Redacción de Contratos Legales Inteligentes en Colombia

Draft Smart Legal Contracts in Colombia

Autores

NICOLÁS FELIPE PARDO VERA

NATALIA SUÁREZ GONZÁLEZ

Trabajo presentado como requisito para optar por el título de Magister en Derecho Corporativo

Director, Tutor

Luisa Jiménez Mahecha

Universidad del Rosario

Facultad de Jurisprudencia

Maestría en Derecho Corporativo

Bogotá, Colombia

Año 2025

Redacción de Contratos Legales Inteligentes en Colombia

Draft Smart Legal Contracts in Colombia

Natalia Suarez González

Nicolas Pardo Vera

Resumen

Este escrito analiza la definición, características y creación de los Contratos *Legales* Inteligentes, con el propósito de responder a la pregunta: ¿cómo puede expresarse la lógica de un contrato comercial mediante un lenguaje de programación? A partir de un enfoque práctico, el texto da respuesta a esta pregunta explicando cómo se pueden expresar los Contratos Legales Inteligentes. El artículo tiene la siguiente estructura: (i) delimita el concepto de Contratos *Legales*; (ii) categoriza los distintos tipos; (iii) hace un breve análisis sobre los requisitos legales y técnicos requeridos para utilizarlos; (iv) profundiza sobre los requisitos prácticos para redactar Contratos Legales Inteligentes; (v) expone una metodología propuesta por la doctrina para su desarrollo; y, finalmente, (vi) presenta las conclusiones.

Summary

This paper analyzes the definition, characteristics, and use of Smart *Legal* Contracts in Colombia. It seeks to answer the question of: how can the logic of a commercial contract be expressed through a programming language? Taking a practical approach, the text addresses these questions by explaining how the Smart Legal Contracts can be implemented. The article is structured as follows: (i) it defines the concept of Smart Legal Contracts; (ii) it categorizes the different types of Smart Legal Contracts; (iii) it provides a brief analysis of the legal and technical requirements for using Smart Legal Contracts; (iv) it digs into the practical requirements for drafting Smart Legal Contracts; (v) it presents a methodology proposed by the doctrine; and finally, (vi) it presents the conclusions.

Palabras Clave:

Contrato, Smart Contract, Smart Legal Contract, Derecho de las Tecnologías, Derecho Contractual.

Key Words:

Contract, Smart Contract, Smart Legal Contract, Tech Law, Contract Law.

Introducción

En el marco de la Cuarta Revolución Industrial los Contratos –Legales– Inteligentes han aparecido como una alternativa a la contratación comercial tradicional. Aun cuando mucho se ha dicho sobre este tema[1][2][3][4], son pocos los análisis sobre cómo realmente se podría aplicar esta figura al ámbito contractual desde la práctica jurídica y técnica. Por eso, este artículo tiene como objetivo responder de qué manera es posible redactar la lógica de un contrato comercial en un lenguaje de programación, de forma tal que se puedan desarrollar acuerdos comerciales que puedan ser entendidos por una máquina, para su ejecución posterior.

Para responder a esta cuestión, en primer lugar, (i) se delimitará el concepto de Contrato Inteligente; luego, (ii) se explicarán los tipos de Contratos Inteligentes, (iii) se analizará brevemente qué requisitos son necesarios para su validez y utilización –tanto desde el punto de vista legal colombiano, como desde un punto de vista técnico–; posteriormente, (iv) se describirá el funcionamiento de la propuesta "Accord Project", como una alternativa para redactar lógica contractual en términos de código computacional; y finalmente (v) se abrirá la discusión a una posible aplicación de una propuesta metodológica para la redacción de Contratos Inteligentes.

1. ¿Qué es un Contrato Inteligente -*Smart Contract*[5]- y un Contrato Legal Inteligente -*Smart Legal Contract*- ?

Un Contra Inteligente es una serie de instrucciones y protocolos que se le da a un computador para que, ante ciertos estímulos, condiciones o actos, ejecute determinado tipo de acciones alternativas a los procedimientos tradicionales[6]. En ese sentido, independientemente de la

---

[1] Erick Rincón y Valeria Molano, "Contratos Inteligentes y automatización como desarrollos aplicados del LegalTech en Colombia" *Revista Direito GV*, (2022): 239-260. https://doi.org/10.1590/2317-6172202211.

[2] Julián Leonardo Hernández Díaz, "Nulidad por objeto ilícito en el smart-contract: el rol del juez en un contrato irreversible" (Trabajo de grado pregrado en Derecho Civil, Bogotá, Universidad Externado de Colombia, 2020), https://doi.org/10.57998/bdigital.handle.001.4262.

[3] Daniel Luque Restrepo, "Análisis de las obligaciones condicionales en la implementación de Smart Contracts en el ordenamiento jurídico colombiano" (Trabajo de grado pregrado en Derecho, Medellín, Universidad Pontificia Bolivariana, 2020).

[4] Viviana Paola Díaz Baquero, "Regulación de los Contratos Inteligentes en colombia" (Trabajo de grado especialista en Derecho Comercial, Bogotá, Pontificia Universidad Javeriana, 2019), https://repository.javeriana.edu.co/bitstream/handle/10554/46186/Trabajo%20de%20grado.pdf.

[5] Para efectos de este escrito, el término "Contrato Inteligente" incluye y es equivalente al de "Smart Contracts" por su traducción literal. De la misma forma, "Contrato Legal Inteligente", lo es de "Smart Legal Contracts". Los términos se han unificado en el idioma español para evitar incongruencias y confusiones.

[6] Robin Matzke, "Smart Contracts statt Zwangsvollstreckung? Zu den Chancen und Risiken

plataforma de despliegue, un Contrato Inteligente define las reglas de un sistema. Así, si bien es cierto que un Contrato Inteligente es el nombre especial que se le ha dado a la funcionalidad de ejecutar operaciones autónomas sin intervención humana[7] sobre una blockchain o cadena de bloques[8]; lo cierto es que un Contrato Inteligente se puede presentar en otro tipo de aplicaciones como sería un sitio web de compras –o el típico ejemplo de la máquina expendedora de dulces[9]–.

Partiendo de la idea de que un Contrato Inteligente es un software que designa las reglas de funcionamiento de un sistema, en estricto sentido, un contrato inteligente no es necesariamente un contrato en sentido jurídico[10][11]. Sin embargo, sí es posible representar las reglas de funcionamiento de un contrato a través de una aplicación o un Contrato Inteligente[12/13].

Para retomar el ejemplo del sitio web de compras, una persona que accede para adquirir un par de zapatos a una tienda online, en esencia, celebra un contrato de compra-venta con un establecimiento de comercio. En este proceso, la intervención humana es nula o previa. No hay una persona atendiendo cada caso particular. El sistema tiene predefinidos los pasos para la celebración de un contrato de compraventa, independientemente de quien sea el potencial cliente.

---

der digitalisierten privaten Rechtsdurchsetzung", en *Smart Contracts* (Tubinga: Mohr Siebeck, 2019), 99-116, https://www.jstor.org/stable/j.ctvn96h9r.10.

[7] Jorge Feliu Rey, "Smart Contract: Concepto, ecosistema y principales cuestiones de Derecho privado", *La Ley Mercantil,* No. 47 (2018): 1-27. https://ssrn.com/abstract=3247775.

[8] Viviana Paola Díaz Baquero, "Regulación de los Contratos Inteligentes en colombia" (Trabajo de grado especialista en Derecho Comercial, Bogotá, Pontificia Universidad Javeriana, 2019), https://repository.javeriana.edu.co/bitstream/handle/10554/46186/Trabajo%20de%20grado.pdf.

[9] Luis Felipe García Rubio, "Contratos Inteligentes en blockchain. Una propuesta de lege data para el derecho privado colombiano en materia contractual", *Anuario de Derecho Privado 01*, n.o 2 (2020): 9–45. dx.doi.org/10.15425/2017.350.

[10] Afirmamos que no *necesariamente* es un contrato "en sentido jurídico" porque esto implicaría que en todos los sistemas informáticos existe un acuerdo de voluntades, una oferta y una aceptación, que sus condiciones son exigibles, etc. Esta no es la regla general para un software.

[11] Ihor Fetsyak, "Contratos Inteligentes: análisis jurídico desde el marco legal español", *Revista Electrónica de Derecho de la Universidad de La Rioja (REDUR)*, No. 18 (2020): 197-236. https://doi.org/10.18172/redur.4898.

[12] Vinay K. Chaudhri, "Computable Contracts in the Financial Services Industry", *arXiv, Cornell University,* (2022): 1-10. https://doi.org/10.48550/arXiv.2208.04685.

[13] José Ricardo Redondo-Luque, "Análisis jurídico de los Smart Contracts o Contratos Inteligentes", *Revista Acta Académica,* n.o 70 (2022): 169-186. http://revista.uaca.ac.cr/index.php/actas/article/view/1343.

En este caso, se trata de un contrato inteligente, desplegado sobre una red centralizada –o descentralizada–, que genera –y está diseñado para generar efectos jurídicos[14]. Así las cosas, consideramos que en los casos en que una aplicación o contrato inteligente definen o representan el funcionamiento de un contrato –en sentido jurídico–, se trata de un Contrato *Legal* Inteligente.

Así pues, consideramos que es útil hacer una distinción entre Contratos Inteligentes y Contratos Legales Inteligentes. Los primeros hacen referencia a una aplicación que define las reglas de un sistema informático y que no genera efectos jurídicos –o cuya intención principal no es generar efectos jurídicos–. Por ejemplo, la aplicación de la calculadora en el celular.

Varios autores coinciden al afirmar que en entornos empresariales los Contratos Inteligentes -legales o no- facilitan los procesos colaborativos al mantener un registro permanente e inalterable de las transacciones, lo cual lo hace confiable[15] [16]. Sin embargo, el desarrollo y la integración de estos es un proceso propenso a errores, lo que subraya la necesidad de un diseño cuidadoso y la aplicación de las mejores prácticas[17].

Por su parte, los Contratos Legales Inteligentes son softwares que definen las reglas de un sistema, que representa –y tiene la intención de representar– la celebración, funcionamiento y/o ejecución de un contrato, en sentido jurídico[18]. Al igual que es posible escribir las reglas de un contrato en un papel y esperar que las partes las ejecuten, es definir las reglas de un contrato en un protocolo de transacción computarizado y esperar que el sistema ejecute los términos del contrato[19]. Se puede afirmar que se trata simplemente de una forma de representación o solemnidad acordada entre las partes –de la misma manera que tradicionalmente lo ha sido el papel–.

---

[14] Anthony Gustavo Chávez Mendoza, "El arbitraje como método de solución de controversias derivados del Smart Contract" (Trabajo de grado pregrado en Derecho, Guayaquil, Universidad Católica de Santiago de Guayaquil, 2022), http://repositorio.ucsg.edu.ec/handle/3317/18600.

[15] Nicolás Sánchez-Gómez, Jesús Torres-Valderrama, Manuel Mejías Risoto y Alejandra Garrido, "Blockchain smart contract meta-modeling*", Journal of Web Engineering 20,* n.o 7 (2021): 1-18. 10.13052/jwe1540-9589.2073.

[16] Reza Moradinejad, "Le contrat intelligent, nouveau vecteur de confiance dans les relations contractuelles: réalité ou rêve?*", Les Cahiers de droit 60,* n.o 3 (2019): 623-651. https://doi.org/10.7202/1064651ar.

[17] Nicolás Sánchez-Gómez, Jesús Torres-Valderrama, Manuel Mejías Risoto y Alejandra Garrido, "Blockchain smart contract meta-modeling".

[18] Gustavo Marchi de Souza Mello, "Smart Legal Contracts: Einsatzmöglichkeiten im recht und rechtliche Rahmenbedingungen"*, Revista da Faculdade de Direito, Universidade de São Paulo 115,* (2020): 751-775. https://doi.org/10.11606/issn.2318-8235.v115p751-775.

[19] Thomas Riehm, "Smart Contracts und verbotene Eigenmacht", en *Smart Contracts* (Tubinga: Mohr Siebeck GmbH and Co. KG, 2019), 85-98, https://www.jstor.org/stable/j.ctvn96h9r.9.

Dentro de la doctrina, existen múltiples definiciones de lo que puede ser considerado un Contrato Legal Inteligente[20]. Para efectos de este trabajo, tomamos la definición –que consideramos amplia– propuesta por la comisión legal del Reino Unido[21], que define a los Contrato Legal Inteligente como "Un acuerdo legal obligatorio en el cual algunos o todos sus términos son definidos y/o ejecutados automáticamente por un programa de computador"[22].

Como se puede observar, y este es un criterio reiterado en la doctrina, la definición parte de dos ideas principales: 1) que los términos sean definidos en un programa de computador; y/o 2) que los términos sean ejecutados por un computador. Teniendo estos criterios como base, la comisión[23] plantea una caracterización de los Contratos Legales Inteligentes, dependiendo del grado de incidencia que tenga la redacción de código y la ejecución del software, así "Existen esencialmente tres formas en las que los Contratos Legales Inteligentes se pueden presentar dependiendo del papel que desempeñe el código. Estas son:

- Lenguaje natural con ejecución automatizada
- Contratos híbridos
- Contratos de sólo código"[24] [25].

Esta definición ha sido tomada como punto de partida por su amplitud. Muchos autores limitan el concepto de Contrato Legal Inteligente a lo que la comisión denomina como contratos híbridos (hybrid contracts) o contratos de sólo código (solely code contract)[26]. No obstante, consideramos que una definición amplia permite incluir dentro del concepto muchas aplicaciones que, en principio, no eran consideradas como tal. Esto permite, a su vez, desmitificar la figura del Contrato Legal Inteligente y entender que es una realidad ya presente en nuestro contexto. A continuación, se analizan cada uno de estos tipos de Contratos Legales Inteligentes.

2. Categorización: Tipos de Contratos Legales Inteligentes

[20] Julián Leonardo Hernández Díaz, "Nulidad por objeto ilícito en el smart-contract: el rol del juez en un contrato irreversible".

[21] Law Commission. "Smart Contracts". Consultado el 22 octubre de 2024, https://lawcom.gov.uk/project/smart-contracts/.

[22] Law Commission, *Smart Legal Contracts. Advice to Goverment* (United Kingdom, Law Commission, 2021), 10 (Traducción propia). https://cloud-platform-e218f50a4812967ba1215eaecede923f.s3.amazonaws.com/uploads/sites/30/2021/11/Smart-legal-contracts-accessible.pdf

[23] Law Commission, *Smart Legal Contracts. Advice to Goverment*.

[24] Law Commission, *Smart Legal Contracts. Advice to Goverment*: 10 (Traducción propia).

[25] Una definición similar se puede encontrar en Fetsyak, Ihor, "Contratos Inteligentes: análisis jurídico desde el marco legal español".

[26] Law Commission, *Smart Legal Contracts. Advice to Goverment*.

Partiendo de la definición propuesta, pueden existir tres tipos principales: contrato en lenguaje natural con desempeño automático; contratos híbridos, y contratos de solo código[27].

2.1. Contratos en lenguaje natural con desempeño automático

Los Contratos Legales Inteligentes, o los contratos en lenguaje natural con desempeño automático, son aquellos en los que las obligaciones y términos contractuales están redactados y/o acordados por las partes en lenguaje natural[28]. Sin embargo, su ejecución se da a través de código o un software. Este es el caso que más se presenta en la actualidad.

Piénsese, por ejemplo, en el pago de unos servicios públicos. En principio, la regulación de los deberes y obligaciones tanto del prestador del servicio como del beneficiario están señalados en un texto contractual y en la Ley. Sin embargo, la prestación del servicio –los medidores, etc.– está automatizada por un software. De la misma manera, en muchos casos, el pago por la prestación del servicio se termina descontando de forma automática o, cuanto menos, por medio de una pasarela de pagos virtuales.

Estas aplicaciones no son nuevas. Todos las hemos visto y usado. Sin embargo, este tipo de Contratos Legales Inteligentes pueden tener un problema que, como se verá, no es propio –o al menos no tan usual– de los otros tipos de Contratos Legales Inteligentes. Al existir una diferencia entre la *redacción* o *celebración* del contrato y su *ejecución* es perfectamente posible que la aplicación no refleje fidedignamente los términos contractuales pactados[29].

En el proceso de elaboración de una aplicación, por ejemplo, bancaria que permite realizar transacciones, de acuerdo con lo previamente pactado en una oficina, en un papel denominado "contrato de depósito en cuenta de ahorros". En este caso, seguramente la aplicación fue realizada por el equipo de TI del banco –o una empresa externa–, mientras que el texto contractual fue redactado por el equipo jurídico del banco –o una firma externa–. De entrada, esto supone dos tipos de actuaciones, lenguajes e interpretaciones diferentes que pueden desencadenar en soluciones o *reglas* distintas.

2.2. Contratos Legales Inteligentes híbridos

En segundo lugar, se encuentran los Contratos Legales Inteligentes híbridos[30]. En este caso, se trata de una obligación legal que se encuentra definida en lenguaje natural y en lenguaje de computación. Además, su ejecución se delega parcial o totalmente a un programa de computador.

---

[27] Law Commission, *Smart Legal Contracts. Advice to Goverment*: 23 (Traducción propia).

[28] Por "Lenguaje natural" entendemos aquellas formas de comunicación usadas por los seres humanos usan para interactuar (español, inglés u otros idiomas).

[29] Markus Kaulartz, "Smart Contract Dispute Resolution", en *Smart Contracts* (Tubinga: Mohr Siebeck, 2019), 73-84, http://www.jstor.org/stable/j.ctvn96h9r.8.

[30] Law Commission, *Smart Legal Contracts. Advice to Goverment*.

Si bien para la Comisión del Derecho[3132] esta definición de Contratos Legales Inteligentes híbridos es únicamente una forma en la que se pueden presentar los Contratos Legales Inteligentes, para otros doctrinantes, esta es, en un sentido más restrictivo, no corresponden a Contratos Legales Inteligentes. En este sector de la doctrina, un contrato inteligente debe tener no sólo una ejecución parcial o total a través de un software, sino que su redacción o acuerdo también debe estar parcial o totalmente reflejado en el código. En ese sentido, un contrato celebrado de forma tradicional, aun cuando su ejecución se de a través de un software, no entraría dentro de la categoría de Contrato Legal Inteligente.

Por ejemplo, Veiga Copo analiza el impacto de los contratos de seguro tradicionales al transformarse en Contratos Inteligentes, proceso en el que se requiere tanto la adaptación a conceptos y mentalidades informáticas como prestar atención al papel de la inteligencia artificial y la big data en el futuro de este tipo de contratos[33] [34].

2.3. Contratos Legales Inteligentes de solo código

En tercer lugar, se encuentran los Contratos Legales Inteligentes cuyas obligaciones están redactadas puramente en código o lenguaje de programación[35]. En este caso existe una correlación entre el código que refleja la obligación contractual y su ejecución: es posible determinar en el código del software la existencia de una obligación legal y además es posible observar su ejecución por el mismo programa de computador. En términos prácticos, este tipo de Contratos Legales Inteligentes permiten una mayor identidad entre las obligaciones pactadas y su ejecución, de tal forma que limita las posibles diferencias que pudieran existir entre código y contrato, pues terminan siendo idénticos, al menos, en principio.

Habiendo definido qué se entiende por Contrato Legal Inteligente a partir de diferentes tipos, procederemos a analizar qué aspectos serían necesarios para su implementación en el contexto colombiano. En este punto, se resalta la importancia de la distinción propuesta pues, si bien los conceptos a los que se hará referencia son aplicables a cualquiera de las tres categorías reseñadas, su aplicación está más orientada hacia los Contrato Legal Inteligente que cuentan con una redacción obligacional en código, estos son, los híbridos y los de solo código.

---

[31] El nombre de la comisión proviene de nuestra traducción propia al nombre en inglés de la entidad productora del texto: "The Law Commission". Esta *comisión* fue creada por el Acto 1965 en el Reino Unido con el propósito de reformar el sistema legal.

[32] Law Commission, *Smart Legal Contracts. Advice to Goverment*.

[33] Abel B. Veiga Copo, "«Smart contract» y contrato de seguro. Una ecuación asimétrica y no sólo algorítmica", *Revista de Derecho del Sistema Financiero*, (2020): 119-184. http://hdl.handle.net/11531/53117.

[34] Liced Morales Higuita y Natalia Isabel Pulgarín Agudelo, "Smart Contracts en materia de arrendamiento de vivienda urbana a partir de los artículos 1, 3 y 22 de la Ley 820 de 2003" (Trabajo de grado pregrado en Derecho, Caldas, Unilasallista Corporación Universitaria, 2022), https://repository.unilasallista.edu.co/items/de54f029-5856-429d-96b5-9ad38dbf4d31.

[35] Law Commission, *Smart Legal Contracts. Advice to Goverment*.

3. Requisitos para la representación de contratos legales en aplicaciones

Para efectos de este escrito nos centraremos en dos principales requisitos para la celebración de Contratos Legales Inteligentes: los requisitos legales y los requisitos técnicos.

3.1. Requisitos Legales

Dentro de los requisitos legales, entendemos qué se necesita, desde el punto de vista legal, para la celebración válida de este tipo de acuerdos. Por regla general, el principio de la autonomía de la voluntad señala que existe un contrato válidamente celebrado cuando se llega a un acuerdo entre las partes[36]. En la práctica, lo usual es que las partes negocien los términos del contrato, luego se escriban en papel y, por último, las partes lo firmen. Si bien podría no ser necesaria esta firma –el simple acuerdo de las partes sobre las reglas de la relación sería suficiente para dar nacimiento al contrato–, lo cierto es que las partes la usan como mecanismo para expresar su voluntad y para marcar un derrotero en el momento de la celebración del acuerdo.

Algo similar ocurre cuando las partes deciden suscribir un contrato por medios electrónicos o digitales. En virtud de la Ley 527 de 1999, en nuestro sistema jurídico, existe el principio de equivalencia funcional y la validez de la firma electrónica. Con base en ellos, podemos afirmar que siempre que exista una prueba del consentimiento que pueda demostrar la autenticidad y conservar la integridad del documento, podemos hablar de un consentimiento válidamente dado. En este sentido, ya sea a través de una validación por medios digitales del consentimiento, o por la aquiescencia implícita que da la persona al interactuar con el Contrato Inteligente, lo cierto es que, siempre que exista dicho consentimiento existirá un contrato vinculante entre las partes, independientemente del medio.

Por lo tanto, los requisitos legales para la celebración de los Contratos Legales Inteligentes son *iguales* a los de cualquier tipo de contrato[37] [38]: sus elementos de validez –consentimiento, capacidad, causa y objeto lícito–; sus elementos de existencia –los esenciales de cada tipo contractual–; y sus elementos de oponibilidad. Ahora bien, consideramos que no es del resorte de este escrito analizar cada uno de los requisitos legales de celebración de los

[36] Presidente de la República de Colombia. Decreto legislativo N° 410 de 1971: Por el cual se expide el Código de Comercio, Art. 845-863. (el 27 de marzo de 1971).
[37] Ihor Fetsyak, "Contratos Inteligentes: análisis jurídico desde el marco legal español".
[38] Si bien existen legislaciones que han definido los Contratos Inteligentes en sus codificaciones, consideramos que esto no es necesario, pues la normativa actual permite responder con suficiencia a las necesidades de este tipo de contratos. Promulgamos por la aplicación de los principios generales del derecho, antes que la proliferación de normas que termine complicando o impidiendo el uso de estos mecanismos.

contratos, cuya doctrina y alcance ha sido desarrollado por varios juristas[39][40][41][42]. En Colombia es posible y válido la celebración de un Contrato Legal Inteligente siempre que cumpla con las mismas condiciones legales que debería cumplir para su celebración válida[43] como si se tratara de un contrato tradicional o normal.

3.2. Requisitos técnicos

Dentro de los requisitos técnicos entendemos qué se necesita para la redacción, celebración y ejecución de un Contrato Legal Inteligente. En este punto, se pueden plantear tantos requisitos como sea posible imaginar: computadoras –más o menos rápidas-; sistemas operativos –más o menos avanzados-; aparatos externos que ayuden al funcionamiento; ingenieros –tantos como recursos humanos puedan manejar–; múltiples lenguajes de programación –con sus respectivas librerías y desarrollos propios; y así sucesivamente.

Lo cierto es que la complejidad en el desarrollo y la operación de un programa informático puede ir desde lo más simple hasta lo más complejo, y aun así tener campo para continuar con un escalamiento masivo. Además, el creciente avance y creación de nuevas tecnologías y desarrollos hacen que no sea posible determinar una lista taxativa y única de requisitos técnicos para desarrollar Contratos Legales Inteligentes. En este sentido, dentro del alcance del presente escrito no se encuentra determinar una única, segura y perfecta forma de crear Contratos Legales Inteligentes. Por el contrario, partiremos de la base de que existen infinidades de posibilidades para crear Contratos Legales Inteligentes, como la inteligencia humana puedan crear e imaginar.

Un Contrato Legal Inteligente puede ser desarrollado en una aplicación web centralizada que funcione con una base de JavaScript y que se conecte a una Base de Datos para asegurar la veracidad de las transacciones, tal como hoy en día funcionan la mayoría de las plataformas

---

[39] Erick Rincón y Valeria Molano, "Contratos Inteligentes y automatización como desarrollos aplicados del LegalTech en Colombia" *Revista Direito GV*, (2022): 239-260. https://doi.org/10.1590/2317-6172202211.

[40] Daniel Luque Restrepo, "Análisis de las obligaciones condicionales en la implementación de Smart Contracts en el ordenamiento jurídico colombiano" (Trabajo de grado pregrado en Derecho, Medellín, Universidad Pontificia Bolivariana, 2020).

[41] Julián Leonardo Hernández Díaz, "Nulidad por objeto ilícito en el smart-contract: el rol del juez en un contrato irreversible" (Trabajo de grado pregrado en Derecho Civil, Bogotá, Universidad Externado de Colombia, 2020), https://doi.org/10.57998/bdigital.handle.001.4262.

[42] Viviana Paola Díaz Baquero, "Regulación de los Contratos Inteligentes en colombia" (Trabajo de grado especialista en Derecho Comercial, Bogotá, Pontificia Universidad Javeriana, 2019), https://repository.javeriana.edu.co/bitstream/handle/10554/46186/Trabajo%20de%20grado.pdf.

[43] Daniel Luque Restrepo, "Análisis de las obligaciones condicionales en la implementación de Smart Contracts en el ordenamiento jurídico colombiano" (Trabajo de grado pregrado en Derecho, Medellín, Universidad Pontificia Bolivariana, 2020).

bancarias por Internet[44] o en el sector financiero y de seguros[45]. Pero también puede existir un Contrato Legal Inteligente que se conecte a una red privada y, utilizando una *IoT*, se conecte a un medidor de un barco o nevera a kilómetros de distancia, y reaccione ante los cambios de temperatura de acuerdo con la lógica contractual esperada. O se puede, simplemente, ejecutar tres líneas de código sobre una red descentralizada, y que esta transfiera recursos a otra *billetera*[46] –en ejecución de un mandato, donación, compraventa, etc.–.

Por tanto, para limitar el alcance del escrito a nuestro objetivo principal nos centraremos en analizar qué es necesario para redactar un Contrato Legal Inteligente. Es decir, qué es necesario para plasmar la lógica jurídica en términos que puedan ser entendidos y ejecutados por una máquina.

Al inicio de este escrito planteamos que un Contrato Legal Inteligente tenía dos vertientes: su *redacción* en forma que pueda ser entendida por humanos y máquinas; y una *ejecución* parcial o total de su contenido por una máquina. Para puntualizar, nuestro análisis sobre cómo *redactar* Contratos Legales Inteligentes se centrará a la primera parte, esto es, su redacción. El segundo aspecto –su *ejecución*– no será abordado por sus múltiples opciones. En la medida que podamos distinguir entre el momento de *redacción* del contrato y el momento de su *ejecución*, podremos seguir distinguiendo entre su código fuente y sus "elementos de *ejecución*", y, a su vez, se podrá hacer la distinción entre lógica jurídica expresada en términos computables y elementos técnicos que permitan el funcionamiento de dicha lógica.

El centro de la discusión de este artículo girará en torno a cómo utilizar un lenguaje de programación entendible por humanos y máquinas, para plasmar el contenido de un contrato y sus obligaciones jurídicas. Y, por tanto, el único requisito para la redacción de un Contrato Legal Inteligente, al menos dentro del alcance de nuestro texto, es un lenguaje de programación.[47] [48].

4. ¿Qué lenguaje de programación utilizar?

Basándonos en la premisa de que la redacción de Contratos Legales Inteligentes requiere únicamente un lenguaje de programación, surge de manera natural la pregunta: ¿qué lenguaje de programación es el más adecuado para este propósito? Y, al igual que en el ejercicio de la

---

[44] Vinay K. Chaudhri, "Computable Contracts in the Financial Services Industry".

[45] Antonio Legerén-Molina, "Los Contratos Inteligentes en España (La disciplina de los Smart Contracts)", *Revista de Derecho Civil 5*, n.o 2 (2018): 193-241. https://www.nreg.es/ojs/index.php/RDC/article/download/320/267.

[46] Billetera o Wallet es una aplicación para transigir, almacenar y, en general, gestionar criptomonedas y otros activos digitales.

[47] Christopher D. Clack, "Languages for Smart and Computable Contracts", *arXiv, Cornell University 1,* (2021): 1-32. https://doi.org/10.48550/arXiv.2104.03764.

[48] Christopher D. Clack, "Languages for Smart and Computable Contracts", en *Smart Legal Contracts: Computable Law in Theory and Practice* (Oxford: Oxford University Press, 2022), 269-304, https://doi.org/10.1093/oso/9780192858467.003.0013.

profesión jurídica, y de las ciencias computacionales, la respuesta es: depende. Un proyecto puede ser desarrollado en *C++*[49] y funcionar correctamente, o puede ser llevado a cabo en *Python*[50] y funcionar aún mejor –o aún peor–. También el día de mañana podría salir un desarrollo nuevo que pudiera ser incluso aún más funcional para lo que se pretende llevar a cabo.

A nuestro juicio, el foco de interés del abogado y de las ciencias jurídicas no debe ser tanto el lenguaje de programación, sino la lógica jurídica detrás de ella. Como indica Mc Cullagh[51], la labor del abogado no será, en estricto sentido, desarrollar estos Contratos Legales Inteligentes, ni su lógica, ni los requisitos para su funcionamiento. Seguramente, esto seguirá siendo del resorte de la ingeniería y las ciencias computacionales; sin embargo, sí será necesario que el abogado entienda el código computacional para poder entender a qué se obliga su cliente y a qué no. Igualmente, es importante que el abogado pueda entender el contenido del código, pues solo así se estaría más cerca de garantizar que el contenido de una aplicación que desarrolla un contrato jurídico fidedignamente, cumpla con la lógica contractual prevista para el negocio[52].

Por lo tanto, el lenguaje de programación elegido debería ser irrelevante para el abogado. No obstante, un conocimiento generalizado sobre ciencias computacionales y programación sí será necesario para entender la lógica detrás de cada desarrollo puntual y poder formular conceptos jurídicos con suficiencia. A día de hoy, existen múltiples desarrollos de lenguajes enfocados a desarrollar Contratos Legales Inteligentes[53]. Sin embargo, lo cierto es que cualquier lenguaje de programación –al igual que cualquier lenguaje natural como el español, el inglés, el wayuunaiki, etc.– puede servir de base para redactar lógica jurídica. Ya será labor del redactor –y del abogado, si es el caso–, valorar que dicho código se ajuste a las necesidades jurídicas y técnicas del lenguaje de programación. Como prueba de concepto de esto, se mostrará y explicará el funcionamiento de un código redactado en *JavaScript* que refleja la lógica de un contrato de mutuo gratuito.

// 1. Definición de las partes:

var Mutuante; // Define la cuenta del mutuante

---

[49] C++ es un lenguaje de programación de alto nivel y rendimiento utilizado en sistemas operativos y aplicaciones complejas que requieran mayor potencia.

[50] Python es un lenguaje de programación sencillo, popular, utilizado hoy en día para análisis de datos, desarrollo web, inteligencia artificial, entre otros.

[51] Adrian McCullagh, "Redesigning the Contract Lawyer", *SSRN Electronic Journal*, (2021): 1-11. https://dx.doi.org/10.2139/ssrn.3958143.

[52] John Cummins and Christopher D. Clack, "Transforming Commercial Contracts through Computable Contracting", *Journal of Strategic Contracting and Negotiation* 6, n.o 1 (2022): 3-25. https://doi.org/10.1177/20555636211072560.

[53] Por ejemplo, Accord Project, LegalXML, BCL (Business Contract Language), CSL (Contract Specification Language), among others. Al respecto, ver Christopher D. Clack, "Languages for Smart and Computable Contracts", *arXiv, Cornell University 1,* (2021): 1-32. https://doi.org/10.48550/arXiv.2104.03764

```
    var Mutuario; // Define la cuenta del mutuario

// 2. Transferencia del bien fungible

    function transferirDinero (valorTransferencia) {
    // Transfiere una suma de dinero de una cuenta a otra
};

// 3. Pago de la obligación

    function pagoOblogacion (valorDeuda) {
    //Transfiere la sume adeudada de regreso al mutuante
}
```

Como señala Vinay K. Chaudhri, ya son muchas instituciones financieras que celebran contratos de mutuo en este tipo de plataformas, además de la posibilidad de encontrar una aproximación a este contrato que, si bien es diferente a la nuestra, busca el mismo objetivo: desglosar el contrato de mutuo en términos de lógica computacional[54]. En su forma más simple, la lógica de un contrato de mutuo –gratuito– se puede definir de la siguiente manera: 1) la definición de las partes –en la que existe un mutuante y un mutuario–; 2) la transferencia de unos recursos de un mutuante a un mutuario –que perfecciona el contrato real de mutuo–; y 3) la transferencia de los recursos adeudados del mutuario al mutuante que da lugar al pago de la obligación.

Tomando esta estructura como base, se pueden definir las variables y las funciones de forma tal que representen la lógica de un contrato en específico. Para facilitar la explicación, se tomará cada uno de los tres pasos generales y se aplicará a un ejemplo particular. En primera medida, las partes pueden ser definidas con tantos datos como sean deseados. Pueden tener sólo un nombre:

```
    var mutuante = 'Juan Román Riquelme';
    var mutuario = 'Martín palermo';
```

O pueden tener todo tipo de datos que permitan su identificación:

```
    var identificacionMutuante = '10';
    var identificacionMutuario = '09';

    var direccionMutuante = 'Buenos Aires';
    var direccionMutuario = 'Avellaneda';

    var correoMutuante = 'jrr@jrr.com';
    var correoMutuario = 'mp09@mp.com';
```

[54] Vinay K. Chaudhri, "Computable Contracts in the Financial Services Industry".

Pero en cualquier caso, en sentido práctico, la transferencia de bienes –o dinero– se da entre el patrimonio de un mutuante al patrimonio de un mutuario. Puede ser una cuenta bancaria, un *wallet*, o un simple un valor en una base de datos. Para el ejemplo particular, el mutuante tendrá un patrimonio de cien mil –pueden ser pesos, dólares, lo que sea– y el mutuario un patrimonio de 0 –lo que sea–:

```
var patrimonioMutuante = 100000;
var patrimonioMutuario = 0;
```

En un mutuo gratuito, el valor de la transferencia será el valor adeudado por el mutuario. Mientras no se realice la transferencia el valor de la deuda será de cero:

```
var deuda = 0;
```

Una vez definidas las partes en una variable, como segundo paso, se definirá la lógica de la transferencia de recursos que da origen al contrato real de mutuo. Esto puede ser redactado o ejecutado por una función que, si bien podría tener cualquier nombre, la denominaremos "transferirDinero":

```
> function transferirDinero (valorTransferencia) {

}
```

La función transferirDinero se encargará de transferir el dominio de determinado bien fungible del mutuante al mutuario. La lógica de la transferencia de un bien es sencilla: lo que sale de un patrimonio entra en el otro patrimonio. Lo que es transferido por el mutuante es lo que entra en el patrimonio del mutuario. ¿Cuánto se transferirá? El valor que se determine en la función y que, en este caso, se denomina **valorTransferencia**. Por eso, el patrimonio del mutuante será el valor actual de su patrimonio menos el valor de la transferencia:

```
var patrimonioMutuante = patrimonioMutuante - valorTransferencia;
```

En el mismo sentido, el patrimonio del mutuario será el valor actual de su patrimonio más el valor de la transferencia:

```
var patrimonioMutuario = patrimonioMutuarioe + valorTransferencia;
```

Finalmente, en un mutuo gratuito el valor de la transferencia será el valor de la deuda. Por eso, el valor de la deuda será de cero más el valor transferido:

```
var deuda = deuda + valorTransferencia;
```

De esta forma culmina "el segundo paso" que, como se vio, consiste en que una parte transfiere los recursos a la otra. Esto tiene tres efectos: 1) el bien fungible desaparece *o se resta* del patrimonio del mutuante; 2) el bien fungible acrescenta *o se suma* al patrimonio del mutuario; y 3) se genera una obligación de pago *o una deuda* del mutuario al mutuante:

```
function transferirDinero (valorTransferencia) {
    var deuda = deuda + valorTransferencia;
    var patrimonioMutuante = patrimonioMutuante - valorTransferencia;
```

```
    var patrimonioMutuario = patrimonioMutuario + valorTransferencia;
}
```

Como tercer paso para definir la lógica de un contrato de mutuo, se tiene el pago de la obligación. Al igual que en el caso de la transferencia, el pago de la obligación se puede representar con una función:

```
> function pagarObligacion (valorPago) {
}
```

Y al igual que en el caso de la función de transferir, la función pagarObligacion recibe como parámetro el valor que se va a pagar –si bien debería ser el valor adeudado, puede ser otro–. Ahora bien, la lógica del pago de la obligación es la misma que de la transferencia de recursos, pero al revés: 1) el bien fungible aparece *o se suma* al patrimonio del mutuante; 2) el bien fungible desaparece *o se resta* del patrimonio del mutuario; y 3) se elimina *o disminuye* la obligación de pago *o la deuda* del mutuario al mutuante:

```
function pagarObligacion (valorPago) {
    var deuda = deuda - valorPago;
    var patrimonioMutuante = patrimonioMutuante + valorPago;
    var patrimonioMutuario = patrimonioMutuario – valorPago;
}
```

Finalmente, solo queda "llamar" o *hacer funcionar* las funciones creadas:

```
transferirDinero(50000);
pagarObligación(50000);
```

Así las cosas, si se extraen las variables "inventadas" del ejemplo, y se deja únicamente la lógica contractual, el código que representa la lógica del contrato de mutuo, sería la siguiente:

```
var patrimonioMutuante;
var patrimonioMutuario;

var deuda;

function transferirDinero (valorTransferencia) {
    var deuda = deuda + valorTransferencia;
    var patrimonioMutuante = patrimonioMutuante - valorTransferencia;
    var patrimonioMutuario = patrimonioMutuario + valorTransferencia;
}

function pagarObligacion (valorPago) {

    var deuda = deuda - valorPago;
    var patrimonioMutuante = patrimonioMutuante + valorPago;
    var patrimonioMutuario = patrimonioMutuario - valorPago;
```

```
}

transferirDinero ();          pagarObligacion();
```

Ahora bien, dicho código puede ser redactado en otro lenguaje de programación cualquiera. A continuación, se presenta la misma lógica contractual, pero en *Python*:

```
patrimonioMutuante
patrimonioMutuario
deuda

def transferirDinero(valorTransferencia):

    global deuda, patrimonioMutuante, patrimonioMutuario
    deuda = deuda + valorTransferencia
    patrimonioMutuante = patrimonioMutuante - valorTransferencia
    patrimonioMutuario = patrimonioMutuario + valorTransferencia

def pagarObligacion(valorPago):

    global deuda, patrimonioMutuante, patrimonioMutuario
    deuda = deuda – valorPago
    patrimonioMutuante = patrimonioMutuante + valorPago
    patrimonioMutuario = patrimonioMutuario – valorPago

transferirDinero()
pagarObligacion()
```

La traducción se puede realizar con cualquier lenguaje imaginable. Con unos lenguajes de programación será más o menos difícil, y con otros tendrá más o menos aplicación. Así, por ejemplo, se puede reflejar la lógica de contrato de mutuo en *Solidity* y crear un verdadero *Contrato Legal Inteligente* sobre un *Contrato Inteligente* desplegable en una blockchain[55] –en este caso, de transferirse los recursos, seguramente el bien "fungible" sean criptoactivos– :

```
pragma solidity 0.8.0;

Contract contratoDeMutuoGratuito {
    uint patrimonioMutuante;
    uint patrimonioMutuario;
```

[55] Sandra A. Jeskie, y Peter L. Michaelson, "Arbitrating Disputes Involving Blockchains, Smart Contracts, and Smart Legal Contracts", *Dispute Resolution Journal 74*, n.o 4 (2019): 89-134. https://kluwerlawonline.com/journalarticle/Dispute+Resolution+Journal/74.4/DRJ2020031.

```
    uint deuda;

function transferirDinero(uint valorTransferencia) public {
    deuda += valorTransferencia;
    patrimonioMutuante -= valorTransferencia;
    patrimonioMutuario += valorTransferencia;
    }

function pagarOblogacion(uint valorPago) public {
    deuda += valorPago;
    patrimonioMutuante += valorPago
    patrimonioMutuario -= valorPago;
    }
}
```

Como se observa, estos códigos, desarrollados para aspectos totalmente diferentes, pueden reflejar la lógica de un contrato de mutuo. Sin embargo, es posible también utilizar un lenguaje especializado para desarrollar la lógica de un contrato de mutuo. El siguiente código refleja la lógica del mismo contrato de mutuo utilizando el sistema propuesto por el Accord Project[56]:

```
namespace org.accordproject.moneytransfer

concept MoneyTransfer {
    o Long patrimonioMutuante
    o Long patrimonioMutuario
    o Long deuda

function transferirDinero(Long valorTransferencia) {
    this.deuda = this.deuda + valorTransferencia
    this.patrimonioMutuante = this.patrimonioMutuante - valorTransferencia
    this.patrimonioMutuario = this.patrimonioMutuario + valorTransferencia
    }

Function pagarObligacion(Long valorPago) {
    this.deuda = this.deuda - valorpago
    this.patrimonioMutuante = this.patrimonioMutuante + valorPago
    this.patrimonioMutuario = this.patrimonioMutuario - valorPago
    }
}
```

[56] "*Accord Project es una iniciativa open source, sin ánimo de lucro enfocada a transformar la administración y automatización de contratos a través de la digitalización de los contratos*" Traducción propia de la documentación del proyecto.

Así pues, cualquier lenguaje de programación puede servir como base para redactar lógica jurídica en términos que pudieran servir para un desarrollo ejecutable. Estos pueden ser más o menos complejos, tener más o menos funcionalidades, y tener un mejor o peor funcionamiento. Igualmente, se pueden agregar interfaces de usuario que permitan a un usuario interactuar con los contratos, modificar las cantidades a transferir, pagar, modificar las partes, etc.

En este orden de ideas, es claro que cualquier lenguaje puede servir para representar la lógica de un contrato jurídico. También lo es que es posible tener un contrato más o menos complejo, con accesibilidad al usuario más o menos agradable, y que se pueda desplegar sobre un sistema general o específico. El siguiente punto por resolver sería de qué manera se puede tomar un lenguaje de programación para desarrollar Contratos Legales Inteligentes.

Para efectos de este trabajo, se expondrá la propuesta de la *Linux Foundation* –ahora propiedad de DocuSign–, el Accord Project, como uno de los lenguajes propuestos para dar respuesta a la pregunta planteada. Lo anterior, al tratarse deun lenguaje especializado que está enfocado en la automatización de contratos y que puede abrir la puerta a muchos desarrollos[57]. Sin embargo, como ya se precisó, en principio, podría ser cualquier otro disponible en el mercado.

4.1. El Accord Project para herramienta para la redacción, celebración y ejecución de Contratos Legales Inteligentes.

El Accord Project es un proyecto Open Source que sirve para la redacción, celebración y ejecución de Contratos Legales Inteligentes. Consideramos que su plataforma permite satisfacer los requisitos legales y técnicos para este tipo de contratos. De acuerdo con su documentación, "*Accord Project es una iniciativa open source, sin ánimo de lucro enfocada a transformar la administración y automatización de contratos a través de la digitalización de los contratos*"[58]. El Accord Project propone una estructura tripartita para la *elaboración* de un mismo Contrato Legal Inteligente. Esta estructura consta de una plantilla o modelo de contrato que describe en lenguaje natural y con variables, el contenido obligacional; un modelo de datos que guarda la información de las variables a ejecutar; y un lenguaje de programación en estricto sentido, que permite la declaración y ejecución de estamentos lógicos.

En ese sentido, el proyecto dispone de tres principales herramientas que permiten realizar esto: Cicero, Concerto y Ergo. Estos tres lenguajes representan la estructura tripartita

[57] Vinay K. Chaudhri, "Computable Contracts in the Financial Services Industry".

[58] Traducción propia: *Accord Project is an open source, non-profit initiative aimed at transforming contract management and contract automation by digitizing contracts. It provides an open, standardized format for Smart Legal Contracts. The Accord Project defines a notion of a legal template with associated computing logic which is expressive, open-source, and portable. Accord Project templates are similar to a clause or contract template in any document format, but they can be read, interpreted, and run by a computer*".

propuesta en el proyecto: la plantilla del texto (Cicero), la plantilla del modelo de datos (Concerto) y la plantilla de la lógica (Ergo):

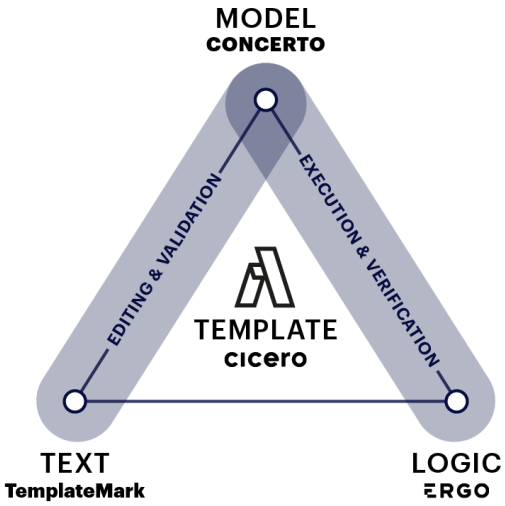

Figura 4: *Relación entre Cicero, Concerto y Ergo.* Figura de Niall Roche et al, 2021[59]

4.2. Funcionamiento y finalidad de Cicero

Cicero es un sistema diseñado para permitir la creación de plantillas de Contratos Legales Inteligentes a partir de contratos legalmente exigibles, utilizando una sintaxis similar a la del lenguaje natural[60]. En otras palabras, Cicero permite convertir cualquier texto contractual redactado en lenguaje natural en un "template" declarativo de un contrato estándar, de forma tal que pueda ser unido a un modelo de datos. En términos simples: todos hemos utilizado modelos de contratos y hemos ajustado el modelo al caso particular de un cliente. Esto funciona así porque el contrato está compuesto de un texto contractual –estándar– y unas variables –estándar– que, al ser completadas por datos –particulares–, dan vida al contrato.

El siguiente texto puede corresponder a una cláusula de pago en un contrato cualquiera: "Juan Palermo pagará a Martín Riquelme la suma de setecientos setenta y seis mil cuatrocientos veinte (776,420.00) pesos". De ser necesario, un abogado podría utilizar dicho texto contractual y ajustarlo a un contrato que necesite para otro cliente. Por ejemplo: "Alfonso Pérez pagará a Omar Cañón la suma de mil novecientos setenta y cinco (1.975) dólares". Igualmente, dicho texto podría ser formulado de forma genérica para que cualquier persona pudiera utilizarlo cambiando los datos y sin dejar de lado dato específico alguno. Por ejemplo: "EL COMPRADOR pagará a EL VENDEDOR la suma de VALORAPAGAR".

---

[59] Niall Roche, Walter Hernandez, Eason Chen, Jérôme Siméon y Dan Selman, "Ergo - a programming language for Smart Legal Contracts", *arXiv, Cornell University,* (2021): 1-3. https://doi.org/10.48550/arXiv.2112.07064.

[60] Roche, Hernandez, Chen, Siméon y Selman, "Ergo - a programming language for Smart Legal Contracts".

Este mismo texto puede ser redactado utilizando Cicero, de forma tal que pueda ser entendido por un ser humano y, a su vez, vincularlo a un modelo de datos utilizando Concerto. Por ejemplo: {COMPRADOR} pagará a {VENDEDOR} la suma de {VALORAPAGAR}. De esta forma, es posible crear una plantilla o "template" de un contrato en Cicero a partir de un contrato cualquiera:

{EL VENDEDOR} transferirá a {EL COMPRADOR} el dominio de {EL BIEN}. {EL COMPRADOR} pagará a {EL VENDEDOR} la suma de {VALOR A PAGAR} el día {DÍA DE PAGO} a través de una consignación bancaria en la {CUENTA BANCARIA}.

4.3. **Funcionamiento y finalidad de Concerto**

Concerto es un lenguaje específicamente diseñado para describir y definir las estructuras de datos, las relaciones y las reglas de negocio que se aplican en un contexto particular[61]. Todo contrato, o mejor, toda operación tiene unos datos inherentes a su funcionamiento. El modelo de datos está compuesto por todas las variables que intervienen en la operación, organizados para describir la estructura del –en este caso– contrato.

Continuando con el ejemplo de modelos de contratos, al momento de descargar uno, se cambian los datos para cada caso –los nombres, el precio, periodicidad, etc.–. Estos datos que los abogados hemos reemplazado en un modelo de contrato múltiples veces son aquellos datos que componen el modelo de datos de un contrato.

EL ARRENDADOR entrega para su uso y goce a EL ARRENDATARIO EL INMUEBLE destinado a vivienda, ubicado en la dirección CALLE 21 # 16-45 en el Barrio Palermo, identificado con la Ficha catastral No. 123 Matrícula inmobiliaria No. 321, ciudad de Manizales, Departamento de Caldas, por el término de un (1) año. El contrato empezará el 1 de enero de 2023 y terminará el 31 de diciembre de 2023. EL ARRENDATARIO deberá pagar a EL ARRENDADOR la suma de tres millones de pesos (3'000.000.00 COP) mensuales a EL ARRENDADOR, dentro de los primeros cinco (5) días de cada mes en la cuenta ABC123.

Este contrato de arrendamiento está compuesto por lo menos, por las siguientes variables que intervienen:

1. El arrendador [EL ARRENDADOR]
2. El arrendatario [EL ARRENDATARIO]
3. El bien inmueble destinado a vivienda [EL INMUEBLE]
4. El pago por el uso y goce del inmueble
5. Fecha de inicio
6. Fecha de finalización

[61] Roche, Hernandez, Chen, Siméon y Selman, "Ergo - a programming language for Smart Legal Contracts".

7. Fecha de pago

A su vez, estas variables pueden ser desagregadas y ampliadas para tener una información más completa.

1. Un arrendador
    1.a. Nombre del arrendador
    1.b. Identificación del arrendador
    1.c.…
2. Un arrendatario
    2.a. Nombre del arrendatario
    2.b. Identificación del arrendatario
    2.c.…
3. Un bien o servicio –supongamos una casa–
    3.a. Tipo de bien
    3.b. Identificación del bien
    3.c.…
4. El pago por el uso y goce del bien o servicio
    4.a. Valor
    4.b. Divisa
    4.c.…
5. Fecha de inicio
6. Fecha de finalización
7. Fecha de pago
    7.a. Anual
    7.b. Mensual
    7.c.…

Así, el arrendador será una variable –o un dato– compuesta por varias características –o datos, o variables–: un nombre, una identificación, un teléfono, etc. La suma de todas estas variables es lo que permite que se pueda identificar el contrato y que este logre su cometido de acuerdo con un caso específico.

Para el caso de Concerto, a diferencia de Cicero, lo importante no es el texto contractual, sino las variables que componen el contrato. Con estas variables, es posible crear el modelo de datos que estructuran el contrato, la columna vertebral del contrato, que, junto con los datos particulares, darán vida al negocio jurídico. A través de Concerto es posible hacer un modelo de datos de un contrato "modelo" o "template" y, posteriormente, ingresar a la computadora los datos de cada caso específico.

```
namespace org.example
    asset ContratoArrendamiento identified by contratoId {
     o String contratoID
     o TipoArrendamiento tipoArrendamiento
     --> Arrendador arrendador
     --> Arrendatario arrendatario
     --> Propiedad propiedad
     o DateTime fechaInicio
     o DateTime fechaFin
     o Double rentaMensual
    }
    enum TipoArrendamiento {
     o RESIDENCIAL
     o COMERCIAL
    }
    participant Arrendador identified by arrendadorId {
     o String nombreArrendador
     o String dirección
    }
    participant Arrendatario identified by arrendatarioId {
     o String nombreArrendatario
     o String dirección
    }
    concept Propiedad {
```

o String idPropiedad

o String nombrePropiedad

o String dirección

}

Con base en este “template” es posible crear un “modelo” para un contrato de arrendamiento de vivienda urbana de un caso particular. Así se pudiera ingresar cualesquiera datos que desee o necesite para cada variable. Por ejemplo:

```
{
 "$class": "org.example.ContratoArrendamiento",
 "leaseId": "CONTRATO-123",
 "tipoArrendamiento": "RESIDENCIAL",
 "arrendador": {
   "$class": "org.example.Arrendador",
   "lessorId": "ARRENDADOR-001",
   "nombreArrendador": "Juan Pérez",
   "dirección": "Calle Principal 123"
 },
 "arrendatario": {
   "$class": "org.example.Arrendatario",
   "lesseeId": "ARRENDATARIO-001",
   "nombreArrendatario": "María López",
   "dirección": "Avenida Central 456"
 },
 "propiedad": {
   "$class": "org.example.Propiedad",
   "idPropiedad": "PROPIEDAD-001",
   "nombrePropiedad": "Casa Azul",
   "dirección": "Calle Secundaria 789"
 },
```

"fechaInicio": "2023-07-01T00:00:00.000Z",

"fechaFin": "2024-06-30T00:00:00.000Z",

"rentaMensual": 1000.00

}

4.4. Funcionamiento y finalidad de Ergo

Ergo es el lenguaje de programación utilizado para definir la lógica y los términos de un Contrato Legal Inteligente[62]. Ergo es quien realmente permite la ejecución del contrato en una plataforma. A diferencia de Cicero y Concerto, que funcionan de forma declarativa, Ergo funciona de forma condicional. Es decir, Ergo es quien realmente lee y ejecuta la lógica detrás de un contrato. Con Cicero y Concerto es posible declarar que:

EL PROVEEDOR} entregará a {EL BENEFICIARIO} cada {PERIODICIDAD} la cantidad de {NUMERO DE BIENES} {BIENES} dentro del término de {PLAZO}. En caso de incumplimiento, {EL PROVEEDOR} pagará a {EL BENEFICIARIO} una multa de {VALOR MULTA}.

Este texto declara la obligación de un proveedor de entregar unos bienes a el beneficiario y, de no hacerlo dentro de los términos, estará obligado a pagar una multa en favor del beneficiario. Por sí solo, este texto simplemente declara dos obligaciones, sin embargo, no las ejecuta. Igualmente, de ser un contrato "estándar" sería necesario acudir a la jurisdicción para buscar su ejecución forzada. Para que un sistema fuera capaz de ejecutar este texto contractual sería necesario introducirle la lógica del negocio usando un lenguaje de programación, como sería Ergo. En este caso, la lógica del contrato puede ser planteada como dos condicionales:

1. Si se acuerda el contrato, se entregarán a {EL PROVEEDOR} la cantidad de {NUMERO DE BIENES} {BIENES} dentro del término de {PLAZO}.
2. Si no se entrega la cantidad de {NUMERO DE BIENES} {BIENES} dentro del término de {PLAZO}, se pagará una multa de {VALOR MULTA}.

En otras palabras, se puede decir:

Si:

[se acuerda la obligación]

Entonces:

[se entregan los bienes]

Si:

[62] Roche, Hernandez, Chen, Siméon y Selman, "Ergo - a programming language for Smart Legal Contracts".

[no se entregan los bienes]

Entonces:

[se paga una multa]

Esta lógica puede ser trasladada a un lenguaje de programación y, posteriormente, ser integrada a un sistema que la entienda y ejecute. Así, por ejemplo, sería posible redactar la lógica de esta obligación utilizando JavaScript:

```
if (obligacion == true) {
        entregarBienes( );
}
if (obligación == true && entregaBienes == false) {
        pagarMulta( );
}
```

En este caso, si existe una obligación, entonces entregue los bienes. Si existe una obligación y no se entregaron los bienes, pague una multa. Utilizando el entorno de Accord Project es posible codificar esta lógica en el lenguaje de programación Ergo, ligarlo a un modelo de contrato declarado en Cicero, y cuyas variables se encuentran en un modelo de datos de Concerto. El texto de Ergo se podría definir de la siguiente manera:

```
contract contratoSuministro identified by contratoId {
 o String contratoId
 o Boolean obligacion
 o Boolean entregaBienes
 clause entregarBienesClausula {
  enforce obligacion
 }
 clause pagarMultaClausula {
  enforce obligacion && entregaBienes == false
 }
 ensure entregarBienesClausula {
  action {
   entregarBienes( );
  }
```

```
}
ensure pagarMultaClausula {
  action {
    pagarMulta( );
  }
}
private action entregarBienes() {
  // Lógica para entregar los bienes
}
private action pagarMulta() {
  // Lógica para pagar la multa
}
}
```

5. Métodos para la redacción de Contratos Legales Inteligentes en Accord Project

Como se ha insistido a lo largo del escrito, *crear* un Contrato Legal Inteligente no es otra cosa diferente que plasmar las reglas de un *contrato* en una *aplicación* o *software.* En otras palabras, de lo que se trata es de *traducir* el contenido de un contrato a un lenguaje que pueda ser entendido y ejecutado por una máquina. Tomando esto como premisa, se tomará como punto de referencia el método ("*algorithmic approach*") propuesto por Toledo Correa para la redacción de Contratos Legales Inteligentes comerciales[63]. El autor propone una serie de cuatro pasos para transformar un contrato comercial escrito en lenguaje natural en un contrato computable y entendible por computadoras. El procedimiento consta de los siguientes pasos[64]:

(i) Separara el texto contractual en cláusulas, las cláusulas en párrafos, y, a su vez, los párrafos en frases;

(i) Clasificar las oraciones en los diferentes tipos de afirmaciones propuestas por el autor;

(i) Estructurar el flujo de cálculos y condiciones descritas en el contrato;

---

[63] Pedro Felipe Toledo Correa, "Transforming Natural Language Stateless Commercial Contracts into Computer Computable Contracts" (Trabajo de grado Maestría en Artes Liberales con Especialización en Estudios de Extensión, 2022), https://nrs.harvard.edu/URN-3:HUL.INSTREPOS:37371701.

[64] Pedro Felipe Toledo Correa, "Transforming Natural Language Stateless Commercial Contracts into Computer Computable Contracts".

(i) Transformar la estructura de datos en un texto redactado en un lenguaje computable por un computador.

A través de este método es posible plantear un contrato en la forma de un Contrato Legal Inteligente. Sin embargo, se considera que una etapa intermedia, complementaria o adicional, al método puede facilitar mucho más la labor: la redacción de la lógica contractual en seudocódigo. Esto, en muchos casos, no exigirá interacción entre el abogado y el desarrollador o ingeniero que efectivamente *cree* el contrato. Si bien lo ideal sería que, cada vez más, ambas profesiones se acercaran al conocimiento y entendimiento del otro, lo cierto es que en la mayoría de los casos esto no será más que una utopía.

En términos generales, dentro de los primeros tres pasos el objetivo será construir un modelo del contrato, en términos de un objeto –muy similar a una estructura *JSON*[65]–. Es decir, se planteará la información contenida en el contrato en forma de estructura de datos. Una vez se tiene este modelo –u objeto *JSON*– el último paso consiste en transcribir el modelo en los diferentes lenguajes de programación –o, en este caso, Ergo–.

Si bien no es del resorte de este escrito entrar a analizar a profundidad el método planteado por Toledo Correa[66] –ni ningún otro método en particular–, se resalta la importancia –o incluso la necesidad– de continuar la investigación sobre posibles métodos estandarizados para la redacción de Contratos Legales Inteligentes. Consideramos que, solo en la medida en que se pueda masificar la producción de estos contratos, serán realmente viables y prácticos en la vida diaria.

6. Conclusiones

Los Contratos Legales Inteligentes son una realidad que vemos día a díaen diferentes industrias. Si bien las utopías de un sistema contractual universal unificado y descentralizado parecen más lejanas de lo que se planteaban en el auge de los Contratos Inteligentes, lo cierto es que esquemas de implementación más básicos –técnicamente hablando– ya se encuentran funcionando y son de amplia aceptación y uso en la sociedad.

Como conclusión, podemos afirmar que los Contratos Legales Inteligentes pueden ser definidos como una serie de instrucciones que procesa un software y que tienen como finalidad generar efectos jurídicos (a diferencia de los Contratos Inteligentes que, aunque designan las reglas de funcionamiento de un sistema, no buscan directa ni indirectamente generar efectos jurídicos). Existen distintos tipos de Contratos Legales Inteligentes y su clasificación depende de cada doctrinante. Sin embargo, se pueden clasificar de la siguiente forma (i) contratos de sólo código; (ii) contratos híbridos; y (iii) contratos en lenguaje natural con desempeño automático.

---

[65] JSON proviene de las siglas “JavaScript Object Notation”, o “Notación de Objetos para JavaScript”(traducción propia). JavaScript es un lenguaje de programación. “JSON” o un archivo “JSON” es una forma en la que se puede presentar información -en forma de objeto- en el lenguaje JavaScript. De esta forma, es más sencillo almacenar y transmitir datos.

[66] Pedro Felipe Toledo Correa, “Transforming Natural Language Stateless Commercial Contracts into Computer Computable Contracts“.

Esta clasificación permite entender no sólo que los Contratos Legales Inteligentes son una realidad con la que convivimos a diario, sino también cuáles son los requerimientos necesarios para poner en práctica estas herramientas. Así, podemos concluir que, bajo la legislación actual, es válido celebrar y ejecutar Contratos Legales Inteligentes en Colombia.

Bajo el estado de arte tecnológico, podemos afirmar que es posible redactar Contratos Legales Inteligentes y, para ello, sólo es necesario un lenguaje de programación. Para facilitar y mejorar esta tarea, muchas herramientas se desarrollan con el paso del tiempo -Accord Project, por ejemplo-. Igualmente, algunos doctrinantes ya empiezan a desarrollar métodos, que permiten acercar más al abogado, ingeniero, programador, etc. a la respuesta de cómo redactar lógica jurídica en un lenguaje de programación. Todo esto facilita la labor y nos acerca mucho más al objetivo y a la respuesta a nuestra respuesta inicial. Sin embargo, al final del día, para redactar un Contrato Legal Inteligente, basta con desglosar la lógica del contrato comercial que se desee, obligación por obligación, y expresarla en el lenguaje de programación de su preferencia.

Se destaca que, más que "descifrar" los Contratos Legales Inteligentes, estos requieren de una desmitificación que acerque más al abogado al contenido propio de la aplicación. Es decir, más que el abogado directamente "redacte" o "programe" los Contratos Legales Inteligentes, implica que el equipo jurídico pueda validar y trabajar conjuntamente con el equipo de IT para asegurarse que el funcionamiento de un sistema responda a las necesidades contractuales, legales y técnicas de la transacción[67].

Esto no excluye, por supuesto, que, con el paso del tiempo, se empiecen a generalizar mucho más otras implementaciones más robustas que permitan que el código refleje con mayor exactitud y relación el contenido obligacional y legal. En ambos casos, de seguir usando los Contratos Legales Inteligentes como los hemos venido usando –lenguaje natural y código–, o de implementar sistemas de contratación en código –puros o híbridos–, se requiere de un conocimiento técnico mínimo del abogado que le permita, por lo menos, entender el código de un sistema, teniendo en cuenta la neutralidad tecnológica como principio regidor de este tipo de contratos[68].

[67] Abel B. Veiga Copo, "«Smart contract» y contrato de seguro. Una ecuación asimétrica y no sólo algorítmica".

[68] Juan Carlos Reyes Garcia, "Smart Legal Contracts: From Theory to Reality*", University of California, Los Angeles School of Law, Law & Economics Research Paper Series,* (2022): 1-9. https://dx.doi.org/10.2139/ssrn.4073055.

Bibliografía:

Cummins, John and Clack, Christopher D. "Transforming Commercial Contracts through Computable Contracting". *Journal of Strategic Contracting and Negotiation* 6, n.o 1 (2022): 3-25. https://doi.org/10.1177/20555636211072560.

Chaudhri, Vinay K. "Computable Contracts in the Financial Services Industry", *arXiv, Cornell University,* (2022): 1-10. https://doi.org/10.48550/arXiv.2208.04685.

Chávez Mendoza, Anthony Gustavo. "El arbitraje como método de solución de controversias derivados del Smart Contract", Trabajo de grado pregrado en Derecho, Universidad Católica de Santiago de Guayaquil, 2022. http://repositorio.ucsg.edu.ec/handle/3317/18600.

Clack, Christopher D. "Languages for Smart and Computable Contracts", *arXiv, Cornell University 1,* (2021): 1-32. https://doi.org/10.48550/arXiv.2104.03764.

Clack, Christopher D. "Languages for Smart and Computable Contracts". En *Smart Legal Contracts: Computable Law in Theory and Practice,* 269-304. Oxford University Press, 2022. https://doi.org/10.1093/oso/9780192858467.003.0013.

De Souza Mello, Gustavo Marchi. "Smart Legal Contracts: Einsatzmöglichkeiten im recht und rechtliche Rahmenbedingungen". *Revista da Faculdade de Direito, Universidade de São Paulo 115,* (2020): 751-775. https://doi.org/10.11606/issn.2318-8235.v115p751-775.

Díaz Baquero, Viviana Paola. "Regulación de los Contratos Inteligentes en Colombia" Trabajo de grado especialista en Derecho Comercial, Pontificia Universidad Javeriana, 2019). https://repository.javeriana.edu.co/bitstream/handle/10554/46186/Trabajo%20de%20grado.pdf.

Feliu Rey, Jorge. "Smart Contract: Concepto, ecosistema y principales cuestiones de Derecho privado". *La Ley Mercantil,* n.o 47 (2018): 1-27. https://ssrn.com/abstract=3247775.

Fetsyak, Ihor. "Contratos Inteligentes: análisis jurídico desde el marco legal español," *Revista Electrónica de Derecho de la Universidad de La Rioja (REDUR)*, n.o 18 (2020): 197-236. https://doi.org/10.18172/redur.4898.

García Rubio, Luis Felipe. "Contratos Inteligentes en blockchain. Una propuesta de lege data para el derecho privado colombiano en materia contractual". *Anuario de Derecho Privado 01*, n.o 2 (2020): 9–45. Dx.doi.org/10.15425/2017.350.

Hernández Díaz, Julián Leonardo. "Nulidad por objeto ilícito en el smart-contract: el rol del juez en un contrato irreversible" Trabajo de grado pregrado en Derecho Civil, Universidad Externado de Colombia, 2020. https://doi.org/10.57998/bdigital.handle.001.4262.

Jeskie, Sandra A. y Michaelson, Peter L. "Arbitrating Disputes Involving Blockchains, Smart Contracts, and Smart Legal Contracts". *Dispute Resolution Journal 74*, n.o 4 (2019): 89-134. https://kluwerlawonline.com/journalarticle/Dispute+Resolution+Journal/74.4/DRJ2020031.

Kaulartz, Markus. "Smart Contract Dispute Resolution". En *Smart Contracts*, 73-84. Tubinga: Mohr Siebeck, 2019. http://www.jstor.org/stable/j.ctvn96h9r.8.

Kuhlmann, Nico. “Smart Enforcement bei Smart Contracts”. En *Smart Contracts*, 117-124. Tubinga: Mohr Siebeck, 2019. https://www.jstor.org/stable/j.ctvn96h9r.11.
Law Commission. *Smart Legal Contracts. Advice to Goverment* (United Kingdom: Law Commission, 2021). https://cloud-platform-e218f50a4812967ba1215eaecede923f.s3.amazonaws.com/uploads/sites/30/2021/11/Smart-legal-contracts-accessible.pdf.
—“Smart Contracts”. Consultado el 22 octubre de 2024. https://lawcom.gov.uk/project/smart-contracts/.
Legerén-Molina, Antonio. “Los Contratos Inteligentes en España (La disciplina de los Smart Contracts)”. *Revista de Derecho Civil 5*, n.o 2 (2018): 193-241. https://www.nreg.es/ojs/index.php/RDC/article/download/320/267.
Luque Restrepo, Daniel. “Análisis de las obligaciones condicionales en la implementación de Smart Contracts en el ordenamiento jurídico colombiano” Trabajo de grado pregrado en Derecho, Universidad Pontificia Bolivariana, 2020.
Matzke, Robin. “Smart Contracts statt Zwangsvollstreckung? Zu den Chancen und Risiken der digitalisierten privaten Rechtsdurchsetzung”. En *Smart Contracts,* 99-116. Tubinga: Mohr Siebeck, 2019. https://www.jstor.org/stable/j.ctvn96h9r.10.
McCullagh, Adrian. “Redesigning the Contract Lawyer”. *SSRN Electronic Journal*, (2021): 1-11. https://dx.doi.org/10.2139/ssrn.3958143.
Moradinejad, Reza. ”Le contrat intelligent, nouveau vecteur de confiance dans les relations contractuelles: réalité ou rêve?*”. Les Cahiers de droit 60,* n.o 3 (2019): 623-651. https://doi.org/10.7202/1064651ar.
Morales Higuita, Liced, y Pulgarín Agudelo, Natalia Isabel. “Smart Contracts en materia de arrendamiento de vivienda urbana a partir de los artículos 1, 3 y 22 de la Ley 820 de 2003” Trabajo de grado pregrado en Derecho, Caldas, Unilasallista Corporación Universitaria, 2022. https://repository.unilasallista.edu.co/items/de54f029-5856-429d-96b5-9ad38dbf4d31.
Presidente de la República de Colombia. Decreto legislativo N° 410 de 1971: Por el cual se expide el Código de Comercio, Art. 845-863. (el 27 de marzo de 1971).
Redondo-Luque, José Ricardo. “Análisis jurídico de los Smart Contracts o Contratos Inteligentes”. *Revista Acta Académica,* n.o 70 (2022): 169-186. http://revista.uaca.ac.cr/index.php/actas/article/view/1343.
Reyes Garcia, Juan Carlos. “Smart Legal Contracts: From Theory to Reality*”. University of California, Los Angeles School of Law, Law & Economics Research Paper Series,* (2022): 1-9. https://dx.doi.org/10.2139/ssrn.4073055.
Riehm, Thomas. “Smart Contracts und verbotene Eigenmacht”. En *Smart Contracts, 85-98.* Tubinga: Mohr Siebeck GmbH and Co. KG, 2019. https://www.jstor.org/stable/j.ctvn96h9r.9.
Rincón, Erick y Martínez, Valeria. “Contratos inteligentes y automatización como desarrollos aplicados del legaltech en Colombia”. *Revista Direito GV,* (2022): 239-260. https://doi.org/10.1590/2317-6172202211.
Roche, N., Hernandez, W., Chen, E., Siméon, J., y Selman, D. “Ergo - a programming language for Smart Legal Contracts”. *arXiv, Cornell University,* (2021): 1-3. https://doi.org/10.48550/arXiv.2112.07064.
Sánchez-Gómez, N., Torres-Valderrama, J. Mejías Risoto, M, y Garrido, A. “Blockchain

smart contract meta-modeling*". Journal of Web Engineering 20,* n.o 7 (2021): 1-18. 10.13052/jwe1540-9589.2073.

Toledo Correa, Pedro Felipe. "Transforming Natural Language Stateless Commercial Contracts into Computer Computable Contracts" Trabajo de grado Maestría en Artes Liberales con Especialización en Estudios de Extensión, 2022. https://nrs.harvard.edu/URN-3:HUL.INSTREPOS:37371701.

Veiga Copo, Abel B. "«Smart contract» y contrato de seguro. Una ecuación asimétrica y no sólo algorítmica". *Revista de Derecho del Sistema Financiero*, (2020): 119-184. http://hdl.handle.net/11531/53117.