derecom

ISSN: 1988-2629. Nº. 28. Nueva Época. Marzo 2020-Septiembre 2020

Cómo citar este texto:
Rodríguez Abril, R. (2020). Una aproximación al ordenamiento algorítmico y a su proyección civil, mercantil y financiera. *Derecom*, 28, 01-28. http://www.derecom.com/derecom/

UNA APROXIMACIÓN AL ORDENAMIENTO ALGORÍTIMICO Y A SU PROYECCIÓN CIVIL, MERCANTIL Y FINANCIERA

AN APPROACH TO THE ALGORITHMIC LEGAL ORDER AND TO ITS CIVIL, TRADE AND FINANCIAL PROJECTION

© Rubén Rodríguez Abril
Universidad de Sevilla (España)
r_r_abril@hotmail.com

Resumen
Este artículo pretende explorar hasta qué punto es posible expresar normas jurídicas utilizando procedimientos estrictamente basados en algoritmos. Se introduce un nuevo concepto, el del ordenamiento algorítmico, al que se compara con el ordenamiento jurídico, y del que se describen sus principales características. A lo largo del artículo se exponen someramente las diferentes instituciones vigentes en las *blockchains*, relacionadas fundamentalmente con el ámbito financiero y el tráfico de los derechos reales, como las criptomonedas, los tokens o los contratos inteligentes. Por último, se ofrece una exposición de los supuestos de interacción entre los protocolos distribuidos y el mundo jurídico.

Summary
In this paper we intend to analyze to which extent it is possible to express legal rules using strictly algorithmic procedures. A new concept, the algorithmic legal order, is introduced and compared with the ordinary legal order. Its essential features are described and explored. Throughout the paper, the main blockchain-law institutions (mainly related with the financial sphere and the transfer of rights *in rem*), like cryptocurrencies, tokens and smart contracts, are outlined. Finally, a description of the interaction instances between distributed ledgers and the ordinary law is also offered.

Palabras clave: Blockchain. Contrato inteligente. Ethereum. Bitcoin.

Keywords: *Blockchain. Smart contract. Ethereum. Bitcoin***.**

1.Introducción

En vísperas de la Feria de Muestras de Hannover del año 2011, tres exponentes de la política, la economía y la ciencia alemanas, Henning Kagerman, Wolf-Dieter Lukas y Wolfgang Wahlster, publicaron en prensa un artículo conjunto[1] en el que acuñaron por primera vez una expresión,

Recibido: 23/10/2019 Revisado: 30/01/2020
Aceptado: 05/02/2020 Publicado:15/03/2020

Cuarta Revolución Industrial, que en los años siguientes acabaría popularizándose en el mundo entero. Los autores, siguiendo muy de cerca al economista estadounidense Jeremy Rifkin, consideraban que el progreso técnico y material en Occidente a lo largo de los últimos trescientos años se había estructurado en cuatro revoluciones industriales sucesivas: La *primera* de ellas supuso la introducción de métodos automatizados de producción (p.e. telares mecánicos), propulsados por la máquina de vapor y con el carbón como principal fuente de energía. La *segunda*, apoyada en buena medida por la electrificación y la invención del motor de explosión, supuso la creación de nuevos modelos de organización industrial como el fordismo, la producción en cadena o el taylorismo.

La *tercera* de estas revoluciones impulsó aún más la automatización de los procesos de producción con la aparición del transistor, los circuitos integrados, la robótica y finalmente sistemas informáticos que permitían controlar la producción industrial a distancia, como, por ejemplo, los sistemas SCADA, que tienen aplicación en áreas tan dispares como la generación de electricidad o la depuración de aguas. En último lugar, la incipiente *cuarta* revolución industrial a la que se referían los tres autores vendría caracterizada por el desarrollo de *sistemas ciberfísicos* que canalizarían y potenciarían la interacción del ciberespacio con el espacio físico ordinario. El desarrollo de estos sistemas ciberfísicos vendría posibilitado, entre otras cosas, por la miniaturización y generalización de los sensores digitales, el desarrollo de amplios sistemas de almacenamiento de datos (*Big Data*), la creación de algoritmos y sistemas de computación paralela capaces de procesar semejante torrente de información y tomar decisiones al respecto (*Inteligencia Artificial*), etc....

Las tecnologías de la llamada Cuarta Revolución Industrial se articulan en torno al concepto de ***ciberespacio***, que puede ser definido como el *dominio de realidad virtual integrado por los datos almacenados en computadoras eléctricas de aritmética binaria interconectadas entre sí a través de Internet, por los algoritmos que regulan la evolución dinámica de estos datos y por las estructuras emergentes que surgen a partir de los mismos*. De acuerdo con esta definición, el concepto de ciberespacio abarca dentro de sí tres ámbitos diferentes:

a) El primero de ellos lo constituye la *información almacenada* en computadoras eléctricas, ya sea en circuitos biestables (registros y caché de la CPU), en condensadores (memoria RAM), en la memoria de los discos duros, la memoria *flash* de los dispositivos USB o en cualquier otro dispositivo de almacenamiento de naturaleza estrictamente electromagnética. Los datos almacenados son de carácter discreto (vienen constituidos por ternas de números naturales) y están codificados en aritmética binaria, a saber, en unos o en ceros, en la presencia o la ausencia de voltaje en un determinado circuito, en la existencia o la inexistencia de carga en un condensador individual. A principios de siglo XX surgieron computadoras analógicas como el *analizador diferencial* de Vannevar Bush o la *máquina algebraica* de Leonardo Torres Quevedo, que no operaban con aritmética binaria, sino que lo hacían sobre variables continuas. Sin embargo, debido a la falta de eficiencia de las mismas, el matemático estadounidense Claude Shannon propuso en su tesis doctoral del año 1938 la construcción de un autómata *discreto*, cuyos estados se describieran exclusivamente utilizando números naturales y cuyas operaciones matemáticas se realizasen utilizando el *álgebra de Boole.*[2] El primer ordenador en reunir estas características fue Z3, construido por el ingeniero alemán Konrad Zuse en el año 1943, en plena II Guerra Mundial. La existencia de variables y estados discretos es, probablemente, la principal diferencia entre el ciberespacio y el mundo físico.[3] En el primero, los estados se describen exclusivamente mediante números naturales guardados en la memoria del sistema informático. Así, las pantallas de los ordenadores se estructuran en píxeles indivisibles, cuyo color es determinado por una terna de tres números, que representan gradaciones discretas de los colores básicos rojo, azul y verde, y que se guardan en la denominada memoria de video. Por el contrario, en el Universo físico, el movimiento de los cuerpos se desarrolla a través de un

espacio-tiempo continuo, sin que sea posible compartimentar el espacio o el tiempo en celdillas indivisibles, como sucede con los píxeles. Tal y como mostraba la paradoja de Zenón, el espacio vacío y el tiempo pueden ser divididos y subdivididos tantas veces como queramos, hasta el infinito. Por eso la realidad virtual siempre constituirá una simple aproximación (por muy precisa que sea) de lo que sucede en el espacio ordinario.

b) En segundo lugar, los estados (discretos) de la computadora evolucionan conforme a unas reglas predeterminadas denominadas algoritmos, que se componen de las instrucciones de código máquina a ejecutar por el ordenador. Dichas instrucciones son de tres tipos: operaciones matemáticas cuya práctica corresponde a aquella parte del procesador denominada *unidad aritmético-lógica*, *instrucciones condicionales de flujo* que regulan la estructura, funcionamiento y ejecución de los programas informáticos y, por último, reglas de transmisión de información entre las diferentes partes de un ordenador e incluso entre diferentes ordenadores conectados a una misma red informática (p.e. Internet).

c) En último lugar y en un nivel superior se sitúan las estructuras emergentes surgidas a partir de la información binaria almacenada en estos autómatas. Es en este ámbito donde residen sistemas autorreplicantes como los virus informáticos, los contratos inteligentes, los autómatas celulares, los archivos (binarios, de texto, de imágenes...) y cualesquiera otros patrones de datos que puedan construirse y programarse por medio de un ordenador.

La definición que hemos proporcionado excluye al soporte físico del ciberespacio y se centra exclusivamente en su aspecto meramente informacional: del mismo modo que el concepto de *mente* no engloba al *cerebro* físico, consideramos que la noción de ciberespacio debería ser asimismo independiente del soporte físico de la información digital. De este modo, los cilindros de un disco duro no formarían parte del ciberespacio, pero sí que lo haría la información almacenada magnéticamente dentro de los mismos.

La principal característica que comparten las tecnologías de la Cuarta Revolución Industrial es la cada vez mayor complejidad de las estructuras existentes en el ciberespacio y la creciente interacción de estas estructuras con el Universo ordinario. Gracias a las tecnologías de tratamiento y almacenamiento masivo de datos (*Big Data*), una ingente cantidad de información (del orden de varios *zettabytes*) es recolectada cada segundo desde el exterior e ingresa dentro del *dominio cibernético* por medio de múltiples medios como cámaras, sensores de temperatura, señales de posicionamiento GPS, WiFi, Bluetooth, o etiquetas RFID, por poner sólo unos pocos ejemplos. Tras ello, dentro del propio ciberespacio toda esta información es validada y procesada por medio de algoritmos. Estos algoritmos toman una decisión (desplazar hacia arriba un contenedor de mercancía, hacer girar un coche sin conductor a la izquierda, alterar la ruta de un dron) que es transmitida a un sistema mecánico que altera de nuevo la configuración del espacio ordinario, completándose de este modo un bucle en cuya virtud las estructuras emergentes cibernéticas influyen en el medio físico y viceversa.

Dentro del ámbito del naciente *derecho cibernético* gozan de un papel fundamental tecnologías como la computación distribuida, la criptografía asimétrica o la cadena de bloques (*blockchain*). Históricamente, el pago fue la primera especie de negocio jurídico que pudo perfeccionarse por medios estrictamente electrónicos, toda vez que las tarjetas de crédito aparecieron en el mercado a finales de los años 50 y que las bases de datos bancarias han sido progresivamente digitalizadas a lo largo de las últimas décadas. La invención de procedimientos de encriptación asimétrica -como el sistema RSA o la criptografía de clave elíptica- posibilitó la

ejecución de órdenes de pago a través de redes informáticas abiertas y no seguras (p.e. Internet), así como la identificación de personas físicas y jurídicas por medio de documentos electrónicos denominados *certificados X.509*, que analizaremos en secciones subsiguientes de este mismo artículo.

El siguiente paso en la informatización del tráfico jurídico lo constituyó la aparición en el año 2008 de la primera criptomoneda de la historia, el Bitcoin, y de la tecnología asociada a la misma, que pronto comenzó a ser conocida con la expresión inglesa de *blockchain* (cadena de bloques). Aunque la posibilidad de generar, perfeccionar y consumar contratos dentro de una red informática ya había sido sopesada por primera vez por Nick Szabo en un célebre artículo suyo del año 1997, no es hasta el surgimiento de la red Ethereum en el año 2014 cuando se materializa por primera vez la posibilidad de ejecutar *contratos inteligentes* en el seno de una máquina virtual instalada en los nodos de una red de *blockchain*.

2.El ordenamiento algorítmico: definición y características

Contratos inteligentes y criptomonedas son las piezas básicas en torno a las cuales está comenzando a articularse una nueva institución a la que venimos en llamar ***ordenamiento algorítmico***, y que puede ser definida como el *conjunto de algoritmos, residentes en la capa de aplicación, que regulan el funcionamiento y la modificación del estado de máquina de una base de datos distribuida*. La tecnología de protocolo distribuido (*DLT -Distributed Ledger Technology*- en sus iniciales inglesas), de la que el *blockchain* es una variedad, ha permitido el almacenamiento en bases de datos distribuidas[4] de la información relativa a la titularidad de criptomonedas y de derechos subjetivos existentes exclusivamente en el plano cibernético (*tokens*). Además, redes como Bitcoin y Ethereum han posibilitado el diseño y la ejecución de pequeños programas de ordenador (algoritmos) que modifican de un modo automatizado y predeterminado los derechos subjetivos almacenados en las mismas.

Estos algoritmos, ejecutados por todos y cada uno de los nodos de la red, guardan una gran analogía con las normas jurídicas: los primeros son reglas de comportamiento automatizado impuestas a sistemas informáticos mientras que las segundas son patrones de conducta impuestos a los sujetos de una determinada comunidad y caracterizados por las notas de *generalidad*, *imperatividad* y *coercibilidad*. Las normas jurídicas son *generales* dado que su cumplimiento se impone a todos los individuos de una comunidad; del mismo modo, los algoritmos que regulan el funcionamiento de las transacciones de criptomonedas y la ejecución de contratos inteligentes son de aplicación en todos y cada uno de los nodos que integran la red. Las normas jurídicas son también *imperativas* y *coercibles*, en la medida en que no son simples recomendaciones, sino mandatos a cumplir por los individuos, y coercibles por el Estado en caso de incumplimiento. Este requisito parece ser cumplido también por las normas reguladoras de las cadenas de bloques y por los contratos inteligentes, que son *autoejecutantes,* es decir, su ejecución discurre automáticamente, sin necesidad de intervención humana alguna.

Figura 1: Comparación entre los ordenamientos jurídico y algorítmico.
Fuente: Elaboración propia

Ordenamiento jurídico	Ordenamiento algorítmico
Unidad	Ausencia de partición
Coherencia	Consistencia
Plenitud	Plenitud de Turing
Sujetos de derecho	Direcciones
Personas físicas	Cuentas externas (EOAs)
Personas jurídicas	Organizaciones autónomas descentralizadas (DAOs)
Derechos subjetivos	*Tokens*
Monedas de curso legal	Criptomonedas
Transferencias de derechos	Transacciones regulares (*regular transactions*)
Adquisiciones originarias	Minado de criptomonedas, creación de *tokens*
Contratos	Contratos inteligentes
Celebración de un contrato	Transacción de despliegue (*deployment transactions*)

El jurista italiano Santi Romano definió al **ordenamiento jurídico** en su obra homónima de 1917 como aquel *conjunto de normas jurídicas vigentes en un determinado territorio, y caracterizado por las notas de unidad, coherencia y plenitud*. Estas tres últimas características son aplicables también al ordenamiento algorítmico:

El ordenamiento jurídico está informado por la característica de la **unidad** porque existe una única Norma Fundamental (*Grundnorm*), en la que fundamentan su validez el resto de las normas. De acuerdo con el esquema creado por el jurista austríaco Hans Kelsen, las normas del Estado se estructuran en una suerte de *pirámide normativa*, en cuya base se sitúan las que tienen un rango más bajo y cuya cúspide está coronada por la Constitución, norma suprema del país. El resto de las normas jurídicas adquieren su validez del hecho de haber sido aprobadas conforme a lo establecido por las normas de rango superior y por la Constitución. Esta situación contrasta con lo que sucedía en el Antiguo Régimen, caracterizado por la existencia de múltiples estatutos legales que se yuxtaponían entre sí: en el Toledo medieval convivían múltiples sistemas legislativos como el Fuero Viejo de Castilla, el Fuero Juzgo, la ley mosaica o el derecho islámico, amén de los privilegios feudales o del derecho canónico sin que todas estas normas jurídicas formasen un sistema unitario de normas.

Dentro del ordenamiento algorítmico, *normas* son aquellas reglas computacionales en cuya virtud se actualiza la información almacenada en una base de datos distribuida, y que pueden ser de lo más heterogéneo: el código de los contratos inteligentes, el juego de instrucciones que puede perfeccionar la máquina virtual que ejecuta los mismos, las reglas de elección de un líder entre los nodos, las normas de verificación de las transacciones de criptomonedas, las reglas de interacción de un nodo con el resto de los que componen la red, etc... Lo que dota de unidad al ordenamiento algorítmico es la existencia de unas mismas reglas de comportamiento, un mismo protocolo para todos los nodos que componen la red. El código del protocolo hace las veces de norma suprema dentro de la blockchain. A modo de ejemplo, dentro de la red de Bitcoin es el *software* Bitcoin Core el que cumple esta misión, y debe ser instalado en todos y cada uno de los nodos que pretendan integrarse dentro de la cadena de

bloques. Del mismo modo en que existe un único ordenamiento jurídico por cada país, existe un único ordenamiento algorítmico por cada cadena de bloques.

Ocasionalmente, determinadas circunstancias del funcionamiento de la red pueden poner en peligro e incluso acabar con la unidad del ordenamiento algorítmico, dando lugar con ello al surgimiento de dos o más cadenas de bloques permanentemente separadas entre sí y, con ello, a dos criptomonedas diferentes. Entre los supuestos más frecuentes de *ruptura de la cadena de bloque*s podemos señalar los siguientes:

a) Existencia de nodos de la red que se comportan de una manera maliciosa y no obedecen a las leyes del protocolo, bien por actitud dolosa de los administradores, bien por estar infectados por virus, bien por cualquier otra causa. Este fenómeno se conoce en el ámbito de la computación con la expresión de ***problema de los generales bizantinos***, que fue descrito por primera vez por Leslie Lamport, Robert Shostak y Marshall Pease en un artículo suyo del año 1982. Dado un determinado sistema informático compuesto de varios nodos, ¿qué sucede cuando existe información contradictoria entre las diferentes partes de dicho sistema? Los autores asemejaban esta situación a la de un grupo de generales bizantinos que tratan de coordinar el ataque a una ciudad. Estos generales se comunican entre sí exclusivamente a través de mensajeros. Y se planteaba la posibilidad de que alguno de los generales no actuase de buena fe y transmitiese maliciosamente mensajes erróneos a los demás. En ese caso, ¿en qué condiciones sería aún posible para los generales leales alcanzar un acuerdo con éxito? Los autores llegaron a la conclusión de que en el caso de que los servidores de una red de computación distribuida (los generales del ejemplo) pudieran encriptar sus comunicaciones, bastaría que el 51% de ellos se comportaran honestamente para que dicha red pudiera seguir funcionando adecuadamente. La llegada del bitcoin y su algoritmo de *prueba de trabajo* en el año 2009 supuso una pequeña alteración en este planteamiento: la nueva criptomoneda exigía que los nodos de su red que pretendieran realizar innovaciones en la cadena de bloques debían demostrar que habían realizado un determinado cálculo (*prueba de trabajo*) para cuya perfección eran necesarios amplios recursos computacionales. Entonces, para que una red dotada de un algoritmo de prueba de trabajo continúe funcionando adecuadamente, lo determinante no es que los nodos honestos sean mayoría sino que acaparen más del 50% del poder computacional de la red.

b) **Bifurcaciones** (*forks*), que tienen lugar cuando parte de los nodos de la red deciden seguir por su cuenta normas de comportamiento diferentes a las del resto. Si las innovaciones en el protocolo aún son compatibles con las antiguas normas, el fenómeno recibe el nombre de *soft fork* (bifurcación blanda). Pero si las nuevas y las viejas normas son incompatibles (los llamados supuestos de *hard fork*, bifurcación dura), se produce la ruptura definitiva de la cadena de bloques y el surgimiento de dos protocolos, dos bases de datos distribuidas y dos criptomonedas diferentes. Las criptomonedas surgidas como consecuencia de la bifurcación de una *blockchain* reciben el nombre genérico de *altcoins*, y entre ellas podemos citar ejemplos tan conocidos como Litecoin, que surgió de una escisión del protocolo Bitcoin, o Ethereum Classic, que lo hizo a partir de Ethereum, como consecuencia de la negativa de una parte de los nodos de esta red de reparar los daños provocados por el ataque a *The DAO* en Ethereum, que tuvo lugar en el año 2017.

c) **Ataques eclipse**, por medio de los cuales, actores maliciosos aíslan una parte de una red de la mayoría de los nodos de la misma. El profesor Eric Brewer presentó en el año 2002 su *teorema CAP* (del inglés *Consistency, Avalaibility and Partition Tolerance* -Consistencia, Disponibilidad y Tolerancia a la partición-), afirmando que en el caso de *partición* temporal de una red, si queremos mantener funcionando todos sus nodos (*disponibilidad*), es necesario renunciar a la *consistencia* de la propia red y admitir que haya temporalmente versiones

divergentes del protocolo en cada una de las partes que están temporalmente desconectadas entre sí. Con ello se abriría la posibilidad de que negocios jurídicos considerados válidos por una parte aislada de la red devengan inválidos al recobrar dichos nodos su conectividad con el resto.

En la práctica no es nada fácil mantener la unidad de una base de datos distribuida, y este problema será sin duda alguna uno de los caballos de batalla de los especialistas en tecnología *blockchain* en los años venideros.

La segunda característica del ordenamiento jurídico es la **coherencia**, que implica la necesidad de disponer de reglas de solución de conflictos que determinen cuál es la ley aplicable a un caso concreto, en el caso en que dos de ellas colisionen por contener mandatos contradictorios. Una de estas reglas de solución de conflictos está contenida en el artículo 1.1 del Código Civil español que dispone que "*Carecerán de validez las disposiciones que contradigan otra de rango superior*" ("*lex superior derogat inferiori*", "la ley superior deroga la inferior"). En su virtud, las leyes aprobadas por el Parlamento se aplican con preferencia a los reglamentos (que tienen rango inferior) y la ley escrita es de aplicación preferente a la normativa de origen consuetudinario.

Dado que los ordenamientos algorítmicos están basados en los paradigmas de la computación distribuida y del almacenamiento redundante de la información, en ellos el principio de *consistencia/coherencia* también ostenta una posición fundamental. Los datos se encuentran replicados en todos y cada uno de los nodos de la red y se plantea el problema de cómo difundir de un modo coherente las modificaciones del *estado de máquina del protocolo* y evitar que haya divergencias entre los nodos. Ello se consigue mediante el uso de dos instrumentos fundamentales: los *protocolos deterministas* y los *algoritmos de consenso*. Los protocolos deterministas son aquellos en los que los cálculos y las modificaciones de estado se realizan de una manera predeterminada, de tal manera que a unos mismos datos de entrada les corresponden siempre unos mismos datos de salida, sin que haya lugar alguno para el azar. Con todo, aun empleando un protocolo determinista, el exceso de *latencia*[5] puede provocar el surgimiento de divergencias incluso entre los nodos honestos de la red. Para eliminar estas divergencias se utiliza un algoritmo de consenso, que determina cuál de las versiones del protocolo es la que debe ser aceptada por toda la red. El software *Bitcoin Core* establece que en el caso de que existan dos versiones diferentes de la cadena de bloques, el nodo ha de escoger la que sea más larga (y tenga más bloques). En otros algoritmos, como el célebre Paxos, los nodos han de elegir un líder, que es el encargado de coordinar la red y resolver los conflictos planteados dentro de la misma.

Por último, los ordenamientos jurídicos son **plenos**, en la medida en que sus normas deben dar respuesta a cualquier conflicto que pueda plantearse en el ámbito social. A este respecto, el artículo 1.7 del Código Civil español afirma que los *Jueces y Tribunales tienen el deber inexcusable de resolver en todo caso los asuntos de que conozcan, ateniéndose al sistema de fuentes establecido*. ¿Deben aspirar también los protocolos distribuidos y las normas almacenadas en los mismos a ser completos y a resolver todas las colisiones que puedan plantearse entre los individuos?

En el plano cibernético-algorítmico, cabe señalar que originariamente el protocolo Bitcoin no tenía mayor pretensión que la de permitir y regular transferencias de criptomonedas. El programador ruso Vitalik Buterin, inspirado en un artículo de Nick Szabo del año 1997, concibió por primera en el año 2014 la posibilidad de que los nodos de una *blockchain* pudieran

simular el funcionamiento de un ordenador (denominado técnicamente con la expresión *máquina virtual*) encargado de almacenar y ejecutar *contratos inteligentes* (*smart contracts*) en el ámbito del ciberespacio. Dichos principios dieron lugar al surgimiento de la red Ethereum.

Un rasgo esencial de la arquitectura de la Máquina Virtual de Ethereum (el ordenador virtualizado que ejecuta los contratos inteligentes) es que su juego de instrucciones es *Turing-completo*, y como consecuencia de ello la misma es capaz de simular el funcionamiento de una *computadora universal.*[6] Se dice que un lenguaje de programación es *Turing-completo* cuando permite la creación de *bucles while*[7] dentro de la estructura de sus programas. Esta posibilidad de crear bucles *while* permite a su vez el cómputo de funciones matemáticas denominadas μ-*recursivas* y dota al sistema de una capacidad computacional máxima (obviando las limitaciones de espacio y de memoria). Sin embargo, la *plenitud de Turing* también abre la posibilidad de que el ordenador se quede colgado cuando el bucle deviene infinito y de que el programa sea incapaz de salir de él.

Además, cuanto más versátil es un lenguaje de programación, mayor es la posibilidad de que la estructura lógica de sus programas contenga *bugs* y de que intrusos puedan hacerse con el control de la ejecución de los mismos.

Por este motivo, y para mejorar la seguridad jurídica dentro del ámbito cibernético algunos programadores han sugerido limitar la funcionalidad de las máquinas virtuales de las *blockchains* y restringir el juego de instrucciones que éstas pueden ejecutar. Sin embargo, en mi opinión, si pretendemos aspirar a la construcción de ordenamientos algorítmicos plenos, que permitan la ejecución de los más versátiles contratos inteligentes, es necesario utilizar lenguajes *Turing-completos*, que son aquellos que se corresponden con las gramáticas de mayor expresividad posible dentro de la **jerarquía de Chomsky**: las gramáticas de tipo 0 y 1.

El lingüista norteamericano Noam Chomsky elaboró, en un célebre artículo suyo del año 1956 titulado *Three models for the description of language*, una clasificación de las gramáticas interpretables y reconocibles por dispositivos automáticos, agrupándolas en tres (con posterioridad cuatro) categorías diferentes. Por increíble que parezca existe una correspondencia entre las diferentes clases de autómatas y las diferentes clases de gramáticas, poniendo en conexión disciplinas en principio tan alejadas como la Lingüística o la Teoría de la Computación:

-En un nivel más bajo se sitúan las **gramáticas de Tipo 3** (*regulares*), por medio de las cuales sólo pueden construirse frases con estructuras simples, sin oraciones subordinadas. Las frases construidas por este tipo de gramáticas pueden ser reconocidas e interpretadas por *autómatas de estado finito*, que son aquellos dispositivos automáticos cuyos estados se identifican mediante ternas de números reales, y entre los que se incluyen dispositivos tan variados como los semáforos, los ascensores o los ordenadores. En el plano jurídico, los autómatas de estado finito, pueden utilizarse para documentar derechos subjetivos cuantificables numéricamente, como las obligaciones pecuniarias, los saldos de los depósitos, las cuotas de un derecho de propiedad o el número de acciones pertenecientes a una persona.

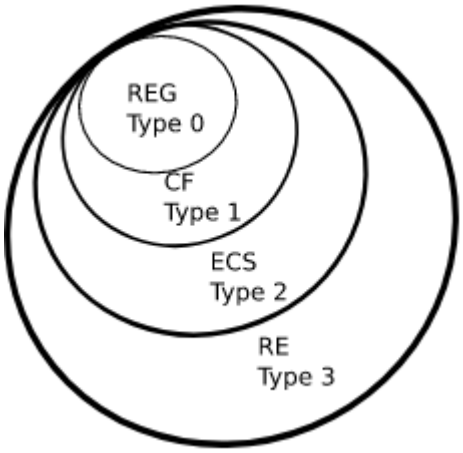
-**Gramáticas de Tipo 2** (*libres de contexto*), por medio de los cuales pueden construirse y elaborarse oraciones subordinadas, que tan importantes son en el ámbito jurídico a la hora de definir los supuestos de hecho de las normas. A modo de ejemplo, la estructura lingüística del artículo 435 del Código Civil español contiene cuatro oraciones subordinadas, cada una dentro de la otra, formando una suerte de estructura anidada: *La posesión adquirida de buena fe no pierde este carácter sino en el caso y desde el momento en que existan actos que acrediten que el poseedor no ignora que posee la cosa indebidamente*. Las gramáticas de Tipo 2 son

computacionalmente equivalentes a los *autómatas con pila*. La exposición del concepto de *autómata con pila* va más allá del alcance de este artículo. Nos limitaremos a señalar que la pila es una suerte de estructura de datos presente en prácticamente todos los procesadores que están en el mercado y que permite la existencia de un fenómeno fundamental dentro de la computación denominado *recursión*, muy útil para los programadores porque les permite definir subrutinas y funciones pero también muy peligroso, porque posibilita la existencia de los denominados *bucles recursivos*, que entre otras cosas sirvieron de base para el ataque a *The DAO* en Ethereum del año 2017.

-Las **gramáticas de Tipo 1** (*sensibles al contexto*) permiten tener en cuenta el contexto a la hora de interpretar las frases y los diferentes sintagmas que componen las mismas. El contexto es esencial dentro del ámbito jurídico. El jurista alemán Friedrich Karl von Savigny consideraba al *criterio sistemático* como uno de los elementos básicos de la interpretación de las leyes. Sin el contexto no sería posible entender expresiones como *la propiedad y el resto de los derechos reales*, *el titular del derecho* o *el cuerpo del delito*. Las gramáticas de Tipo 1 pueden ser reconocidas y *comprendidas* por *autómatas linealmente acotados*, una categoría en la que quedarían englobados la mayoría de los ordenadores actuales.[8] Estos autómatas se caracterizan por que su funcionamiento es similar al de una máquina de Turing, salvo por restricciones en el tamaño de su memoria, que forzosamente debe ser finita (de ahí la expresión linealmente *acotado*). Su lenguaje de programación ha de ser *Turing-completo*.

-**Gramáticas de Tipo 0** (*recursivamente enumerables*): Son las de mayor potencia expresiva posible y se corresponden con las *máquinas de Turing* dentro de la jerarquía de autómatas. Una máquina de Turing es una computadora ideal concebida por el matemático inglés Alan Turing en un célebre artículo suyo del año 1936. Es un autómata de estados finitos, dotado con pila, *y que además* dispone de memoria externa infinita, consistente en una cinta dividida en celdas y de longitud ilimitada. En cada paso, la máquina lee y escribe sobre una celda de la cinta. Tras ello se desplaza una celda a la izquierda o derecha, y finalmente, cambia de estado. Lo que caracteriza a las máquinas de Turing es que son capaces de calcular cualquier función computable (*tesis de Church-Turing*). Las máquinas de Turing se diferencian de los autómatas linealmente acotados en que las primeras disponen de una memoria externa (cinta) infinita, mientras que en los segundos esta última debe ser finita. Sin embargo, tal y como Marvin Minsky señala en su libro *Computation: Finite and Infinite Machines*, el tamaño de la memoria RAM de las computadoras actuales es de tal magnitud (en un ordenador personal, del orden de varios *gigabytes*), que para todos los efectos prácticos la máquina de Turing puede servir como un modelo teórico bastante aproximado del comportamiento de los ordenadores personales ordinarios.

Figura 2: Diagrama de los diferentes tipos de autómatas.
Fuente: Wikimedia Commons



La clasificación que acabamos de describir es de naturaleza jerárquica, de tal manera que las gramáticas situadas en un nivel superior quedan englobadas dentro de las del nivel inferior, formando una suerte de *colección de muñecas rusas*, tal y como se muestra en la imagen.

Con un lenguaje de programación que no es *Turing-completo* (como, por ejemplo, el que interpreta los *scripts* de Bitcoin) sólo pueden construirse determinados tipos de autómatas de estado finito y de autómatas con pila. En otras palabras, sólo pueden *interpretarse* gramáticas de los tipos 3 y 2, pero no de los tipos 0 y 1.

En consecuencia, un lenguaje algorítmico que renunciase a la *plenitud de Turing* estaría sometido a dos grandes limitaciones:

a) No podría construir bucles infinitos y, por lo tanto, los contratos inteligentes no serían capaces de funcionar indefinidamente. Esto es problemático porque dentro de los ordenamientos jurídicos existen muchas situaciones que son en principio perpetuas y están destinadas a perdurar indefinidamente, como sucede con el derecho de propiedad o el contrato de sociedad. Tal y como señala el artículo 63 del *Reglamento n.º 2157/2001 de la Unión Europea*, un contrato de Sociedad Europea sigue vigente hasta que no concurran ninguna de las causas de disolución señaladas por la legislación nacional correspondiente. Una *sociedad cibernético-algorítmica,* constituida por medio de *smart contracts,* se asemejaría a un autómata que funciona indefinidamente hasta la llegada de una serie de datos de entrada (p.e., acuerdo de disolución), cuya lectura provoca la parada de dicho autómata (disolución de la sociedad).

b) El lenguaje debería prescindir por completo de la noción de contexto, y por lo tanto del criterio *sistemático* de interpretación de las normas jurídicas.

Por todos estos motivos, considero que todo ordenamiento algorítmico que aspirase a ser pleno, universal y versátil debería permitir la construcción de sus contratos inteligentes mediante lenguajes *Turing-completos*, aun a riesgo de incrementar las posibilidades de *corrupción de su estructura lógica* (una forma de ataque informático) por parte de actores maliciosos.

En el fondo, la discusión expuesta en los párrafos anteriores enlaza con la vieja problemática de la computabilidad del conocimiento y de los razonamientos humanos, analizada en su día por filósofos como Ramón Llull o Gottfried Leibniz. Este último consideraba que era posible expresar los pensamientos humanos por medio de un lenguaje artificial y simbólico, al que vino a denominar *characteristica universalis* y que no debía ser muy diferente a los sistemas de escritura de tipo ideográfico, como el chino o el egipcio.

Los razonamientos y los silogismos operados sobre este idioma se realizarían por medio de un sistema de cálculo denominado *calculus ratiocinator*, que realizaría operaciones aritméticas y de permutación sobre los símbolos, de un modo no muy diferente al que tiene lugar dentro del álgebra clásica. Leibniz expresaba su esperanza de que las disputas filosóficas y religiosas pudiesen resolverse definitivamente de un modo cáustico, objetivo y conciso mediante la aplicación de procedimientos algorítmicos predeterminados, consistentes en la concatenación de operaciones de cálculo numérico/simbólico.

Cabría preguntarse si las instituciones jurídicas (que no dejan de ser creaciones del razonamiento abstracto) podrían ser descritas también por medio de una suerte de *characteristica universalis*, y si los razonamientos de los jueces y demás operadores jurídicos son susceptibles asimismo de ser reemplazados mediante la aplicación algebraica de un *calculus ratiocinator*. En resumidas cuentas, ¿puede ser la esencia del derecho capturada exclusivamente mediante procedimientos algorítmicos? ¿Podemos formular normas jurídicas mediante algoritmos?

En este artículo no pretendemos dar una respuesta definitiva a esta cuestión. Nos limitamos a afirmar que si efectivamente quisiéramos construir normas jurídicas y contratos de naturaleza algorítmica, lo más recomendable sería utilizar una gramática lo más potente posible dentro de la jerarquía de Chomsky, que correspondería a las de nivel 0 ó 1, utilizando para ello un lenguaje de programación Turing-completo.

3.Los riesgos de la informatización del tráfico jurídico

A menudo, desde sectores *criptoanarquistas* se dibuja una visión quizá demasiado idealizada del ciberespacio, como una suerte de dominio libre donde los ciudadanos pueden desarrollar sus actividades sin intromisión alguna del poder estatal. Esta filosofía está detrás de la creación del Bitcoin y de otras criptomonedas. Supuestamente, la aparición del *blockchain* y de los protocolos distribuidos habría permitido garantizar la inmutabilidad de la información almacenada en sistemas informáticos y automatizar tareas asignadas hasta ahora a funcionarios públicos, disminuyendo con ello la posibilidad de corrupción, siempre alta, en ciertos países en desarrollo. Aparentemente, el uso de la criptografía garantizaría el anonimato de los ciudadanos dentro del ciberespacio y la posibilidad de realizar actividades prohibidas por la Ley, sin temor a persecución alguna por parte del aparato estatal.

Sin embargo, un examen más circunspecto demuestra que este tipo de enfoques quizá pequen de una cierta ingenuidad, pues como ya se señaló en apartados anteriores, en las *blockchains* que funcionan mediante un sistema de prueba de trabajo, el control de la red corresponde a los nodos que ostentan más del 51% del poder computacional de la misma (algunos estudios demuestran que basta con tener tan sólo el 30% del *poder de hash* para que un ataque contra la red pueda tener ciertas posibilidades de éxito). No hay democracia en el seno de la red de Bitcoin, sino que el peso de cada servidor viene determinado por su poder computacional en *FLOPS*.[8] Además, la aparición de chips especializados en el minado de criptomonedas y el surgimiento de *pools de minería* (agrupaciones de mineros) está centralizando el control de la red de Bitcoin en muy pocas manos. Se estima que en la actualidad (octubre de 2019), más del 70% del *poder de hash* de la red de Bitcoin pertenece a servidores situados en la República Popular China, particularmente, en la provincia de Cantón.[9]

A mi juicio, desde el punto de vista de la seguridad informática, la digitalización del tráfico jurídico presenta los siguientes riesgos, relacionados con el robo de claves privadas y la consiguiente transferencia (involuntaria) a terceras personas del poder de disposición sobre derechos subjetivos:

a) Existencia de estructuras matemáticas ocultas dentro de las funciones que sirven de base para los procedimientos de encriptación, como las funciones elípticas. La criptomoneda Bitcoin fue creada por una persona o grupo de personas que operaban con el pseudónimo de Satoshi Nakamoto y cuya identidad tal vez nunca conozcamos. Lo que sí sabemos en la actualidad es que el algoritmo SHA-256, base de esta criptomoneda, fue estandarizado por la NSA (*National Security Agency*, Agencia Nacional de la Seguridad), y que esta misma institución -según las revelaciones de Edward Snowden- debilitó deliberadamente algunos de los estándares de encriptación que desarrolló, plantando a sabiendas vulnerabilidades en los mismos.[10] Los matemáticos que trabajan para los servicios secretos criptológicos descubren a menudo estructuras matemáticas cuya existencia es clasificada como secreto oficial y permanece sin divulgar durante años. El experto británico en Teoría de Números Clifford Cocks, que trabajaba para el servicio criptográfico del Reino Unido -el GCHQ- diseñó a principios de los años 70 un algoritmo de encriptación asimétrica muy parecido a RSA, que fue mantenido en secreto durante 24 años. Si algún servicio criptológico hubiera encontrado algún procedimiento secreto para realizar de un modo efectivo *ataques de colisión* o de *preimagen* al algoritmo SHA-256, no tendríamos manera alguna de saberlo. Si así fuera, ese servicio secreto estaría en condiciones de tomar el control de la red de Bitcoin y cualquier otra *blockchain* que utilizara el algoritmo SHA-256 como base de su *prueba de trabajo*.

b) Infección del sistema informático por parte de *keyloggers*, que son programas residentes capaces de detectar las teclas que el usuario pulsa en el teclado, con la consiguiente posibilidad de que las contraseñas sean secuestradas por actores maliciosos.[11]

c) Fallos de seguridad en las CPU de los ordenadores personales, los móviles, los servidores y los centros de datos. Conocidos son los casos de las vulnerabilidades Spectre y Meltdown, que afectaron a procesadores con *arquitectura x86* (AMD en el caso de Spectre, Intel en los dos casos), y que fueron hechos públicas a principios del año 2018. Ambas permitían a muchas aplicaciones de usuario acceder a regiones prohibidas de la memoria RAM y con ello a información tan delicada como las claves privadas y las contraseñas de los usuarios.

d) Posibilidad de que los servidores sean infectados por *ransomware* o cualquier otro *software* malicioso especializado en secuestrar los contenidos de bases de datos. Es verdad que el almacenamiento de la información en una *blockchain* se realiza de una manera redundante, y que la información se encuentra replicada en todos los nodos de la red. Sin embargo, algunas *blockchains* privadas no están diseñadas con toda la diligencia que se debiera, ya que en ellas se exige que todos los servidores tengan el mismo tipo de procesador y de sistema operativo, lo que sin duda alguna facilitaría la replicación de gusanos en caso de infección, incrementando con ello las posibilidades de éxito de un *ataque del 51%* sobre la *blockchain*.

e) Escasa presencia de entropía dentro de los servidores encargados de perfeccionar los algoritmos de cifrado. *Entropía* es una palabra multidisciplinar que se usa en diversas áreas de la ciencia. Dentro de las ciencias de la computación se denomina como tal a la *cantidad de azar* de la que el ordenador dispone para generar números puramente aleatorios. En Linux y sus sistemas operativos derivados, la entropía se recolecta a partir de fuentes como las pulsaciones del usuario en el teclado o el ratón y posteriormente se almacena en los archivos */dev/random*

y */dev/urandom*, donde queda a disposición de las diferentes aplicaciones. La imposibilidad de un ordenador de generar números genuinamente aleatorios es un fallo de seguridad bastante grave y fue la causa de que muchos usuarios del monedero Android Bitcoin Wallet perdieran sus fondos en el año 2013[12] [13]. Las *fuentes de azar* (entropía) son escasas dentro de los sistemas informáticos, particularmente si no disponen de teclado y ratón ("servidores *headless"*). Esta escasez puede ser explotada por piratas informáticos para organizar un ataque que he venido a denominar ***ataque por agotamiento de entropía***. En el mismo, los atacantes tratan de forzar al servidor a calcular repetidamente firmas de curva elíptica, para lo cual es necesario generar números aleatorios y gastar la entropía almacenada. Cuando esta última se agota, el servidor puede reaccionar de dos maneras. La primera de ellas es paralizando su funcionamiento, en cuyo caso el ataque sería una variante de un *ataque de denegación de servicio*. En segundo lugar, el servidor puede optar por continuar corriendo, en cuyo caso los números generados a partir de ese momento ya no serían aleatorios, sino pseudoaleatorios. En ellos aparecerían patrones y estructuras que permitirían a los atacantes romper el cifrado y adivinar las claves privadas de los participantes en las comunicaciones.

Ninguno de estos riesgos se da en el caso de los registros públicos que funcionan con soporte papel. La única manera que tendría una potencia extranjera de robar los datos de un protocolo notarial o de los libros del Registro de la Propiedad es enviando soldados para que se hicieran físicamente con el control de los libros. La informatización y *algoritmización* del tráfico jurídico permiten, ciertamente, la agilización del mismo, pero a costa de comprometer la seguridad de la información.

Contra lo que puede suponer el *criptoanarquismo* más romántico, que ve en Internet un espacio de libertad absoluta, el ciberespacio es un ámbito inestable e inseguro, en el que por todas partes acechan posibles atacantes y se percibe continuamente la huella de las grandes potencias tecnológicas y sus servicios secretos.

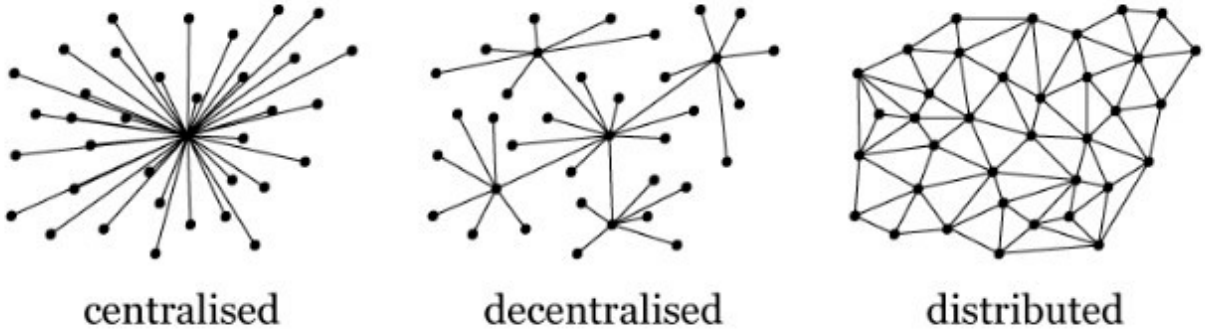
4.Principios del ordenamiento algorítmico

Veamos, a continuación, los ocho principios del ordenamiento algorítmico.

a) Principio de redundancia

El concepto de *redundancia* dentro de la arquitectura de una red informática fue introducido por primera vez por Paul Baran, miembro de la RAND Corporation, en un trabajo suyo del año 1962, en el que exploraba las posibilidades de supervivencia que una red nacional de ordenadores tendría en el caso de un ataque nuclear. Baran introducía en su trabajo la distinción entre redes *centralizadas* y redes *distribuidas* de computación*:* en las primeras todos los nodos se comunican forzosamente a través de un nodo central, que actúa como *único punto de fallo* (*single point of failure*) y cuya destrucción acarrea la puesta fuera de funcionamiento de toda la red. En las redes *distribuidas,* por el contrario, existen caminos alternativos entre los nodos, que podrían utilizarse en caso de destrucción de la vía de comunicación principal. La *redundancia* es equivalente al número de conexiones medio de cada nodo, para el supuesto de una red idealmente infinita.

Figura 3: Diferentes tipos de topología de red, según Paul Baran
Fuente: Wikimedia Commons



El trabajo de Baran fue una de las referencias clave en el diseño de las primeras redes informáticas de escala nacional dentro de EEUU, como ARPANET o PRNET que, a partir del otoño de 1977, darían origen al nacimiento de Internet. El concepto de redundancia también hizo fortuna dentro de las bases de datos: a mediados de los años 80 comenzaron a comercializarse las primeras *bases de datos distribuidas* en las que todo su contenido se hallaba replicado en varios servidores, todos los cuales habrían de proceder de un modo coordinado a la actualización de la misma, lo que se hacía mediante un administrador único. *Blockchain*, surgida a partir de la aparición del Bitcoin, es una modalidad de base de datos distribuida en la que sus transacciones son ensambladas por medio de criptografía en bloques, que a su vez se integran en una única cadena. No existe ningún administrador centralizado que pueda actuar como *único punto de fallo*, sino que los cambios en la base de datos son consensuados entre todos los nodos.

b) Principio de descentralización

Hasta ahora, en la mayoría de los derechos continentales, la transmisión de los derechos sobre propiedad inmobiliaria, la creación de sociedades mercantiles, así como la creación del dinero y la concesión de crédito han sido controlados por instancias centralizadas. En algunos casos, por funcionarios públicos, como los notarios o los registradores de la propiedad. En otros, por bancos centrales que monopolizan la creación y la emisión de dinero y por sistemas de pago dependientes de cámaras de compensación, como los sistemas EURO1 o STEP2, operados por la EBA Clearing Company, y que han sido declarados como *"sistemas de pago de importancia sistémica"* por parte del Banco Central Europeo. En los nuevos sistemas descentralizados que han emergido durante la última década, la seguridad de las transacciones es garantizada por una red descentralizada de nodos, que se comunican y se ponen de acuerdo entre sí a través de un protocolo de consenso, que impide que los datos sean alterados de un modo malicioso por terceras personas.

c) Principio de anonimato

Dentro de cada ordenamiento algorítmico, la identidad real (personal) de los intervinientes no se revela, sino que éstos se identifican por un simple número denominado *dirección* o *clave pública*. La aparición de sistemas de criptografía asimétrica a partir de los años 70 permitió la construcción de estructuras de transmisión de información completamente anónimas como la red Tor o el protocolo Bitcoin. En los sistemas de criptografía asimétrica, el ordenador -utilizando

algoritmos de Teoría de Números u operando con funciones elípticas- genera dos números entrelazados entre sí: las claves pública y privada. La clave pública se utiliza para identificar a sujetos, y a partir de ella se derivan matemáticamente las direcciones Bitcoin y Ethereum. La clave privada permite firmar y perfeccionar negocios jurídicos válidos dentro de la *blockchain*. La clave pública identifica al sujeto. La clave privada le otorga el *ius disponendi* sobre un derecho.

d) Principio de celeridad

Mientras que en determinados registros administrativos las inscripciones pueden demorarse por días o incluso por meses, en el protocolo Bitcoin, cada 10 minutos un nuevo bloque es incorporado a la cadena. En Ethereum y Ripple este periodo de tiempo se reduce a 17 y 4 segundos respectivamente. El principio de celeridad trata de ser potenciado por las denominadas capas *de segundo nivel* (*Layer 2*), como Lightning Network, que es una estructura apoyada en la red de Bitcoin, y que permite perfeccionar transmisiones denominadas en esta criptomoneda en contados segundos.

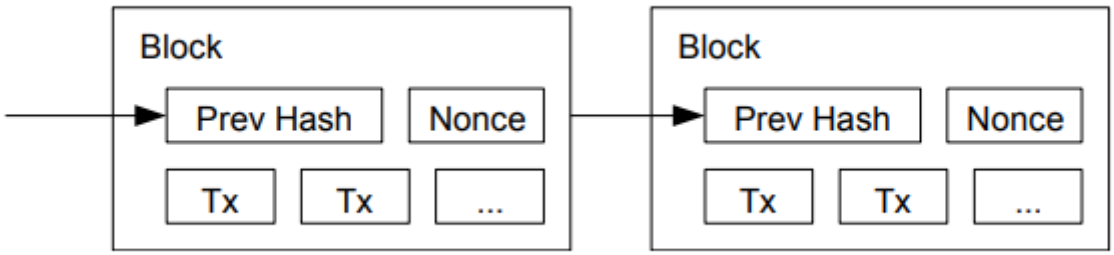
e) Principio de inscripción constitutiva

Cualquier modificación de un derecho subjetivo debe implicar una correlativa modificación del *estado de máquina* del protocolo distribuido. Es equivalente al viejo principio latino *quod non est in actis non est in mundus* (*lo que no está en el Registro no está en el mundo*), vigente en sistemas registrales como el alemán o el suizo, que imponen una identidad absoluta entre el contenido del *Grundbuch* (Registro de la Propiedad) y la realidad jurídica extrarregistral.

f) Principio de tracto sucesivo criptográfico

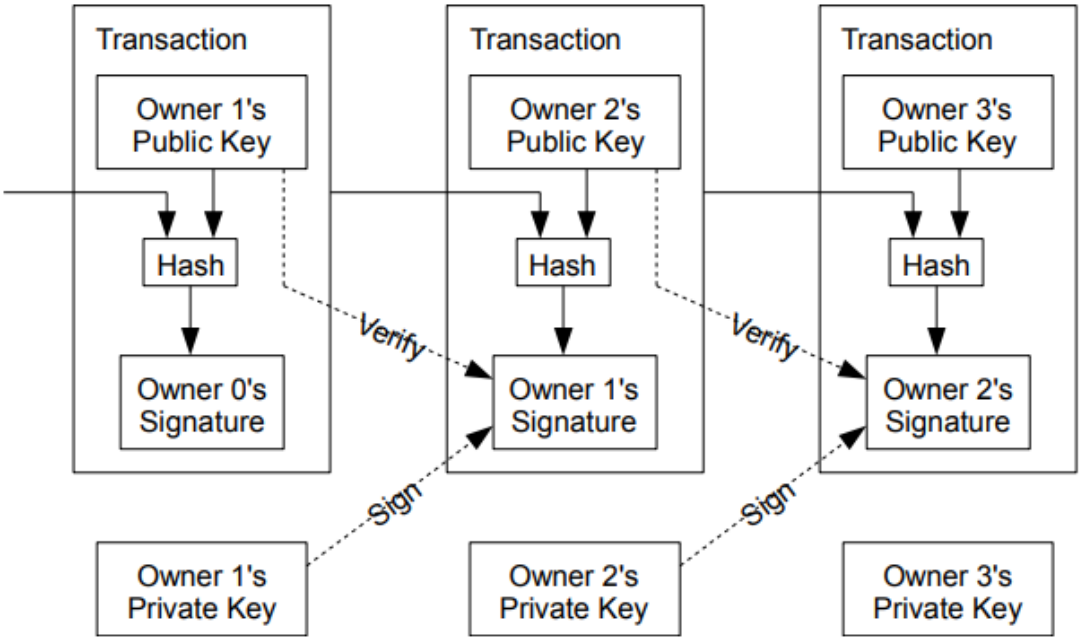
Dentro de cada *blockchain*, los bloques se ordenan cronológicamente, y cada uno de ellos contiene el *hash* del bloque inmediatamente anterior, formando una suerte de ***tracto sucesivo criptográfico*** en cuyo inicio se sitúa el *bloque del Génesis* (el primer bloque del protocolo *Bitcoin*, aparecido en el año 2009), y en cuyo final se halla el bloque más reciente. Este principio de tracto sucesivo remonta su existencia a la creación en los años 50 de las *listas enlazadas*, estructuras de datos en las que cada uno de sus elementos contenía una referencia al elemento inmediatamente anterior desde un punto de vista cronológico. En las protocolos distribuidos de criptomonedas, surgidos a partir del año 2009, esta referencia es de naturaleza criptográfica: se trata del *hash* del bloque anterior.

Fig. 4: Tracto sucesivo de bloques dentro de la cadena Bitcoin
Fuente: Bitcoin's white paper



En el protocolo *Bitcoin* y sus criptomonedas derivadas (*Litecoin*, *Zcash*), el principio de tracto sucesivo existe también en el ámbito de las transacciones individuales, cada una de las cuales incorpora el *hash* de la transacción anterior de la que trae causa, formando una suerte de cadena que comienza con una *transacción de acuñación* (adquisición originaria de criptomonedas por parte del *minero* que ensambla un bloque) y concluye en una transacción aún no gastada.

Fig. 5: Tracto sucesivo de transacciones dentro de la cadena Bitcoin
Fuente: Bitcoin's white paper



En Ethereum, por el contrario, las transacciones no están enlazadas entre sí y sólo contienen una referencia a la cuenta de la persona que ha realizado la transacción. Si el saldo del transmitente es suficiente, la transacción queda validada. De lo contrario, los nodos rechazarán su validación en virtud del principio ***nemo dat quod non habet*,** de un modo parecido a lo que sucede con las transacciones bancarias.

g) Principio de publicidad formal

La información relativa a los derechos y a las transacciones de derechos está disponible para cualquier persona que tenga acceso a un servidor que albergue toda la cadena de bloques. La publicidad formal es facilitada por utilidades como Etherscan que permiten realizar consultas sobre direcciones electrónicas, transacciones individuales, *tokens* y cualesquiera otras actividades realizadas dentro de la red.

Algunos protocolos como *Monero* tratan de eliminar parcialmente este principio mediante un fenómeno denominado *ofuscación del protocolo*, en cuya virtud, los datos relativos a las transacciones (participantes, importe…) quedan ocultos por medio de diversas técnicas criptográficas.

h) Principio de inembargabilidad

Los derechos existentes en los ordenamientos algorítmicos son, en la práctica, inembargables por los tribunales de justicia, toda vez que el *ius disponendi* sobre los mismos está inescindiblemente vinculado a la clave privada del titular. No es posible técnicamente para un Tribunal de Justicia modificar el *estado de máquina* de un protocolo distribuido si previamente no ha podido recuperar de algún modo las claves privadas del titular del derecho que se pretende embargar.

5.Las instituciones del ordenamiento algorítmico

Dentro de los ordenamientos algorítmicos, son **sujetos** los titulares de criptomonedas, los tenedores de otros derechos representables mediante asientos electrónicos (*tokens*), así como las partes intervinientes en un contrato inteligente.

a) Sujetos de derecho

Por lo general, como hemos visto en secciones anteriores, cada sujeto está identificado por una *dirección* que deriva de una *clave pública*, mientras que el poder de disposición sobre el derecho se deriva de la posesión de la *clave privada*, matemáticamente entrelazada con la primera. Sin embargo, existen excepciones a esta norma, y no siempre es posible identificar a los sujetos dentro de una cadena de bloques mediante una dirección. Esto es lo que sucede en las transacciones *bitcoin* del tipo *P2SH* (*Pay to Script Hash*), en las que el destinatario de las criptomonedas no es una dirección electrónica, sino cualquier persona que presente un *script* cuyo *hash* se corresponda con el almacenado, y cuya subsiguiente ejecución retorne un valor de *verdadero* (esto es, ejecución satisfactoria). Por otro lado, algunos protocolos como *Monero* dan un paso más allá en la anonimización de sus transacciones y utilizan para ello *direcciones furtivas* (*stealth addresses*), usadas exclusivamente en una única transacción (y por lo tanto inútiles para realizar rastreos) y *firmas de anillo*, para cuya práctica se usan las claves públicas de varias personas, sin que se sepa quién de ellas ha firmado realmente con su clave privada.

En otras ocasiones, la titularidad de los derechos corresponde a varias personas (**situaciones de cotitularidad)**, requiriéndose la firma de todos o algunos de ellos para la perfección de negocios dispositivos sobre los mismos. Es el caso de las transacciones de bitcoins del tipo P2MS (*Pay to Multisignature*), también llamadas transacciones *multifirma*, cuya validez requiere de la firma de un número mínimo de cotitulares.

Por último, el protocolo de Ethereum distingue entre dos tipos de propietarios de cuenta: El primer tipo es el de los usuarios externos, que son aquellas **personas físicas** situadas fuera de la *blockchain* que pueden controlar y disponer de las criptomonedas de una cuenta, por medio de una clave privada. En segundo lugar, las cuentas pueden pertenecer a organizaciones autónomas descentralizadas (*Decentralized Autonomous Account*, DAO), que conforman un concepto análogo al de **personas jurídicas**: son patrimonios autónomos cuyo funcionamiento no es controlado por un usuario externo, sino por el código de un contrato inteligente, que opera de una forma completamente autónoma.

b) Criptomonedas

El dinero puede definirse como aquel *medio de pago de aceptación general por parte de los individuos de una comunidad*. Dentro de esta definición quedan incluidas no sólo las monedas de curso legal en un determinado país, sino también las monedas locales (como las vigentes en ciertos barrios de Londres) y las criptomonedas.

El dinero puede consistir en objetos tangibles (como el oro o la plata), en billetes, monedas o, incluso, en simples anotaciones en registros electrónicos. En la actualidad, más del 90% del dinero viene constituido por anotaciones electrónicas en registros electrónicos centralizados, localizados en bancos comerciales, cámaras de compensación o bancos centrales.

Durante la segunda mitad del siglo XIX, en Europa se generalizó el uso de billetes en amplias capas de la población. Se hizo bajo los esquemas de *patrón oro* o *patrón bimetálico*: el dinero circulante venía determinado por la cantidad de metal precioso almacenado por un estado dentro de sus reservas. Para que un estado pudiera emitir papel moneda, era preciso que dicha emisión estuviese respaldada por la correspondiente adquisición y almacenamiento de metal precioso. La emisión de dinero estaba limitada por la tenencia de metales preciosos por parte del estado que llevaba a cabo la acuñación de moneda.

Progresivamente, a lo largo del siglo XX se fueron generalizando en Occidente los sistemas basados en el dinero *fiat*, no respaldado por oro o plata, que no tenía valor intrínseco y cuya utilidad derivaba del hecho de haber sido declarado como medio de pago oficial por la legislación de un determinado país. Los sistemas de dinero *fiat* comenzaron a implantarse en Europa, particularmente en Alemania, durante el periodo de entreguerras, y se asentaron definitivamente en Occidente a partir del año 1971, fecha en que Richard Nixon anuló la convertibilidad del dólar en oro y destruyó definitivamente el sistema monetario de Bretton-Woods. Las monedas occidentales dejaron de estar respaldadas por el oro, iniciándose a partir de entonces el proceso de *digitalización del dinero*. En este momento, casi cincuenta años después del *shock de Nixon*, más del 90% del dinero circulante, tanto en Estados Unidos como en la Eurozona, viene constituido por simples anotaciones electrónicas realizadas en servidores informáticos. Menos del 10% del dinero circulante es dinero físico, y es probable que este último porcentaje descienda en los próximos años debido a las presiones existentes para eliminar el dinero en efectivo. Por este motivo, puede decirse sin temor a exagerar que el dinero en la actualidad está formado por simples cadenas de bits (de unos y ceros) existentes en bases de datos de carácter electrónico. Su valor reside en que a estas cadenas de bits se les atribuye legalmente la función de ser medio de pago: Pueden usarse para realizar transferencias bancarias, pagar impuestos, abonar sueldos y para perfeccionar cualquier tipo de transacción comercial. De hecho, la posición del dólar como moneda de reserva internacional se deriva -entre otras razones- del hecho de que el 70% de las transacciones comerciales en el mundo se realizan en esta moneda.

La aparición de las **criptomonedas** en el año 2008 (Bitcoin) constituyó un nuevo hito en el proceso de digitalización del dinero. Pueden ser definidas como aquel *medio de pago consistente en anotaciones electrónicas realizadas en un protocolo distribuido*. Las diferentes partes de este protocolo están enlazadas entre sí a través de funciones criptográficas (de donde deriva su prefijo *cripto*-) denominadas *hashes,* formando estructuras de datos en forma de cadenas o de árboles y otorgando con ello una extraordinaria solidez al sistema, dificultando su

falsificación. Remito al lector interesado, pero poco versado aún en el funcionamiento de las *blockchains*, a mis trabajos anteriores (consúltese la sección bibliografía) publicados en esta misma revista, en los que realizo una breve exposición de los conceptos de criptografía de uso más frecuente en el ámbito de los protocolos distribuidos.

A mi juicio, existen dos grandes diferencias entre las criptomonedas y el dinero emitido por bancos centrales:

- Las criptomonedas no son moneda de curso legal en ningún país.

- Los datos de los titulares de criptomonedas no se almacenan en servidores centralizados, sino que lo hacen de un modo redundante en múltiples servidores interconectados entre sí. En el caso de Bitcoin, toda la información relativa a las transacciones de criptomonedas producidas desde el año 2009 (fecha del bloque del Génesis) se halla replicada en más de 10.000 nodos.

Desde el punto de vista jurídico, las criptomonedas son derechos de contenido patrimonial, valorables en dinero. Y a mi juicio, deben ser clasificadas entre los derechos reales porque reúnen los dos requisitos propios de los mismos: *inmediatez* y *carácter absoluto*. Los derechos reales (como la propiedad) son *inmediatos* porque atribuyen a su titular un poder directo sobre una cosa, que se ejerce sin intermediarios. Este carácter inmediato está presente también en el ámbito de las criptomonedas, puesto que es la posesión de un número o un conjunto de números denominado *clave privada*, el que atribuye exclusivamente a su titular poder de disposición sobre el mismo. El carácter *absoluto* de los derechos implica que éstos se ejercen *erga omnes* y que la información relativa a los mismos también está disponible frente a todos por medio de la publicidad registral. En el caso de las *blockchains* públicas, la información relativa a la transmisión y tenencia de criptomonedas también está a disposición de cualquier usuario que tenga acceso a un nodo de la red.

c) Asientos electrónicos (*tokens*)

Un *token* (“asiento”, en lengua inglesa) es un *asiento electrónico practicado en un libro mayor distribuido, controlado por un contrato inteligente y que representa la tenencia de un derecho subjetivo*. Lo que diferencia a los *tokens* de las criptomonedas (que también son asientos/anotaciones en cuenta) es que la emisión, transmisión e incluso la eliminación de los primeros están controladas por contratos inteligentes, mientras que la emisión de las criptomonedas se realiza mediante el proceso de *minado* y sus transferencias se regulan y validan conforme a las normas genéricas del protocolo.

Dentro de Ethereum, los contratos inteligentes creadores de *tokens* a menudo se someten a la *interfaz ERC20*, un estándar técnico que impone que el contrato inteligente esté dotado -como mínimo- de las siguientes funcionalidades:

-Función *totalSupply*: Obtiene la cantidad total de *tokens* que el contrato inteligente ha puesto en circulación hasta el momento.
-Función *balanceOf*: Obtiene el saldo de *tokens* de cualquier usuario de la red.
-Función *transfer*: Envía una cierta cantidad de *tokens* a un usuario.
-Función *transferFrom*: Transfiere *tokens* de una cuenta de usuario a otra.

-Función *approve*: Permite a un usuario retirar *tokens* de la cuenta de otro usuario, hasta un cierto límite.
-Función *allowance*: Obtiene la cantidad límite de *tokens* que un usuario puede retirar de la cuenta de otro.

Desde un punto de vista estrictamente jurídico, los asientos electrónicos (*tokens)* pueden utilizarse para documentar tanto valores mobiliarios como derechos reales. Con respecto a los primeros, cabe señalar que los *tokens* emitidos en Ethereum conforme a la citada *interfaz ERC20* tienen claramente la naturaleza jurídica de **valores mobiliarios**, dado que reúnen las características esenciales de los mismos, a saber, son emitidos en masad; son representados mediante anotaciones en cuenta (electrónicas y practicadas en la *blockchain*, en este caso); cada anotación *se identifica* con el derecho que representa (este fenómeno jurídico se conoce con el nombre de *incorporación*). Los *tokens* no son sólo simples anotaciones, sino que **también** son derechos; pueden representar tanto acciones como títulos de crédito; son homogéneos. Atribuyen a cada titular el mismo juego de derechos. Esta característica permite que las anotaciones electrónicas puedan documentar *clases* de acciones y *series* de obligaciones; son libremente transmisibles por su titular.

En el caso de los derechos reales, la digitalización del tráfico inmobiliario y su representación por *tokens* son posibles, pero requieren la creación previa de un equivalente en código máquina del conjunto de las disposiciones que regulan la transmisión de la propiedad y demás derechos reales sobre bienes inmuebles y la publicidad registral de los mismos. A fin de cuentas, un Registro de la Propiedad no es más que una base de datos centralizada: Sus *tokens* se practican en papel y reciben el nombre de *asientos registrales.* La creación, modificación y cancelación de éstos se realiza conforme a las normas de la Ley y del Reglamento Hipotecario, que harían las veces de un *smart contract*. Y la labor calificadora del Registrador de la Propiedad sería análoga a la de los algoritmos validadores de las redes de Ethereum o Bitcoin, que determinan cuándo un determinado negocio jurídico se incorpora o no a la *blockchain*.

d) Transacciones

Una transacción es una modificación de una base de datos distribuida, que provoca una alteración en la titularidad de una criptomoneda o de otro derecho subjetivo existente en el seno de la cadena de bloques. Dentro de Bitcoin se distingue entre *transacciones de acuñación* (*coinbase transactions*), que determinan la adquisición originaria de criptomonedas por parte del servidor que acaba de minar un bloque, y *transacciones regulares* (*regular transactions*), de naturaleza derivativa y que provocan el cambio de titularidad de *bitcoins* entre dos direcciones (sujetos) diferentes.

En el marco de Ethereum, y de otras redes que disponen de máquina virtual que ejecuta contratos inteligentes, se incluyen además otros dos tipos de transacciones: *transacciones de despliegue* (*deployment transactions*), que inyectan nuevo código ejecutable (contratos inteligentes) en la *blockchain*, y *transacciones de invocación* (*invocation transaction*), que ejecutan dicho código.

e) Contratos inteligentes

La expresión *smart contract* (*contrato inteligente*) fue acuñada por primera vez por el programador estadounidense Nick Szabo en un artículo suyo del año 1997 en el que exploraba la posibilidad de que determinados contratos celebrados electrónicamente pudieran ejecutarse de un modo automático, como sucedía con las máquinas expendedoras. El protocolo Bitcoin,

puesto en marcha en el año 2009, permitía que las transacciones de criptomonedas quedasen subordinadas a la ejecución satisfactoria de un algoritmo predeterminado, programado en un lenguaje específico, *script*. El programador ruso Vitalik Buterin, inspirado en las ideas de Szabo, propuso reformar el protocolo Bitcoin para ampliar la funcionalidad de su lenguaje *script*, y permitir que con éste pudieran construirse contratos inteligentes tan versátiles como los descritos por Szabo. Dado que la comunidad de Bitcoin no aceptó las propuestas de Buterin, en el año 2014 se fundó Ethereum, una *blockchain* dotada de una máquina virtual (EVM) capaz de ejecutar contratos inteligentes, utilizando para ello un juego de instrucciones de código-máquina *Turing-completo*.

Llegados a este punto, se impone la obligación de distinguir a los ***contratos ordinarios celebrados por vía electrónica*** de los *contratos inteligente*s. Los primeros no se diferencian en su tipología del resto de los contratos, celebrados oralmente, por escrito, o mediante forma pública: consisten en un acuerdo de voluntades entre dos o más personas, por el que acuerdan la constitución o la transmisión de una obligación civil. Su única particularidad es que las declaraciones de voluntad de las partes se han emitido por medios telemáticos, como sucedería en el caso de un contrato de arrendamiento cuyo contenido constase en un documento PDF firmado electrónicamente por ambas partes. Por el contrario, un ***contrato inteligente*** es una *estructura algorítmica capaz de crear y de ejecutar automáticamente una obligación entre una persona y un autómata*. En el ejemplo de la máquina expendedora, su contrato inteligente vendría constituido por el programa que regula la compraventa automática de sus artículos por los particulares. En el caso de Ethereum, la compraventa de *tokens* a través de un contrato inteligente se perfecciona de un modo no muy diferente al de una máquina expendedora: El comprador ha de realizar una transacción a favor de la cuenta que expide y almacena los tokens. En dicha cuenta, hay almacenado un contrato inteligente cuyo código (indexado en la sección *codeHash*) es el que perfecciona la transferencia de *tokens*. La información relativa a los nuevos *tokens* adquiridos queda guardada en la sección de almacenamiento (*account storage contents Trie*).

En resumidas cuentas, los contratos inteligentes son una variedad contractual caracterizada por tres notas principales: a) desde el punto de vista de los **sujetos**, una de sus partes contratantes es un autómata; b) desde el punto de vista de su **contenido**, éste consiste en un algoritmo; c) desde el punto de vista de su **ejecución**, ésta tiene lugar de un modo automatizado.

Evidentemente, los contratos inteligentes tienen plena eficacia en el ámbito cibernético dado que son capaces de inducir por sí mismos cambios en el *estado de máquina* de un protocolo distribuido (carácter *autoejecutante*), pero ¿pueden también servir de base para el surgimiento de una obligación civil, exigible ante los tribunales de justicia?

En Europa, si bien el principio de libertad -que informa buena parte de los ordenamientos civiles europeos- permitía ya desde la Edad Media la perfección de contratos en cualquier forma siempre y cuando su contenido no sea contrario a la ley, la posibilidad de celebrar contratos electrónicamente adquirió carta de naturaleza por medio de la Directiva 2000/31/CE, cuyo artículo 9.1 imponía a los Estados miembros la obligatoriedad de remover los obstáculos que pudieran entorpecer la celebración de contratos por vía electrónica (*Los Estados miembros velarán por que su legislación permita la celebración de contratos por vía electrónica*).

Aunque Kevin Werbach y Nicolas Cornell en su estudio *Contracts ex machina* del año 2017 dudaban de la validez jurídica de los contratos inteligentes en el marco de los sistemas legales basados en el Common Law angloamericano, en mi opinión esta forma contractual puede incardinarse perfectamente dentro de los derechos de obligaciones (de origen romano) vigentes en Europa continental: así, el Código Civil español reconoce el principio de libertad de forma en su artículo 1278, para el cual *los contratos serán válidos, cualquiera que sea la forma en que se hayan celebrado*, mientras que el *Code Civil* francés hace lo mismo en su artículo 1102 (*Chacun est libre [...]de déterminer le contenu et la forme du contrat dans les limites fixées par la loi*). Por su parte, el parágrafo § 128a del *BGB* alemán permite la celebración de contratos por vía electrónica, siempre que la declaración de voluntad de las partes venga acompañada de una *firma electrónica cualificada*. El Reglamento europeo n.º 910/2014 (*Reglamento eIDAS*), en su artículo 25.2, reconoce de un modo general la posibilidad de celebrar contratos por vía electrónica y atribuye a las firmas electrónicas cualificadas (esto es, acompañadas de un certificado criptográfico) el mismo valor que las firmas manuscritas.

Una vez que el contrato ha sido perfeccionado, su contenido tiene el carácter de *lex privata* entre las partes, en los términos del artículo 1091 del Código Civil español (*Las obligaciones que nacen de los contratos tienen fuerza de ley entre las partes*) y del artículo 1103 del Código Civil francés (*Les contrats légalement formés tiennent lieu de loi à ceux qui les ont faits*). Nada obsta, pues, a que el contenido del contrato pueda consistir, en todo o en parte, en un bloque de código informático y a que los derechos y deberes de las partes puedan determinarse algorítmicamente. Es más, la Ley de Enjuiciamiento Civil española hace referencia en algunos de sus preceptos a la determinación del importe de una obligación pecuniaria mediante un procedimiento algebraico-algorítmico. De este modo, el artículo 219.2 establece que *la sentencia de condena establecerá el importe exacto de las cantidades respectivas, o fijará con claridad y precisión las bases para su liquidación, que deberá consistir en una simple operación aritmética que se efectuará en la ejecución*, mientras que el art. 572.2, en sede de ejecución, establece que

> *también podrá despacharse ejecución por el importe del saldo resultante de operaciones derivadas de contratos formalizados en escritura pública o en póliza intervenida por corredor de comercio colegiado, siempre que se haya pactado en el título que la cantidad exigible en caso de ejecución será la resultante de la liquidación efectuada por el acreedor en la forma convenida por las partes en el propio título ejecutivo*".

En mi opinión, tal y como se puede deducir de los preceptos citados en los párrafos anteriores, los contratos inteligentes pueden dar lugar al surgimiento de obligaciones civiles si así lo acuerdan las partes, en virtud del principio de la autonomía de la voluntad. Sin embargo, para que puedan dar inicio a un procedimiento de ejecución es necesario que su algoritmo aparezca consignado en un título ejecutivo. Para lograr esto último caben dos opciones:

-La primera de ellas es que, dentro de la práctica notarial, se abra la posibilidad de que las escrituras públicas contengan en su parte dispositiva el código de un contrato inteligente, así como la descripción de su contenido y funcionamiento en lenguaje ordinario. El notario ejercería un control de legalidad preventivo sobre el mismo.

-La segunda opción es promover una reforma de la Ley de Enjuiciamiento Civil que atribuya carácter ejecutivo a los contratos inteligentes. El derecho a la tutela judicial efectiva de los deudores quedaría protegido atribuyendo al juez la facultad de examinar la legalidad de los mismos al inicio de cada procedimiento ejecutivo, de un modo similar al que tiene lugar cuando hay cláusulas abusivas (art 552.2 LEC).

6.Consideraciones finales: la interdependencia entre el ordenamiento jurídico y el ordenamiento algorítmico

En el primer apartado de este artículo, hemos visto que lo que caracteriza a las tecnologías de la Cuarta Revolución Industrial es la interdependencia e interacción entre el mundo físico y las estructuras del ciberespacio. En esta (postrera) sección de conclusiones, expondremos que en el ámbito jurídico se da también un fenómeno similar: *Los estados de máquina de los protocolos distribuidos pueden influir en las situaciones jurídicas de los individuos y viceversa*. Para ilustrar este último fenómeno examinaremos brevemente el ataque *The DAO*, que tuvo lugar en la red *Ethereum* en el año 2017, y la posible trascendencia jurídica que habría tenido el mismo, de haber realizado los atacantes sus acciones en un país de Derecho Civil continental.

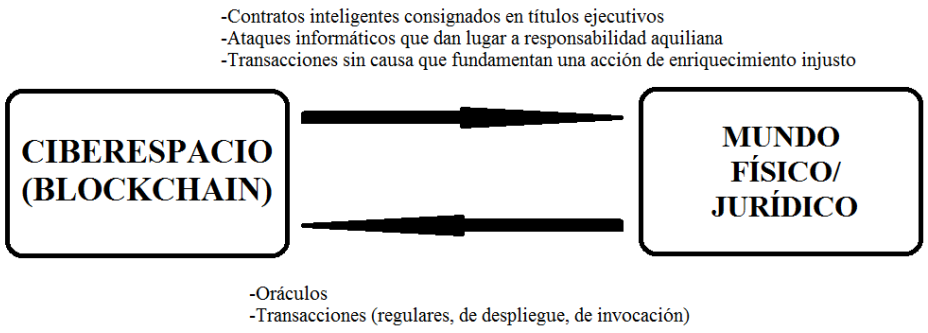
En el año 2016 se constituyó dentro de la red de Ethereum una suerte de sociedad mercantil criptográfica denominada *The DAO* (*Decentralized Autonomous Organization*), que captaba fondos por medio de ampliaciones de capital, con el ánimo de invertirlos en proyectos de capital-riesgo. El funcionamiento de *The DAO* era regulado por un conjunto de contratos inteligentes, que hacían las veces de estatutos societarios. Un error en el código, particularmente en aquella sección que regulaba los procedimientos de escisión, permitió que un atacante sustrajese 3.641.694 *ethers*, equivalentes en aquél entonces a unos 50 millones de dólares. El atacante publicó una carta en la que reafirmaba la legitimidad de su adquisición, pues la había consumado formalmente con arreglo al código informático regulador de *The DAO*, siendo un principio básico de los ordenamientos algorítmicos que *el código de los smarts contracts es la Ley.*[14] Dentro de las *blockchains,* las ideas no expresables algorítmicamente, como las de *justicia* o *equidad,* no juegan papel alguno. Lo que cuenta es que una adquisición se haya realizado formalmente de acuerdo con las normas del protocolo. Aunque finalmente la mayoría de los nodos de Ethereum acordaron crear un *hard fork* que retrotrajo el estado de máquina de la *blockchain* al momento inmediatamente anterior al ataque, en mi opinión, los perjudicados por ese ataque habrían tenido a su disposición dos remedios procesales diferentes, de acuerdo con el Derecho Civil continental, a saber, a) en primer lugar, podrían haber ejercitado una **acción aquiliana** (*de responsabilidad extracontractual*) contra los programadores cuya negligencia hubiera sido causa del ataque, pues en este supuesto se cumplían dos de los requisitos exigidos por la jurisprudencia continental para la estimación de una *acción de responsabilidad extracontractual*: (1)daño patrimonial (cuantificable) a los inversores y (2)existencia de negligencia en los programadores, cuyo diseño defectuoso del procedimiento de escisión de la sociedad posibilitó la aparición de *ataques de reentrada* (*reentrancy attacks*) que vaciaron buena parte de los fondos de la sociedad. Los programadores vendrían obligados a reparar el daño causado, y todos ellos devendrían responsables solidariamente de la indemnización que fijasen los tribunales de justicia. Aunque quizá la responsabilidad patrimonial de los mismos debería ser modulada en función de si su labor fue retribuida o no. b) En segundo lugar, los perjudicados podrían haber ejercitado una **acción de enriquecimiento injusto** contra el atacante. Originariamente proveniente del Derecho Romano, esta institución ha penetrado también en los ordenamientos jurídicos anglosajones. Las diferentes jurisprudencias nacionales, por lo general, exigen cinco requisitos para el ejercicio de esta acción: (1)enriquecimiento de una persona,

(2)empobrecimiento de otra, (3)conexión entre ambos, (4)ausencia de un negocio jurídico causal que justifique la transparencia patrimonial entre ambas personas y (5)ausencia de otro remedio procesal disponible para el perjudicado. Es evidente que el ataque a *The Dao* -que supuso la transferencia acausal e injustificada de 50 millones de dólares desde el patrimonio de los accionistas/inversores hacia el del atacante- es subsumible claramente dentro de los supuestos de enriquecimiento injusto.

Así pues, tal y como hemos expuesto en párrafos anteriores, múltiples fenómenos acaecidos en el ámbito del ciberespacio puede tener repercusiones en el ámbito del tráfico jurídico ordinario: contratos inteligentes consignados en títulos ejecutivos, ataques informáticos que dan lugar al surgimiento de responsabilidad extracontractual, transacciones erróneas que pueden servir de base para plantear una acción de enriquecimiento injusto... todos ellos son supuestos en los que anotaciones electrónicas de una *blockchain* pueden dar lugar al inicio de procedimientos judiciales.

En un sentido inverso, el flujo de información con relevancia jurídica también puede ir desde el mundo físico hacia las *blockchains*, fundamentalmente a través de dos vías diferentes: oráculos y transacciones. Los **oráculos** son canales de entrada de datos (*data feed channels*) que dan fe de que en el mundo físico se han producido determinados hechos relevantes para un contrato inteligente y que pueden ser de lo más heterogéneo (cotización de acciones y de divisas, nacimiento de una persona determinada, etc.). Las **transacciones** son declaraciones de voluntad, firmadas criptográficamente por un actor externo, y capaces de inducir un cambio en el estado de máquina de un protocolo distribuido, en los términos señalados en los artículos anteriores.

Fig. 6: Interacción entre el mundo jurídico y el ciberespacio (blockchain)
Fuente: Elaboración propia



Con todo, la integración entre el mundo jurídico y el ciberespacio es aún incipiente. Como podrá comprobar el lector, la mayoría de las instituciones vigentes en las *blockchains* y analizadas en los párrafos anteriores encuentran únicamente aplicación en determinadas áreas del Derecho Privado como el Derecho Mercantil, Civil y Financiero. El análisis del uso de algoritmos en áreas como el Derecho Penal[15] o el Constitucional va más allá del alcance de este artículo, no sólo por razones de espacio, sino porque, hasta ahora, las principales aplicaciones de los protocolos distribuidos han tenido lugar fundamentalmente en el ámbito de los derechos

patrimoniales. Por otra parte, la inmensa mayoría de los contratos inteligentes consignados en las diversas *blockchains* tienen una estructura relativamente simple y no alcanzan ni de lejos la versatilidad de normas como la Ley o el Reglamento Hipotecarios o la Ley de Sociedades de Capital, que regulan el tráfico inmobiliario y mercantil en España respectivamente. Por último, la legislación europea es aún silente sobre los efectos civiles que pueden tener los contratos inteligentes, sobre qué papel se concedería a los fedatarios públicos en su elaboración y control y en qué tipo de responsabilidades (extra)contractuales podrían incurrir los programadores que realizasen defectuosamente su labor.

Sólo el tiempo nos dirá hasta dónde puede llegar la *algoritmización del Derecho*,y cuáles son los límites intrínsecos de la misma.

Notas

[1]http://www.wolfgang-wahlster.de/wordpress/wp-content/uploads/Industrie_4_0_Mit_dem_Internet_der_Dinge_auf_dem_Weg_zur_vierten_industriellen_Revolution_2.pdf (consultado el 23 de octubre de 2019).

[2] El *álgebra de Boole* es aquélla que opera exclusivamente con unos y ceros.

[3] Algunos físicos consideran que en última instancia el Universo es digital. El espacio y el tiempo estarían compartimentados en *cuantos* y sólo serían continuos a niveles macroscópicos. El propio Konrad Zuse publicó un artículo, *Rechnender Raum* (El Espacio Computador), en el que defendía esta postura.

[4] Una base de datos distribuida es aquella cuyo contenido está replicado en varios servidores.

[5] Latencia es el retraso en la transmisión de información dentro de una red.

[6] Computadora universal es aquélla que, si dispusiera de memoria ilimitada, podría calcular los valores de cualquier función computable.

[7] Un bucle es un conjunto de instrucciones que un programa ejecuta una y otra vez. Si el bucle se va a ejecutar un número determinado de veces, se lo denomina bucle *for*. Si se va a ejecutar un número indeterminado de veces, se lo denomina bucle *while*. En ocasiones, los bucles *while* devienen infinitos, de tal modo que el programa no es capaz de salir de los mismos y el ordenador se queda colgado.

[8] Estrictamente hablando, la entrada debe ser proporcionada en un medio de almacenamiento reescribible, como un disco duro, o una memoria USB.

[8] Operaciones de coma flotante por segundo.

[9] https://arxiv.org/pdf/1810.02466.pdf (consultado el 23 de octubre de 2019).

[10]https://www.itnews.com.au/news/nsa-can-break-some-encryption-new-snowden-leak-355942 (consultado el 23 de octubre de 2019).

[11]https://elbinario.net/2015/01/29/qwerty-el-keylogger-desarrollado-por-la-nsa-segun-kaspersky-lab/ (consultado el 23 de octubre de 2019).

[12] https://bitcoin.org/en/alert/2013-08-11-android (consultado el 23 de octubre de 2019).

[13]https://nakedsecurity.sophos.com/2013/08/12/android-random-number-flaw-implicated-in-bitcoin-thefts/ (consultado el 23 de octubre de 2019).

[14] https://pastebin.com/CcGUBgDG (consultado el 23 de octubre de 2019).

[15] https://elpais.com/elpais/2018/01/17/ciencia/1516194073_122982.html (consultado el 23 de octubre de 2019).

Bibliografía

ANTONOPOULOS, A. M. (2014). *Mastering bitcoin: unlocking digital cryptocurrencies*. O'Reilly Media.

BARAN, P. (1962). *On distributed communication networks*. Santa Monica, California: The RAND Corporation.

BUTERIN, V. *Ethereum white paper. A next generation smart contract & decentralized application platform*. Disponible en: http://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf (consultado el 18 de Octubre de 2019).

DIFFIE, W. y HELLMAN, M. E. "New directions in cryptography". *IEEE Transactions of Information Theory*. Vol IT-22, n.º 6 (November 1976).

GOMÁ SALCEDO, E. (2005). *Instituciones de Derecho Civil, Común y Foral: Tomo I*. Barcelona: Editorial Bosch.

GONZÁLEZ MENESES, M. (2017). *Entender blockchain. Una introducción a la tecnología de registro distribuido*. Navarra: Thomson-Reuters Aranzadi.

KELSEN, H. (2000). *Teoría pura del derecho*. México: Porrúa.

LAMPORT, L., SHOSTAK, R. & PEASE, M. *The Byzantine generals problem*. Disponible en: https://people.eecs.berkeley.edu/~luca/cs174/byzantine.pdf (consultado el 26 de Agosto 2019)

LESSIG, L. (2001). *El código y otras leyes del ciberespacio*. Madrid: Grupo Santillana de Ediciones.

NAKAMOTO, S. *Bitcoin: A peer-topPeere electronic cash system*. Disponible en: https://bitcoin.org/bitcoin.pdf (consultado el 26 de Agosto 2019)

RIVEST, R., SHAMIR, A. & ADLEMAN, L. *A method for obtaining digital signatures and public-key cryptosystems*. Disponible en: https://people.csail.mit.edu/rivest/Rsapaper.pdf (consultado el 23 de octubre 2019)

RODRÍGUEZ ABRIL, R. (2019). "Sobre la legitimación criptográfica de firmas en los contratos". *Derecom.La Revista Internacional del Derecho de la Comunicación y de las Nuevas Tecnologías*, 27, 112-139. http://www.derecom.com/derecom (consultado el 23 de octubre 2019)

ROJAS, R. "Conditional branching is not necessary for universal computation in Von Neumann computers". J*ournal of Universal Computer Science,* vol. 2, no. 11 (1996), 756-768. Springer. Pub. Co..

ROMANO, S. (2013). *El ordenamiento jurídico.* Madrid: Centro de Estudios Políticos y Constitucionales.

SHANNON, C. (1937). *A symbolic analysis of relay and switching circuits.* Disponible en: https://www.cs.virginia.edu/~evans/greatworks/shannon38.pdf (consultado el 23 de Octubre de 2019)

SZABO, N. (1997). "Formalizing and securing relationships on public networks". Disponible en: https://nakamotoinstitute.org/formalizing-securing-relationships/ (consultado el 23 de octubre de 2019)

VAN SABERHAGEN, N. *CryptoNote v 2.0.*
Disponible en: https://cryptonote.org/whitepaper.pdf (consultado el 18 de octubre 2019)

VARGAS, C. (2013). "La jerarquía de Chomsky: una evolución esperable en teoría de la computabilidad". *Aporía, Revista Internacional de Investigaciones Filosóficas,* nº5 (2013), p. 63-68.

WERBACH, K., & CORNELL, N. (2017). "Contracts ex machina". En *Duke Law Journal,* 67, p. 320 y nota 26.
Disponible en:
https://scholarship.law.duke.edu/cgi/viewcontent.cgi?article=3913&context=dlj (consultado el 18 de octubre 2019)

WOOD, G. "Ethereum: A secure decentralised generalised transaction ledger". Disponible en: https://gavwood.com/paper.pdf (consultado el 18 de octubre 2019)