de programación Solidity | Solitidy. (2022). *Solidity documentation*. Solitidy. <https://docs.soliditylang.org/_/downloads/en/latest/pdf/> | Manual electrónico | <https://docs.soliditylang.org/_/downloads/en/latest/pdf/> |
| Definición de contrato inteligente | Remix. (s.f.). *File explorers*. Remix. <https://remix.ethereum.org> | Página web | <https://remix.ethereum.org> |

**e. GLOSARIO**

| **TÉRMINO** | **SIGNIFICADO** |
| --- | --- |
| **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. |

**f. 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*.

**g. CONTROL DEL DOCUMENTO**

|  | **Nombre** | **Cargo** | **Dependencia**  ***(Para el SENA indicar Regional y Centro de Formación)*** | **Fecha** |
| --- | --- | --- | --- | --- |
| **Autor (es)** | Hernando José Peña Hidalgo | Experto temático | Regional Cauca, Centro de Teleinformática y Producción Industrial | Abril de 2022 |
| María Inés Machado López | Diseñadora instruccional | Regional Norte de Santander. Centro de la Industria, la Empresa y los Servicios | Abril de 2022 |
| Carolina Coca Salazar | Asesora metodológica | Regional Distrito Capital- Centro de Diseño y Metrología | Abril de 2022 |
| Rafael Neftalí Lizcano Reyes | Responsable Equipo desarrollo curricular | Regional Santander - Centro Industrial del Diseño y la Manufactura | Abril de 2022 |
|  | Julia Isabel Roberto | Correctora de estilo | Regional Distrito Capital- Centro de Diseño y Metrología | Junio de 2022 |

**h. CONTROL DE CAMBIOS**

**(Diligenciar únicamente si realiza ajustes a la Unidad Temática)**

|  | **Nombre** | **Cargo** | **Dependencia** | **Fecha** | **Razón del Cambio** |
| --- | --- | --- | --- | --- | --- |
| **Autor (es)** |  |  |  |  |  |