



Universidad Autónoma
del Estado de México

Universidad Autónoma del Estado de México



FACULTAD DE INGENIERÍA
DOCTORADO EN CIENCIAS DE LA INGENIERÍA

Un modelo para la resolución de conflictos en disputas
legales sobre propiedad intelectual utilizando un Sistema
Multiagentes

RODRIGO VIDAL LÓPEZ

TESIS

que para obtener el grado de:
Doctor en Ciencias de la Ingeniería

DIRECTOR DE TESIS	DR. JOSÉ ANTONIO HERNÁNDEZ SERVÍN
TUTORES ADJUNTOS	DRA. VIANNEY MUÑOZ JIMENÉZ
	DR. MARCO ANTONIO RAMOS CORCHADO
ASESORES EXTERNOS	DR. JACOBO LEONARDO GONZÁLEZ RUÍZ
	DR. DAVID VALLE CRUZ

Toluca, México, 03 Diciembre 2025



Índice general

Resumen de la tesis	1
I Introducción	2
I.1 Planteamiento del problema	2
I.2 Justificación	4
I.3 Objetivos	5
I.4 Hipótesis	6
I.5 Contribuciones esperadas	6
I.6 Organización del documento	7
II Marco conceptual y estado del arte	8
II.1 Resolución de disputas en propiedad intelectual	9
II.2 Sistemas Multiagente (SMA) y negociación	11
II.3 Plataformas de resolución en línea de disputas (ODR)	13
II.4 Modelos de argumentación y razonamiento jurídico (AI & Law)	15
II.5 Bases lógicas y semánticas para el modelo propuesto	17
II.6 Lenguajes y plataformas para implementación	19
II.7 Brechas identificadas en la literatura	21
III Modelo propuesto	23
III.1 Arquitectura general del sistema multiagente	24
III.2 Formalización de las normas jurídicas en el modelo	27
III.3 Protocolo de negociación y mediación normativa	30
III.4 Propiedades y criterios de evaluación	32
IV Productos obtenidos	35
IV.1 Inclusión de la inteligencia artificial en disputas legales sobre propiedad intelectual <i>Komputer Sapiens</i>	36
IV.2 Towards a Formal Language in Law <i>IGI Global</i>	37

IV.3 Modelling Legal Judgements with BDI Agents in Mercury: Analysis of the Pierson Versus Post Case <i>Lecture Notes in Networks and Systems</i>	39
V Evaluación experimental y discusión	42
V.1 Diseño experimental	44
V.2 Escenarios de prueba	46
V.3 Métricas de evaluación del modelo	48
V.3.1 Coherencia normativa	49
V.3.2 Trazabilidad lógica	49
V.3.3 Adaptabilidad deliberativa	50
V.3.4 Explicabilidad jurídica	50
V.4 Resultados	51
V.4.1 Configuración inicial del agente Tomkins	52
V.4.2 Generación inicial de argumentos	54
V.4.3 Simulación de ataque y reevaluación de argumentos	56
V.4.4 Correspondencia doctrinal y análisis de resultados	59
VI Conclusiones y trabajo futuro	61
VI.1 Síntesis del modelo y su alcance	61
VI.2 Desarrollo y articulación de los componentes del modelo	63
VI.3 Aportaciones teóricas, metodológicas y aplicadas	65
VI.4 Limitaciones y proyección del trabajo futuro	67
Referencias	70
Anexos	77

Resumen de la tesis

El avance acelerado de la inteligencia artificial ha impulsado su adopción en el ámbito jurídico, particularmente en tareas de análisis documental, predicción de resultados y resolución de disputas. Sin embargo, la mayoría de estos sistemas —basados en redes neuronales o enfoques estadísticos— carecen de explicabilidad y trazabilidad normativa, lo que limita su legitimidad jurídica y su utilidad en contextos donde la fundamentación racional es indispensable. Frente a ello, esta investigación propone un modelo formal de razonamiento jurídico computacional basado en agentes BDI (*Belief–Desire–Intention*), capaz de representar y reevaluar decisiones conforme a principios normativos explícitos.

El objetivo general fue modelar computacionalmente el razonamiento jurídico en disputas de propiedad intelectual mediante un sistema multiagente BDI, integrando marcos teóricos de Hans Kelsen (jerarquía normativa), la lógica contextual (IHL), la semántica de mundos posibles (Kripke) y los estándares de interoperabilidad FIPA. El modelo se formalizó y validó mediante su implementación en el lenguaje lógico-declarativo **Mercury**, elegido por su tipado fuerte, determinismo y capacidad de trazabilidad.

Como estudio de caso se utilizó *Pierson v. Post* (1805), un precedente clásico del derecho de propiedad, implementando un agente denominado *Tomkins*, que genera, evalúa y reevalúa argumentos jurídicos ante variaciones en sus creencias y condiciones normativas. La evaluación cualitativa, centrada en criterios de coherencia normativa, trazabilidad lógica, adaptabilidad deliberativa y explicabilidad jurídica, demostró que el modelo puede sostener procesos argumentativos consistentes y normativamente fundados, aun sin resolver la disputa completa.

La investigación aporta una metodología interdisciplinaria entre Filosofía, Derecho y Computación, validando que la formalización del razonamiento jurídico puede ejecutarse de forma explicable y verificable. El modelo constituye una base para el desarrollo de sistemas multiagente deliberativos, capaces de representar la negociación y mediación jurídica entre distintas doctrinas. En un contexto donde los modelos de lenguaje (LLMs) producen resultados sin justificación semántica, este trabajo representa un paso hacia una inteligencia artificial jurídica más transparente, justificable y conforme al razonamiento normativo humano.

Palabras clave: Razonamiento jurídico, sistema multiagente, BDI, lógica normativa, Mercury, explicabilidad, *Pierson v. Post*.

Capítulo I

Introducción

I.1. Planteamiento del problema

La aceleración tecnológica contemporánea ha desafiado las estructuras tradicionales del Derecho. La aparición de tecnologías digitales, la proliferación de contenidos generados por inteligencia artificial (IA) y la globalización de los mercados de información han intensificado las disputas en materia de propiedad intelectual (PI). Sin embargo, mientras la tecnología avanza a un ritmo exponencial, el Derecho continúa sujeto a procesos de deliberación legislativa, jurisprudencial y doctrinal mucho más lentos [1, 2]. Esta asimetría genera un desfase que no solo repercute en la seguridad jurídica, sino también en la legitimidad de los mecanismos de resolución de conflictos.

Los sistemas jurídicos, por su propia naturaleza, se enfrentan a la tensión entre certeza y flexibilidad. El caso paradigmático *Pierson v. Post* (1805) es ilustrativo en este sentido: un conflicto aparentemente trivial —la disputa por la caza de un zorro— derivó en interpretaciones jurídicas opuestas dentro de la misma Corte [3]. Mientras el juez Tompkins adoptó una visión estrictamente formalista, el juez Livingston sostuvo una perspectiva más pragmática y utilitarista. Este caso histórico evidencia que el Derecho no produce una única respuesta correcta, sino un espacio de argumentación en el que coexisten soluciones alternativas, cada una respaldada por un marco normativo y axiológico distinto.

En la actualidad, disputas semejantes emergen en un entorno mucho más complejo. La IA generativa plantea interrogantes sin precedentes: ¿a quién pertenece una obra producida por un modelo algorítmico?, ¿cómo proteger los derechos de autores humanos cuando los sistemas automáticos reproducen, transforman o imitan su creación? Las plataformas comerciales de análisis contractual impulsadas por IA, como Kira Systems o Luminance, muestran que la tecnología ya participa en procesos de interpretación y gestión documental [4, 5]. Asimismo,

herramientas de búsqueda y predicción de resultados judiciales como Casetext o CaseMine han comenzado a ofrecer ventajas operativas para abogados y jueces [6, 7]. No obstante, todas estas soluciones privilegian la eficiencia y la explotación masiva de datos, sin ofrecer un fundamento normativo explícito que legitime sus resultados.

El campo de la Inteligencia Artificial y el Derecho (AI & Law) ha desarrollado modelos de razonamiento lógico, esquemas de argumentación y aproximaciones dialógicas de gran sofisticación [?, 8]. Sin embargo, una parte significativa de estas propuestas permanece en un plano eminentemente teórico. La formalización en lógicas deónticas, marcos argumentativos o razonamiento basado en casos ha enriquecido la comprensión de cómo se estructura la deliberación jurídica [9, 10], pero rara vez se ha demostrado su viabilidad computacional en sistemas ejecutables. En particular, persiste una brecha entre el modelo formal del razonamiento normativo y la implementación en entornos computacionales que preserven la fidelidad lógica y eviten reduccionismos indebidos.

Los sistemas multiagente (SMA) se han consolidado como una de las arquitecturas más prometedoras para modelar interacciones sociales y jurídicas [11, 12]. Su potencial para representar múltiples perspectivas, gestionar conflictos de normas y simular procesos de negociación los convierte en un enfoque idóneo para la resolución de disputas. Sin embargo, la mayor parte de las aplicaciones en el ámbito jurídico se limita a modelos abstractos o simulaciones conceptuales, sin ofrecer prototipos funcionales que articulen la teoría jurídica con la práctica computacional. Este vacío limita tanto el avance académico como el impacto práctico de la investigación.

En este contexto, el problema central que orienta la presente investigación puede formularse de la siguiente manera: ¿cómo construir un modelo computacional que integre, de manera fiel y verificable, teorías jurídicas y arquitecturas multiagente para representar la generación y confrontación de argumentos normativos en disputas de propiedad intelectual? La dificultad radica en que no se trata únicamente de representar reglas legales como condiciones lógicas, sino de capturar la dinámica de adaptación, ataque y defensa que caracteriza a la deliberación jurídica. Resolver este problema implica tender un puente entre la teoría formal —Kelsen, Kripke, la lógica de normas y las ontologías FIPA— y la implementación computacional en un lenguaje lo suficientemente expresivo y riguroso como para garantizar consistencia, como es el caso de Mercury [13, 14].

El planteamiento anterior no busca reducir la práctica jurídica a un conjunto de reglas mecánicas ni sustituir la labor humana de jueces y abogados. Por el contrario, se orienta a demostrar que es posible diseñar agentes computacionales que, al menos en un nivel de prototipo, repliquen estructuras de razonamiento normativo y produzcan argumentos cohe-

rentes con doctrinas reconocidas en la literatura jurídica. Este paso constituye un requisito previo para avanzar hacia sistemas multiagente jurídicos capaces de deliberar de manera autónoma y ofrecer soporte legítimo en disputas de alta complejidad. En suma, el problema que se enfrenta no es únicamente técnico, sino profundamente interdisciplinario: articular el lenguaje del Derecho con el lenguaje de la Computación sin sacrificar la riqueza normativa de uno ni la precisión formal de la otra.

I.2. Justificación

La presente investigación se justifica en tres dimensiones complementarias: científica, interdisciplinaria y social. En primer lugar, desde la perspectiva científica, existe una brecha evidente entre la formalización del razonamiento jurídico y su implementación computacional. La literatura en Inteligencia Artificial y Derecho (AI & Law) ha desarrollado marcos teóricos sofisticados —desde la lógica de normas hasta los modelos argumentativos de Carneades [?] y CATO [9, 15]—, pero la mayoría de estos enfoques permanecen en el nivel conceptual. La falta de prototipos verificables limita tanto la validación empírica de las propuestas como su potencial para evolucionar hacia sistemas funcionales. En este sentido, la investigación aporta al campo de las Ciencias de la Computación al tender un puente entre teoría formal y ejecución computacional, utilizando un lenguaje declarativo de tipado fuerte como Mercury [13, 14] para preservar la fidelidad lógica y evitar reduccionismos.

En segundo lugar, la relevancia interdisciplinaria es ineludible. El Derecho y la Computación son dominios con tradiciones epistemológicas distintas: el primero se centra en la interpretación normativa y en la deliberación argumentativa; el segundo, en la precisión formal y la verificabilidad. Históricamente, los intentos por articular ambos lenguajes han tropezado con la dificultad de no trivializar el razonamiento jurídico ni sobrecargar el modelo computacional con formalismos inabarcables [8, 16]. En este punto resulta importante subrayar que, aunque en la actualidad existen modelos de lenguaje de gran escala (LLMs), como los *Generative Pre-Trained*, que aparentan producir razonamientos jurídicos plausibles, su funcionamiento no responde a principios normativos verificables, sino a correlaciones estadísticas en grandes volúmenes de texto. Dichas herramientas carecen de mecanismos internos de trazabilidad que permitan garantizar la coherencia normativa de sus argumentos. En contraste, la presente investigación demuestra que es posible construir un modelo computacional donde cada argumento esté anclado en estructuras normativas explícitas (Kelsen, IHL, BDI), asegurando transparencia y auditable fidelidad. Así, mientras los LLM ofrecen utilidad práctica en la exploración textual, el aporte de este trabajo es eminentemente cien-

tífico y metodológico: mostrar que el razonamiento jurídico puede implementarse de manera formal, verificable y explicable en un agente computacional.

Finalmente, la justificación social radica en el impacto directo que este tipo de modelos puede tener en la resolución de disputas jurídicas contemporáneas. La expansión de los contenidos digitales, los desafíos de la IA generativa y la complejidad de los mercados globales hacen cada vez más frecuentes los conflictos en materia de propiedad intelectual [17, 18, 19]. Las plataformas de resolución en línea de disputas (*Online Disputes Resolution*, ODR) han mostrado avances en accesibilidad y eficiencia [20], pero carecen de mecanismos argumentativos transparentes que legitimen sus resultados. En este sentido, la presente investigación demuestra que es posible avanzar hacia sistemas de resolución asistida por agentes que no solo optimicen tiempos y costos, sino que también generen confianza mediante la producción de argumentos jurídicamente fundamentados.

En suma, la investigación se justifica porque aborda simultáneamente un vacío científico, una necesidad interdisciplinaria y un desafío social. Su contribución no se limita a la programación de un agente aislado, sino a la demostración de que es posible construir un puente sólido entre el razonamiento normativo y la implementación computacional. Este paso inicial constituye un requisito indispensable para el desarrollo de sistemas multiagente capaces de deliberar sobre disputas jurídicas reales con transparencia, rigor y legitimidad.

I.3. Objetivos

Objetivo General

Modelar computacionalmente el razonamiento jurídico en disputas de propiedad intelectual mediante un sistema multiagente BDI, capaz de reevaluar creencias y decisiones frente a nuevas consideraciones normativas, preservando la coherencia con principios teóricos y lógicos.

Objetivos Específicos

1. Formalizar un modelo basado en sistemas multiagente (BDI) para representar interacciones jurídicas.
2. Definir un mecanismo de argumentación normativa sustentado en lógica modal y teorías jurídicas estructuradas.
3. Prototipar el modelo en un lenguaje lógico-declarativo a fin de validar la simulación del razonamiento normativo.

4. Evaluar la evolución de creencias e intenciones del agente frente a nuevas circunstancias argumentativas.
5. Demostrar la aplicabilidad del modelo mediante el caso *Pierson v. Post*, como ejemplo ilustrativo de la evolución argumentativa.

I.4. Hipótesis

El razonamiento jurídico en disputas de propiedad intelectual es modelable computacionalmente mediante un sistema multiagente BDI, capaz de reevaluar creencias e intenciones frente a nuevas consideraciones normativas, preservando la coherencia lógica y garantizando la trazabilidad de los argumentos generados.

I.5. Contribuciones esperadas

Las contribuciones de esta investigación se enmarcan en tres niveles complementarios: teórico, metodológico y aplicado.

1. **Contribución teórica:** Establecer un puente formal entre teorías jurídicas clásicas (como la teoría pura del derecho de Kelsen y la lógica de normas) y arquitecturas computacionales de agentes BDI, mostrando que es posible traducir estructuras normativas en representaciones computacionales verificables.
2. **Contribución metodológica:** Proponer un mecanismo de generación de argumentos normativos en un agente BDI, sustentado en lógica modal y deóntica, que permite la reevaluación de creencias e intenciones ante nuevas condiciones normativas. Esta contribución demuestra cómo articular de manera coherente un modelo de razonamiento jurídico dentro de un marco multiagente.
3. **Contribución aplicada:** Desarrollar un prototipo en Mercury que simula el proceso de generación de argumentos jurídicos, validado mediante el caso *Pierson v. Post*. Aunque se limita a un agente individual, el prototipo constituye una prueba de concepto que evidencia la factibilidad de extender el modelo hacia escenarios multiagente más complejos.

En conjunto, estas contribuciones no solo refuerzan la discusión académica en torno a la Computación, Filosofía y el Derecho, sino que sientan bases sólidas para futuras investigaciones orientadas a sistemas multiagente normativos más amplios, transparentes y explicables.

I.6. Organización del documento

El presente documento se estructura en seis capítulos, además de apéndices y bibliografía, con la finalidad de guiar al lector desde la identificación del problema hasta la demostración de las aportaciones y limitaciones de la investigación.

En el **Capítulo 1** se introducen los elementos fundamentales de la investigación: planteamiento del problema, justificación, objetivos, hipótesis y contribuciones esperadas. Este capítulo establece el marco inicial que orienta todo el trabajo doctoral.

El **Capítulo 2** presenta el marco conceptual y el estado del arte. Se analizan los fundamentos del razonamiento jurídico en disputas de propiedad intelectual, los principios de los sistemas multiagente y los avances en plataformas de resolución en línea de disputas (ODR). Asimismo, se identifican las brechas que motivan la propuesta de este trabajo.

El **Capítulo 3** describe el modelo propuesto. Se detalla la arquitectura general del sistema multiagente, la formalización de normas jurídicas dentro de la arquitectura BDI, el protocolo de argumentación normativa y los criterios de evaluación que sustentan la coherencia del modelo.

El **Capítulo 4** integra las publicaciones realizadas durante el proceso doctoral. Se explica cómo cada una de ellas —artículo en revista, capítulo de libro y ponencia en congreso— contribuye a distintos componentes del modelo y cómo, en conjunto, conforman una aportación unificada.

El **Capítulo 5** expone la evaluación experimental y la discusión crítica. Se describe el diseño experimental, los escenarios de prueba, las métricas utilizadas y los resultados obtenidos, para después analizar sus implicaciones, limitaciones y alcances.

El **Capítulo 6** presenta las conclusiones y el trabajo futuro. Se sintetizan las aportaciones teóricas, metodológicas y aplicadas de la investigación, se discute su impacto potencial en el ámbito jurídico y computacional, y se señalan posibles líneas de investigación a desarrollar en el futuro.

Finalmente, los **apéndices** reúnen materiales de apoyo que fortalecen la validez del trabajo, incluyendo evidencias de publicaciones, detalles técnicos de implementación y escenarios de disputa empleados. La **bibliografía** consolida las referencias académicas utilizadas a lo largo del documento.

Capítulo II

Marco conceptual y estado del arte

Este capítulo expone los fundamentos conceptuales y la revisión crítica del estado del arte que enmarcan la presente investigación. Su propósito es doble: por un lado, situar los antecedentes en torno a las disputas de propiedad intelectual y las tecnologías emergentes que las complejizan; por otro, identificar los avances en sistemas multiagente, modelos de argumentación y lenguajes de programación que constituyen la base para el modelo propuesto en el Capítulo 3.

La organización del capítulo responde a una lógica progresiva. En primer lugar, se analizan las características de las disputas en materia de propiedad intelectual en el contexto digital y la relevancia de contar con modelos explicables que aseguren legitimidad en su resolución. En segundo lugar, se revisan los principios de los sistemas multiagente, con énfasis en la arquitectura BDI y en la incorporación de normas, negociación y comunicación entre agentes. Posteriormente, se estudian las plataformas de resolución en línea de disputas (ODR), destacando tanto sus avances en accesibilidad como sus limitaciones en cuanto a fundamentación normativa y transparencia.

A continuación, se presenta una revisión de los principales modelos de argumentación y razonamiento jurídico desarrollados en el campo de la Inteligencia Artificial y el Derecho, enfatizando su potencial y también las brechas que han dejado abiertas. Seguidamente, se abordan las bases lógicas y semánticas que sustentan la propuesta de este trabajo, incluyendo las contribuciones de la lógica modal, deóntica e híbrida, así como las perspectivas categoriales y topológicas. Finalmente, se discute la relevancia de los lenguajes declarativos de programación, en particular Mercury, como soporte para garantizar la fidelidad lógica y la trazabilidad en la implementación.

En conjunto, este capítulo no pretende ofrecer una recopilación exhaustiva, sino una revisión selectiva, crítica y articulada que permita identificar las brechas de conocimiento.

Estas brechas constituyen el punto de partida del modelo propuesto en el Capítulo 3, donde se plantea una integración novedosa entre teoría jurídica y sistemas computacionales basados en agentes.

II.1. Resolución de disputas en propiedad intelectual

En este trabajo se entiende por *disputa legal* aquel conflicto entre dos o más partes en torno a la interpretación de derechos u obligaciones jurídicas, cuya resolución requiere mecanismos institucionales como la negociación, el arbitraje o el litigio [21]. A diferencia de un desacuerdo cotidiano, la disputa legal implica un proceso argumentativo estructurado, donde cada parte debe presentar razones que respalden su postura y refuten las del contrario. Este matiz es crucial en la propiedad intelectual (PI), donde las interpretaciones sobre derechos de autor, patentes o marcas suelen apoyarse en principios normativos y doctrinales que no siempre convergen.

La propiedad intelectual constituye uno de los ámbitos jurídicos más dinámicos y conflictivos en la era digital. La expansión de contenidos en línea, la proliferación de obras generadas mediante inteligencia artificial y la creciente complejidad de los mercados globales han intensificado los conflictos relacionados con la creación, reproducción y explotación de obras [17, 18]. A diferencia de otras ramas jurídicas, la PI enfrenta el reto de proteger simultáneamente la creatividad individual, la innovación tecnológica y el acceso colectivo al conocimiento, lo cual genera tensiones permanentes entre intereses privados y bienes públicos.

Este panorama se ve agravado por la velocidad del cambio tecnológico. Mientras que los marcos normativos requieren procesos deliberativos largos, las tecnologías digitales se transforman de manera exponencial [2]. Ello produce un desfase que se traduce en incertidumbre normativa: por un lado, existe la necesidad de proteger derechos adquiridos; por otro, se demanda flexibilidad para incorporar nuevas formas de creación y explotación. La consecuencia práctica es que las disputas en PI rara vez admiten soluciones únicas o definitivas, sino que abren un espacio de argumentación donde confluyen posturas divergentes.

La literatura jurídica ha mostrado desde hace siglos la naturaleza compleja y argumentativa de la propiedad intelectual. Un ejemplo clásico es el caso *Pierson v. Post* (1805), en el cual se debatió si la persecución de un animal silvestre constituía suficiente fundamento para reclamar su propiedad [3]. La Corte produjo opiniones enfrentadas: una formalista, que exigía la captura efectiva, y otra más pragmática, que reconocía la intención y el esfuerzo de la caza. Este precedente, aunque alejado en el tiempo, refleja un problema que sigue vigente:

la propiedad intelectual no se agota en la aplicación mecánica de normas, sino que implica ponderar principios, valores y contextos.

En disputas más recientes, los enfoques de distancia legal y de negociación internacional evidencian que las diferencias en sistemas normativos y percepciones culturales influyen decisivamente en la resolución de conflictos [22]. Asimismo, herramientas de teoría de juegos han sido empleadas para modelar negociaciones en materia de patentes, mostrando cómo las estrategias de los actores pueden condicionar el desenlace más allá de la literalidad de la norma [23]. Estos estudios refuerzan la idea de que los conflictos en PI son esencialmente dinámicos y requieren mecanismos que integren tanto reglas formales como procesos adaptativos de argumentación.

Paralelamente, el desarrollo de soluciones tecnológicas en el ámbito legal ha dado lugar a un ecosistema de herramientas conocidas como *LegalTech*. Plataformas como Kira Systems [4] o Luminance [5] aplican técnicas de aprendizaje automático para la revisión masiva de contratos, mientras que sistemas como Casetext [6], CaseMine [7] o Lex Machina [24] utilizan algoritmos de análisis predictivo y minería de jurisprudencia para apoyar la toma de decisiones. Aunque estas aplicaciones han demostrado ser útiles en términos de eficiencia y reducción de costos, carecen de un componente argumentativo normativo: su valor reside en procesar grandes volúmenes de datos, no en justificar por qué una solución es jurídicamente válida o legítima.

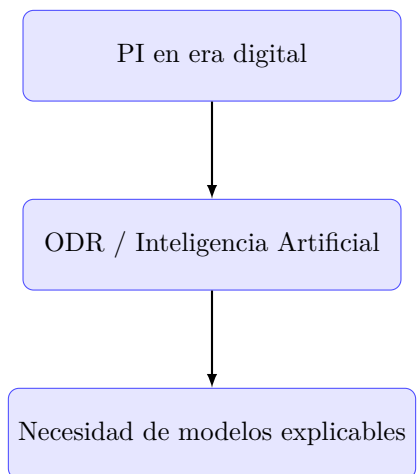


Figura II-1: Síntesis conceptual de la evolución en disputas de PI: de la era digital hacia la necesidad de modelos explicables.

De este modo, el panorama actual de la propiedad intelectual combina tres elementos: disputas normativas complejas, un entorno tecnológico acelerado y la presencia de herra-

mientas algorítmicas que, si bien ofrecen ventajas operativas, no satisfacen la necesidad de transparencia y fundamentación jurídica. Esta evolución puede representarse de manera sintética en la Figura II-1, donde se muestra el tránsito desde la problemática de la PI en la era digital, pasando por las soluciones basadas en ODR e inteligencia artificial, hasta la necesidad actual de desarrollar modelos computacionales explicables. Este reconocimiento plantea la pregunta de qué arquitecturas computacionales permiten representar de manera rigurosa dichos procesos argumentativos. En el siguiente apartado II.2 se abordan los sistemas multiagente (SMA) y, en particular, la arquitectura BDI, como un marco idóneo para modelar la negociación y la dinámica de normas en contextos jurídicos.

II.2. Sistemas Multiagente (SMA) y negociación

La necesidad de modelos computacionales explicables en disputas jurídicas, identificada al final de la sección anterior, exige explorar arquitecturas que permitan representar no solo reglas estáticas, sino también procesos dinámicos de deliberación. En este sentido, los sistemas multiagente (SMA) constituyen un marco particularmente fértil, ya que posibilitan modelar interacciones entre entidades autónomas que razonan, negocian y coordinan sus acciones en entornos complejos. A diferencia de aproximaciones centradas en reglas rígidas, los SMA ofrecen la capacidad de capturar tanto el comportamiento individual de un agente como las dinámicas colectivas que emergen de la interacción.

El concepto de *agente* ha sido definido como una entidad computacional autónoma, capaz de percibir su entorno, razonar sobre sus objetivos y actuar en consecuencia [25]. Entre las diversas arquitecturas de agentes, el paradigma Belief–Desire–Intention (BDI) ha demostrado ser uno de los más robustos para modelar procesos de toma de decisiones racionales [26, 27, 11]. Un agente BDI mantiene representaciones explícitas de sus creencias, deseos e intenciones, lo que permite no solo planificar acciones, sino también reevaluarlas cuando cambian las circunstancias [28]. Esta capacidad resulta crucial en contextos jurídicos, donde las normas y argumentos pueden modificarse durante el curso de una disputa.

Ahora bien, la investigación en SMA ha demostrado que los agentes no se limitan a ejecutar planes individuales [16], sino que pueden incorporar normas que regulan sus conductas y las de otros agentes. El campo de los *norm-aware agents* ha estudiado cómo integrar restricciones normativas en arquitecturas BDI [29], de modo que los agentes no solo persigan metas, sino que también respeten reglas institucionales [30] y adapten su comportamiento en función de ellas [31, 32]. Este enfoque ha dado lugar al desarrollo de instituciones electrónicas (*e-institutions*) [33], donde los agentes interactúan bajo protocolos normativos explícitos

[34], lo que aproxima significativamente estas arquitecturas a escenarios jurídicos reales [35].

Más allá del cumplimiento de normas, las disputas legales requieren mecanismos de negociación y coordinación entre agentes con intereses divergentes [36]. La literatura en SMA ha abordado este reto desde múltiples perspectivas: la negociación estratégica y la cooperación en entornos lógicos [37, 38], la síntesis de leyes sociales útiles para sociedades artificiales [39], el estudio de normas sociales en dinámicas de grupo [40], así como la modelación de la cooperación y confianza [41, 42]. Estos trabajos demuestran que la incorporación de negociación y normas sociales en SMA constituye un paso necesario para simular la complejidad de las interacciones jurídicas, donde los actores no solo aplican reglas, sino que las interpretan, discuten y eventualmente modifican.

La comunicación entre agentes representa otro componente fundamental. Para que los SMA sean viables en entornos jurídicos, los agentes deben intercambiar información bajo lenguajes comunes y semánticamente consistentes. En este terreno, los lenguajes de comunicación de agentes (ACL) y los estándares propuestos por la FIPA han proporcionado protocolos formales que aseguran que los mensajes intercambiados sean comprensibles y verificables [43]. El uso de ontologías compartidas refuerza esta coherencia, permitiendo que los agentes razonen no solo con base en símbolos [44], sino también en significados alineados con marcos normativos [45].

En este contexto, resulta pertinente diferenciar tres aproximaciones que han marcado la evolución de la inteligencia artificial aplicada al Derecho: los sistemas expertos, los agentes BDI y los SMA normativos. La comparación no pretende establecer una jerarquía, sino mostrar cómo cada enfoque ha ampliado las posibilidades de representación y deliberación. Mientras que los sistemas expertos fueron pioneros en la automatización jurídica, los agentes BDI introdujeron la noción de estados mentales explícitos, y los SMA normativos extendieron esta capacidad hacia entornos multiagente con reglas e instituciones compartidas. La Tabla II-1 sintetiza estas diferencias de manera sistemática, contrastando criterios como el estado mental, la adaptación al entorno, la coordinación, el manejo de normas y la trazabilidad.

Tabla II-1: Comparación conceptual entre sistema experto, agente BDI y SMA normativo

Criterio	Sistema experto	Agente BDI	SMA normativo
Representación del estado mental	No aplica (basado en reglas fijas)	Explícita (creencias, deseos, intenciones)	Explícita y compartida entre agentes
Adaptación al entorno	Estática	Reevaluación dinámica de BDI	Reevaluación dinámica bajo normas institucionales
Diálogo y coordinación	Limitado o inexistente	Interacción básica entre agentes	Interacción estructurada con ACL y normas sociales
Manejo de normas	Implícito en reglas	Posible como restricciones adicionales	Central: instituciones electrónicas y gestión explícita de normas
Trazabilidad y explicabilidad	Limitada	Alta (por el modelo BDI)	Alta y colectiva (deliberación multiagente)

Como se observa en la Tabla II-1, la evolución de sistemas expertos a SMA normativos refleja un cambio cualitativo en la capacidad de representar razonamiento jurídico: de reglas estáticas hacia procesos dinámicos y colectivos, donde los agentes no solo actúan, sino que deliberan bajo marcos normativos explícitos. Este avance constituye el fundamento para considerar a los SMA como un marco idóneo en la modelación de disputas legales complejas. No obstante, aunque prometedores, estos desarrollos aún no han sido ampliamente implementados en plataformas reales de resolución de disputas. Esta brecha conecta con la siguiente sección, dedicada a las plataformas de resolución en línea de disputas (ODR), las cuales han priorizado accesibilidad y eficiencia, pero todavía carecen de fundamentación normativa y transparencia argumentativa.

II.3. Plataformas de resolución en línea de disputas (ODR)

En la sección anterior se destacó que los sistemas multiagente normativos ofrecen un marco robusto para modelar disputas jurídicas, aunque todavía se encuentran en una fase principalmente conceptual. Frente a esta brecha entre teoría y práctica, las plataformas de resolución en línea de disputas (*Online Dispute Resolution*, ODR) han emergido como una alternativa aplicada, con presencia creciente en ámbitos comerciales e institucionales. Su objetivo central ha sido reducir costos, mejorar la accesibilidad y acelerar la resolución de conflictos mediante el uso de tecnologías digitales [20]. De este modo, mientras los SMA representan una promesa académica, las plataformas ODR constituyen hoy el esfuerzo más visible por trasladar la resolución de disputas al plano tecnológico.

Los modelos de ODR han incorporado herramientas para la mediación en línea, mecanismos de negociación asistida y sistemas de arbitraje electrónico. Algunas plataformas permiten a las partes interactuar de manera asincrónica, facilitando el intercambio de propuestas y contraofertas; otras ofrecen algoritmos que sugieren soluciones de compromiso con base en parámetros previamente definidos. Estos avances han mejorado notablemente la accesibilidad de la justicia, al reducir la necesidad de presencia física y ofrecer procesos más rápidos y menos costosos [20]. En consecuencia, el campo ODR ha demostrado que la tecnología puede desempeñar un papel relevante en la gestión eficiente de conflictos.

No obstante, la crítica recurrente a estas plataformas es que, si bien resuelven problemas de eficiencia y accesibilidad, no abordan la cuestión fundamental de la legitimidad normativa. Diversos estudios han señalado que las soluciones ODR tienden a funcionar como “cajas negras” tecnológicas: producen resultados útiles, pero carecen de mecanismos internos de explicabilidad que permitan justificar sus decisiones en términos jurídicos [19]. Ello implica que las partes pueden obtener un acuerdo, pero no siempre comprender por qué esa solución es normativamente correcta o qué principios la respaldan. Además, existe el riesgo de que la automatización se perciba como una amenaza a derechos fundamentales, como el derecho a una audiencia justa y a la motivación explícita de las resoluciones [46, 2].

Para clarificar este contraste, resulta útil sintetizar las capacidades actuales de las plataformas ODR frente a sus limitaciones en términos normativos. La Tabla II-2 resume esta diferencia, destacando lo que dichas herramientas ya logran hacer de manera efectiva y lo que aún queda pendiente desde la perspectiva de la fundamentación jurídica.

Tabla II-2: Capacidades actuales de ODR frente a sus limitaciones normativas

Qué hace ODR hoy	Qué no hace ODR
Facilita negociación en línea mediante interfaces digitales	No garantiza fundamentación normativa explícita en las soluciones
Reduce costos y tiempos procesales	Carece de mecanismos de explicabilidad jurídica transparentes
Amplía la accesibilidad a usuarios que no podrían acudir presencialmente	No asegura legitimidad percibida ni protección plena de derechos procesales
Incorpora algoritmos para sugerir acuerdos de compromiso	No proporciona trazabilidad normativa de las recomendaciones

Como se aprecia en la Tabla II-2, las plataformas ODR resuelven el problema de la eficiencia, pero no el de la legitimidad. Esta distinción es crucial para la presente investigación: si bien la accesibilidad y rapidez son valores importantes, el Derecho exige además mecanismos de justificación que garanticen transparencia y trazabilidad en la producción de

argumentos. En este punto se hace necesario acudir a los modelos de argumentación desarrollados en el campo de la Inteligencia Artificial y el Derecho (AI & Law), que han trabajado precisamente en formalizar cómo se generan, confrontan y evalúan los argumentos jurídicos. La siguiente sección aborda estos modelos, que constituyen la base teórica indispensable para avanzar de plataformas meramente funcionales hacia sistemas explicables y normativamente fundamentados.

II.4. Modelos de argumentación y razonamiento jurídico (AI & Law)

Como se señaló al analizar las plataformas ODR, los avances prácticos en resolución de disputas han priorizado la accesibilidad y la eficiencia, pero han descuidado la fundamentación normativa y la transparencia explicativa. En contraste, el campo de la Inteligencia Artificial y el Derecho (AI & Law) ha concentrado buena parte de sus esfuerzos precisamente en el estudio de cómo se generan, estructuran y evalúan los argumentos jurídicos. A lo largo de varias décadas, esta disciplina ha desarrollado modelos formales y computacionales que, aunque en su mayoría permanecen en el plano teórico, ofrecen las bases necesarias para construir sistemas explicables y normativamente consistentes.

Uno de los desarrollos más influyentes corresponde a los modelos de argumentación formal y dialógica. Entre ellos, el marco de Carneades ha demostrado cómo estructurar argumentos a partir de premisas, excepciones y estándares de prueba, asignando dinámicamente la carga de la prueba en función del avance del diálogo [?]. Estas propuestas han sido ampliadas desde la perspectiva argumentativa por Prakken y Sartor, quienes destacan que el razonamiento jurídico es inherentemente dialógico y depende de la interacción de partes que atacan y defienden posturas bajo reglas explícitas de procedimiento [8]. La crítica de Blair a las esquemas de Walton también resalta que la argumentación jurídica no puede reducirse a patrones fijos, sino que requiere flexibilidad para capturar contextos cambiantes [47]. En conjunto, estas aproximaciones dialógicas subrayan que la validez de un argumento no reside solo en su estructura lógica, sino en su capacidad de resistir ataques dentro de un marco normativo.

Otro eje de investigación ha sido el razonamiento con casos y valores. Trabajos como los de Bench-Capon y Sartor han mostrado que los precedentes pueden entenderse como teorías en construcción, donde cada caso no solo aplica reglas, sino que incorpora valores sociales que justifican la decisión [9]. Atkinson y Bench-Capon profundizaron esta línea al presentar el razonamiento jurídico como una forma de razonamiento práctico, en el que la

deliberación se apoya en argumentos normativos y valorativos [10]. Por su parte, Aleven desarrolló modelos de razonamiento con casos aplicados a entornos educativos, demostrando que este enfoque puede generar predicciones de resultados coherentes con la jurisprudencia [15]. De manera complementaria, Ashley y Rissland plantearon la necesidad de modelos híbridos que integren el razonamiento basado en casos con enfoques argumentativos formales [48, 49]. Estas propuestas destacan que el Derecho no solo opera sobre reglas, sino que está anclado en valores que orientan la interpretación y aplicación de las normas.

El razonamiento derrotable o *defeasible reasoning* constituye otra de las bases teóricas fundamentales. El trabajo de Simari y Loui introdujo un marco matemático en el que los argumentos no son necesariamente concluyentes, sino susceptibles de ser derrotados por otros más específicos o más fuertes [50]. Esta perspectiva ha tenido gran impacto en AI & Law, pues refleja con precisión cómo los jueces y abogados construyen y revisan argumentos que pueden ser aceptados o rechazados dependiendo de la evolución del debate. El razonamiento derrotable se convirtió así en un pilar para explicar la naturaleza abierta y no monotónica del razonamiento jurídico.

En años más recientes, la investigación también ha explorado cómo dotar a los modelos de argumentación de capacidades de explicabilidad aplicadas a casos concretos. Collenette, Atkinson y Bench-Capon presentaron herramientas de *explainable AI* para el Tribunal Europeo de Derechos Humanos, mostrando que es posible diseñar sistemas capaces de reproducir decisiones judiciales con altos niveles de precisión y, al mismo tiempo, ofrecer explicaciones comprensibles para los usuarios [51]. Este tipo de proyectos evidencian que la línea de trabajo en AI & Law no se limita a la teoría, sino que puede transformar la práctica jurídica si logra articular modelos de argumentación con entornos de aplicación real.

Para sintetizar las distintas escuelas revisadas, la Figura II-2 presenta un mapa conceptual de los principales enfoques de argumentación en AI & Law: razonamiento con casos y valores, esquemas dialógicos, razonamiento derrotable y modelos explicables recientes. El diagrama ilustra no solo sus diferencias, sino también los elementos que esta investigación retoma para construir un modelo normativo basado en agentes.

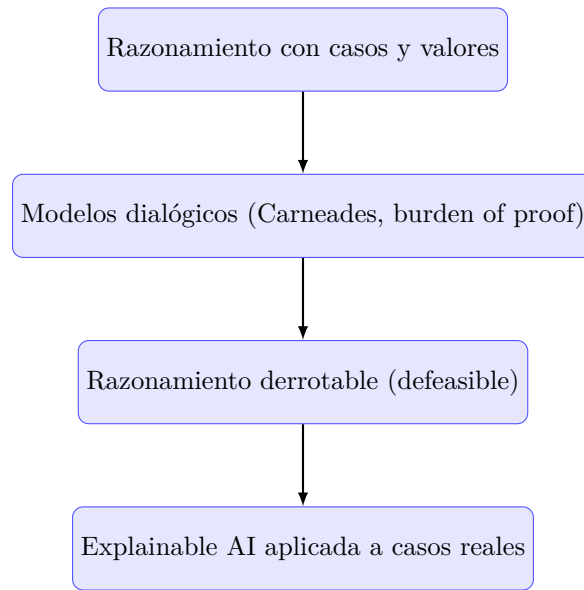


Figura II-2: Principales enfoques de argumentación en AI & Law y su relación con la presente investigación.

En conjunto, los modelos revisados muestran que el campo de AI & Law ha desarrollado herramientas conceptuales y formales para representar la argumentación jurídica en toda su complejidad. Sin embargo, la mayoría de estas aproximaciones han permanecido en el nivel abstracto, sin traducirse en implementaciones computacionales verificables. Precisamente aquí radica el aporte de esta investigación: aprovechar los avances teóricos de la argumentación para construir un modelo formal que pueda ser implementado en un lenguaje declarativo como Mercury, asegurando trazabilidad lógica y fidelidad normativa. Para ello, resulta necesario examinar los fundamentos semánticos que permiten articular las estructuras lógicas con el modelo computacional, aspecto que se desarrolla en la siguiente sección.

II.5. Bases lógicas y semánticas para el modelo propuesto

La revisión de los modelos de argumentación en AI & Law mostró que existen múltiples aproximaciones para representar el razonamiento jurídico. Sin embargo, su aplicación en un sistema computacional requiere un fundamento semántico que asegure coherencia lógica y explicabilidad. En otras palabras, no basta con reproducir esquemas de ataque y defensa de argumentos: es necesario definir el marco formal que determine qué significa que un argumento sea válido, cómo se relacionan las normas y de qué manera se representan las estructuras normativas en un agente. Esta sección expone las bases lógicas y semánticas que

orientan el modelo propuesto.

Un primer eje proviene de la normatividad y de la teoría jurídica. La tradición kelseniana subraya que el Derecho puede concebirse como un sistema jerárquico de normas, donde cada disposición deriva su validez de otra superior hasta llegar a una norma fundamental [52]. Este principio, conocido como *validez derivada*, puede expresarse formalmente de la siguiente manera:

$$\forall N_i \in \mathcal{N}, \exists (N_0 \vdash N_1 \vdash \dots \vdash N_i)$$

$$\text{o equivalentemente: } \forall N_i \in \mathcal{N}, \exists N_j (j < i) : N_j \vdash N_i$$

donde N_0 corresponde a la *Grundnorm* o norma fundamental. Esta formulación no solo establece un criterio de validez, sino también un mecanismo de resolución de conflictos: las normas de nivel superior prevalecen sobre las inferiores. En el modelo propuesto, esta jerarquía se traducirá en un orden de preferencias normativas que guiará la selección de argumentos.

El segundo eje corresponde a la lógica intuicionista híbrida (IHL), que combina operadores de la lógica intuicionista con mecanismos de referencia a mundos concretos [53]. Esta lógica resulta particularmente útil para modelar la condicionalidad contextual de las normas, es decir, su aplicación solo en determinados escenarios. Formalmente, se define de la siguiente manera:

$$@_w(\varphi \rightarrow \psi) \Rightarrow (@_w\varphi \Rightarrow @_w\psi) \quad (\text{implicación constructiva local})$$

$$@_w\Diamond\varphi \equiv \exists v(w R v \wedge @_v\varphi) \quad (\text{posibilidad contextual})$$

De este modo, una norma puede activarse únicamente en aquellos mundos donde se cumplen las condiciones que la habilitan. En el modelo computacional, esta idea se traducirá en reglas condicionales que modifican las creencias del agente según el contexto.

Un tercer eje lo constituyen las ontologías de la *Foundation for Intelligent Physical Agents* (FIPA), que proveen un estándar para la comunicación semántica entre agentes [43]. Una ontología FIPA puede formalizarse como un cuádruple:

$$\mathcal{O} = \langle C, P, I, A \rangle$$

donde C es el conjunto de clases o conceptos (como **Agente**, **Mensaje**, **Acción**), P el con-

junto de propiedades, I el conjunto de instancias y A los axiomas que definen restricciones lógicas. La ventaja de este marco es que garantiza que todos los agentes compartan un vocabulario común y que la deliberación multiagente se produzca en un espacio semánticamente coherente.

Finalmente, la arquitectura BDI, que constituye el núcleo de los agentes propuestos, puede representarse formalmente como una tupla:

$$\mathcal{A} = \langle B, D, I \rangle$$

donde B son las creencias, D los deseos y I las intenciones [27]. Esta formulación no es meramente descriptiva, sino que permite formalizar el ciclo deliberativo del agente: actualizar creencias según nueva información normativa, generar deseos coherentes con esos cambios y, finalmente, comprometerse con intenciones que se traduzcan en argumentos jurídicamente válidos.

En síntesis, los formalismos revisados en esta sección (Kelsen, IHL, FIPA y BDI) constituyen los pilares que aseguran la coherencia lógica y la trazabilidad normativa del modelo propuesto. En el Capítulo 3 estos fundamentos serán retomados no como teoría aislada, sino como componentes que se integran explícitamente en la formalización computacional del agente.

II.6. Lenguajes y plataformas para implementación

La solidez de las bases lógicas y semánticas revisadas en la sección anterior exige que la implementación computacional se realice en un entorno que garantice fidelidad normativa. Esto implica que el lenguaje de programación no puede ser seleccionado únicamente por conveniencia técnica, sino que debe preservar propiedades como el tipado fuerte, el control del determinismo, la trazabilidad de las inferencias y la capacidad de manejar razonamiento declarativo de manera eficiente. En este contexto, la elección de un lenguaje no constituye un detalle accesorio, sino un componente fundamental para asegurar que el modelo conserve coherencia y explicabilidad al pasar del plano teórico al plano computacional.

Entre las opciones disponibles, los lenguajes de programación lógica ofrecen ventajas significativas. El lenguaje analizado en este trabajo se distingue por su arquitectura declarativa basada en ejecución eficiente, su sistema de tipos estáticos, y su modelo de modos y determinismo que permiten verificar en tiempo de compilación gran parte de las propiedades del programa [13, 14, 54, 55]. Estas características refuerzan la confiabilidad del sistema, al minimizar ambigüedades y garantizar que cada cláusula programada respete las condiciones

lógicas definidas por el modelo normativo.

El análisis comparativo con otros lenguajes declarativos y funcionales refuerza este punto. Prolog, en sus distintas variantes, ha sido históricamente el lenguaje más extendido en programación lógica, pero su tipado dinámico y su gestión implícita del control dificultan la trazabilidad de inferencias complejas [56, 57]. Logtalk y λ Prolog han ampliado el paradigma hacia enfoques orientados a objetos y al razonamiento sobre abstracciones, pero con costos adicionales de complejidad [58, 59]. Curry y Haskell ofrecen integración con programación funcional y expresividad matemática, pero carecen de mecanismos internos de determinismo explícito aplicables al razonamiento normativo [60, 61]. Erlang y Scala, aunque potentes en concurrencia, fueron diseñados para sistemas distribuidos industriales y no para modelar de manera directa lógicas normativas [62, 63]. En comparación, el lenguaje adoptado en esta investigación proporciona un equilibrio entre expresividad lógica y verificación estática, lo que resulta decisivo para representar normas de manera transparente y auditable.

Asimismo, el vínculo histórico entre programación lógica y la planificación en inteligencia artificial resulta relevante. La formulación de problemas de planificación como satisfacibilidad [64], la conexión con modelos estables en dominios de planificación [65], y los trabajos clásicos de Wilkins en representación y generación de planes [66] sentaron las bases para lenguajes como Answer Set Programming (ASP), que se consolidaron como herramientas para resolver problemas no monotónicos [67, 68]. El lenguaje empleado en este trabajo hereda esa tradición, al combinar eficiencia de ejecución con mecanismos declarativos que permiten mantener consistencia lógica, lo que lo convierte en un candidato natural para implementar modelos normativos en entornos jurídicos.

La relación entre los requisitos de fidelidad normativa y las propiedades del lenguaje puede sintetizarse en la Tabla II-3. Este cuadro ilustra cómo las exigencias teóricas del modelo se traducen en capacidades tecnológicas verificables en la práctica, reforzando que la elección del lenguaje no es arbitraria, sino un elemento central de la investigación.

Tabla II-3: Correspondencia entre requisitos normativos y propiedades del lenguaje

Requisito de fidelidad normativa	Propiedad del lenguaje
Necesidad de consistencia normativa jerárquica	Tipado fuerte que detecta inconsistencias en tiempo de compilación
Explicabilidad de las decisiones	Modos y determinismo que documentan el flujo de inferencias
Trazabilidad de argumentos jurídicos	Representación declarativa que preserva el vínculo entre premisas y conclusiones
Adaptación a cambios normativos	Mecanismos de backtracking y actualización de estados controlados
Eficiencia en entornos complejos	Ejecución optimizada mediante compilación a código eficiente

En conjunto, el análisis muestra que la plataforma seleccionada permite trasladar de manera fiel los fundamentos normativos a un entorno computacional verificable, garantizando transparencia y consistencia en la simulación del razonamiento jurídico. Sin embargo, a pesar de estas ventajas, el estado actual de la literatura revela que pocas investigaciones han dado el paso de integrar de forma plena la teoría normativa con implementaciones ejecutables. Identificar estas brechas constituye el siguiente paso, lo que permitirá precisar el aporte diferencial de la presente investigación.

II.7. Brechas identificadas en la literatura

La revisión del estado del arte permite reconocer avances sustantivos en la intersección entre Derecho y Computación, pero también revela brechas significativas que motivan esta investigación. Identificar estos vacíos no es un ejercicio accesorio, sino un paso esencial para establecer el aporte original del modelo propuesto en el Capítulo 3. A continuación, se sintetizan las principales carencias encontradas en la literatura.

En primer lugar, los sistemas multiagente han sido objeto de estudios exhaustivos, tanto desde el punto de vista de los lenguajes de programación como de su aplicación conceptual. Revisiones recientes, como la de Cardoso y Ferrando, muestran la riqueza del ecosistema de lenguajes MAS y sus aplicaciones potenciales [69, 70]. Sin embargo, la mayoría de estos trabajos se concentra en comparaciones teóricas o en estudios de factibilidad, sin llegar a implementar prototipos ejecutables que preserven la fidelidad normativa y la trazabilidad lógica requeridas en un contexto jurídico.

En segundo lugar, el campo de la argumentación jurídica en AI & Law ha producido modelos de gran sofisticación, como Carneades [?] y los marcos de prueba dinámica [8], así

como teorías de razonamiento con casos y valores [9]. No obstante, estas aproximaciones permanecen en gran medida en el nivel formal y abstracto: existe una abundancia de modelos conceptuales, pero una carencia de implementaciones verificables que demuestren su aplicabilidad en entornos computacionales reales.

En tercer lugar, las plataformas ODR y las soluciones de LegalTech han demostrado avances prácticos en términos de accesibilidad, eficiencia y gestión documental [20]. Sin embargo, han sido ampliamente criticadas por su falta de fundamentación normativa explícita, al operar como “cajas negras” que generan resultados útiles pero no explicables en términos jurídicos [19]. Esta opacidad limita su legitimidad en disputas de alta complejidad, donde no basta con obtener acuerdos rápidos, sino que se requiere justificar la corrección normativa de las soluciones.

Un cuarto vacío se relaciona con la irrupción de los modelos de lenguaje de gran escala (LLMs). Aunque han mostrado un potencial notable para tareas de recuperación y generación de texto, su funcionamiento estadístico carece de anclaje normativo verificable, lo que los convierte en herramientas útiles pero no confiables para fundamentar decisiones jurídicas [71]. El contraste con experiencias recientes en *explainable AI* aplicadas a tribunales reales, como el caso del Tribunal Europeo de Derechos Humanos [51], refuerza la necesidad de modelos que combinen transparencia, fidelidad lógica y aplicabilidad práctica.

En conclusión, la literatura revisada ha generado aportaciones valiosas en cada uno de los dominios analizados, pero carece de una integración que articule teoría jurídica, modelos de argumentación y plataformas computacionales ejecutables con fidelidad normativa y explicabilidad. Precisamente este vacío constituye el espacio de innovación de la presente investigación, que propone un modelo formal basado en agentes normativos, sustentado en fundamentos lógicos y jurídicos, e implementado en un lenguaje declarativo que preserve la trazabilidad. El siguiente capítulo desarrolla este modelo en detalle, describiendo su arquitectura, formalización y criterios de evaluación.

Capítulo III

Modelo propuesto

El capítulo anterior estableció las bases conceptuales y el estado del arte en torno a la resolución de disputas de propiedad intelectual, los sistemas multiagente normativos y los lenguajes formales que permiten su representación computacional [8]. Allí se identificaron brechas fundamentales: la ausencia de prototipos verificables que preserven la fidelidad normativa, la limitada integración entre teorías jurídicas clásicas y arquitecturas computacionales, y la falta de mecanismos explícitos de explicabilidad en las plataformas de resolución en línea de disputas [20]. Estas carencias señalaron la necesidad de un modelo capaz de tender un puente sólido entre el razonamiento jurídico normativo y su validación en un entorno computacional [19, 71].

En este contexto, el presente capítulo expone el modelo propuesto, cuyo objetivo es demostrar que es posible articular un agente computacional con base en la arquitectura BDI, sustentado en teorías jurídicas reconocidas, y validado mediante un lenguaje declarativo de tipado fuerte [11]. El modelo no debe entenderse como un sistema experto [12], sino como un diseño multiagente en el que cada instancia refleja de manera explícita los principios teóricos que guían su razonamiento. La implementación de un agente individual (Tompkins) constituye una prueba de concepto mínima, pero suficiente para mostrar la coherencia del diseño, su capacidad de adaptación normativa y su potencial de escalamiento hacia interacciones multiagente [72].

El capítulo se organiza en cuatro apartados. En la Sección 3.1 se describe la arquitectura general del sistema, mostrando cómo se estructura un agente BDI y cómo esta arquitectura está preparada para incorporar múltiples instancias con diferentes posiciones doctrinales (por ejemplo, Tompkins y Livingston en el caso *Pierson v. Post*). La Sección 3.2 presenta la formalización de las normas jurídicas dentro del modelo, estableciendo la correspondencia entre teorías como la jerarquía normativa de Kelsen, la semántica de Kripke o las ontolo-

gías FIPA, y su representación computacional. En la Sección 3.3 se detalla el protocolo de negociación y mediación normativa, explicando cómo el ciclo BDI soporta la generación y confrontación de argumentos. Finalmente, la Sección 3.4 discute las propiedades y criterios de evaluación del modelo, con especial énfasis en la coherencia normativa, la adaptabilidad y la capacidad de representar interpretaciones doctrinales múltiples.

En suma, este capítulo constituye el núcleo de la investigación: aquí se plasma el tránsito desde los fundamentos teóricos revisados en el Capítulo 2 hacia un modelo computacional verificable, demostrando que la teoría jurídica puede implementarse en un agente explicable y trazable. Con ello se sientan las bases para la evaluación experimental que se presentará en el Capítulo 5.

III.1. Arquitectura general del sistema multiagente

La propuesta se fundamenta en la arquitectura BDI (*Beliefs, Desires, Intentions*), ampliamente utilizada para modelar agentes autónomos con estados mentales explícitos y un ciclo de deliberación racional [26, 27, 11, 28]. En el dominio jurídico, BDI es idónea porque separa con nitidez (i) qué cree el agente sobre el caso y las normas (creencias), (ii) qué fines normativos pretende alcanzar (deseos) y (iii) a qué cursos de acción argumentativa se compromete tras deliberar (intenciones). Esta distinción permite superar los límites de los sistemas expertos puramente reactivos, favoreciendo una *reevaluación* de decisiones ante nueva información o ataques argumentativos, rasgo propio del razonamiento jurídico dialógico [8].

Formalismo BDI. Un agente se modela como una tupla canónica:

$$\mathcal{A} = \langle B, D, I \rangle,$$

donde B es el conjunto de creencias, D el conjunto de deseos, e I el conjunto de intenciones vigentes. El paso de D a I se formaliza mediante un operador auxiliar de *construcción de argumentos*, denotado por $\Gamma(B, D)$, que selecciona los compromisos viables dados los hechos y condiciones normativas activas. Así, la **selección de intenciones** puede expresarse como:

$$I = \{ a \in \Gamma(B, d) \mid d \in D, B \models \text{cond}(d) \wedge \neg \text{bloqueado}(d, B) \}. \quad (\text{III-1})$$

La dinámica ante ataques argumentativos se refleja como una actualización de creencias $B \xrightarrow{\alpha} B'$, lo cual dispara un nuevo ciclo de selección:

$$I' = \Gamma(B', D). \quad (\text{III-2})$$

De esta manera, el modelo captura el ciclo BDI clásico: *actualizar creencias* $\rightarrow$ *filtrar deseos habilitados por el contexto* $\rightarrow$ *adoptar intenciones como argumentos comprometidos* [26, 27, 28].

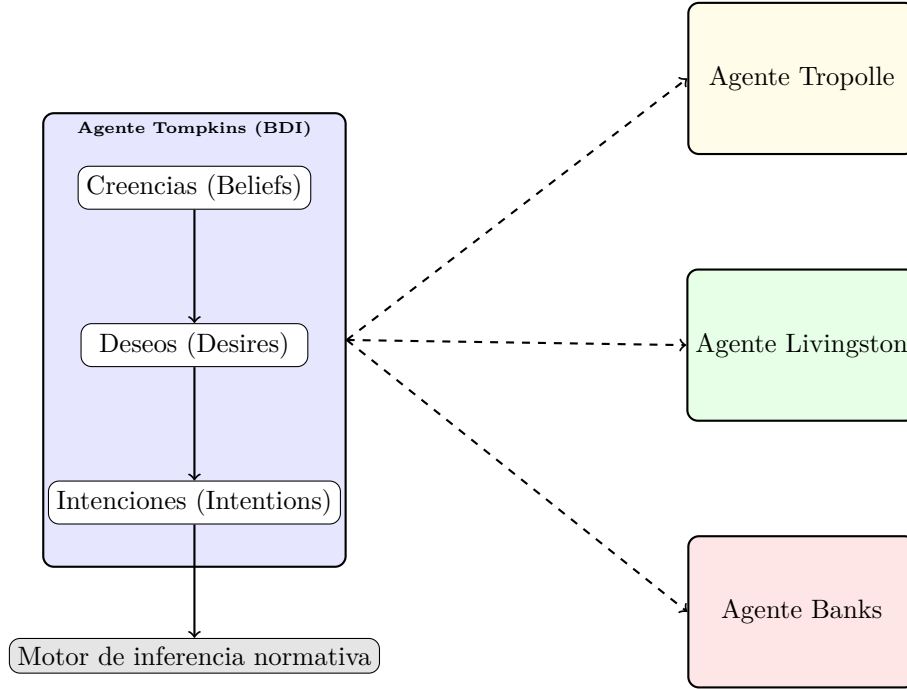


Figura III-1: Arquitectura general del sistema multiagente: agente Tompkins implementado y proyección hacia otros agentes con marcos normativos distintos.

Correspondencia BDI $\leftrightarrow$ Mercury. La Figura III-1 mostró el esqueleto funcional. En el prototipo implementado en Mercury (Listado VI.1), cada componente BDI se refleja directamente en un tipo o construcción:

Tabla III-1: Correspondencia entre componentes BDI y su implementación en Mercury

Componente BDI	Implementación en Mercury
Creencias (B)	Tipo <code>belief</code> sobre <code>proposition</code> : <code>belief(f1)</code> , <code>unknown(f5)</code> , etc.
Deseos (D)	Tipo <code>desire(name, value, condition)</code> : cada deseo combina fin, valor y condición normativa.
Intenciones (I)	Tipo <code>argument</code> : <code>argument(beliefs, action, goals, values)</code> . Las acciones representan compromisos normativos (<code>find_ownership_established</code> , etc.).
Operador $\Gamma(B, D)$	Función <code>add_arguments_to_Tomkins</code> , que filtra y construye argumentos válidos a partir de creencias y deseos.
Actualización $B \rightarrow B'$	Predicado <code>apply_attack/3</code> , que modifica el estado del agente y fuerza re-selección de intenciones.

Por ejemplo, la inicialización del agente en Mercury refleja directamente la tupla $\langle B, D, I = \emptyset \rangle$:

```

1  Tomkins = agent(
2  [belief(f1), belief(f2), belief(f3), belief(f4), unknown(f5),
3  belief(f6), unknown(f7)],
4  [desire(clear_law, less_litigation, ownership_plaintiff),
5  desire(unclear_law, less_litigation, ownership_no_possession),
6  desire(trade_restricted, economic_benefit,
7  malicious_intent_productive_activity_defendant)],
8  [],
9  ),

```

Este fragmento muestra cómo las proposiciones se fijan como *creencias*, los fines normativos como *deseos*, y las intenciones se inicializan como vacías. La posterior ejecución de `add_arguments_to_Tomkins` corresponde a la selección definida en (III-1), mientras que `apply_attack/3` implementa la actualización descrita en (III-2).

Alcance multiagente. Aunque el prototipo implementa un solo agente (Tompkins), la arquitectura conserva su naturaleza multiagente. Otros agentes como Livingston o Banks se definen con la misma tupla $\langle B, D, I \rangle$, pero con creencias y deseos iniciales distintos, lo que refleja diferentes posiciones doctrinales. El diagrama de la Figura III-1 ilustra esta proyección. En la práctica, esto significa que la arquitectura propuesta no es un sistema experto cerrado, sino un modelo preparado para la interacción y negociación multiagente.

En síntesis, la Sección 3.1 establece el andamiaje conceptual y computacional del modelo: un agente BDI cuya dinámica se refleja fielmente en el prototipo Mercury. Sobre esta base,

la Sección 3.2 detallará la formalización de normas jurídicas que alimentan las condiciones normativas y guían la construcción de argumentos.

III.2. Formalización de las normas jurídicas en el modelo

La arquitectura BDI descrita en la Sección 3.1 proporciona el esqueleto deliberativo del agente. Falta precisar *qué* estructura normativa alimenta sus estados mentales y *cómo* se traduce esa estructura en decisiones argumentativas. Esta sección formaliza cuatro pilares teóricos —jerarquía kelseniana, condicionalidad contextual (IHL), semántica de Kripke y estándares FIPA— y muestra su *operacionalización* directa en el prototipo Mercury (Listado VI.1). El objetivo es asegurar *fidelidad normativa* y *trazabilidad* entre teoría, modelo y ejecución.

(i) Jerarquía normativa de Kelsen: validez derivada y precedencia. La teoría pura del Derecho concibe el ordenamiento como una jerarquía de validez: cada norma N_i obtiene su validez de otra superior N_j ($j < i$), remontando hasta una *norma fundamental* N_0 [52]. Esta idea puede capturarse como una relación de derivación $\vdash$ (validez derivada):

$$\forall N_i \in \mathcal{N} \ \exists N_j \ (j < i) : N_j \vdash N_i, \quad (\text{III-3})$$

que induce un *orden de precedencia* $\triangleright$ en caso de conflicto:

$$N_a \triangleright N_b \text{ si y sólo si } \text{rank}(N_a) < \text{rank}(N_b). \quad (\text{III-4})$$

En el agente. El orden $\triangleright$ se implementa como *condiciones normativas priorizadas*, materializadas en el tipo `condition` del Listado VI.1 (p. ej., `ownership_plaintiff`, `no_ownership_defendant`, ...). El mecanismo de selección de intenciones descrito en la Sección 3.1 respeta esta precedencia: si dos deseos habilitan cursos de acción incompatibles, prevalece el sustentado en la norma de mayor rango.

(ii) Condicionalidad contextual: activación normativa (IHL y enfoques normativos). En marcos normativos contemporáneos, la aplicabilidad de una norma depende de *condiciones* de caso, tiempo, jurisdicción o estatus de las partes [73]. Expresamos esta idea con una relación de satisfacción contextual:

$$B \models \text{cond}(N) \Rightarrow \text{“la norma } N \text{ se activa bajo el contexto de creencias } B\text{”}. \quad (\text{III-5})$$

En el agente. Las condiciones normativas se codifican en el universo `condition` y se

evalúan contra B (lista de **belief/unknown** sobre **proposition**). Cuando $B \models \text{cond}(d)$ es verdadero, el deseo `desire(..., condition)` que la invoca queda *habilitado* para generar intenciones (ecuación (III-1)); si la condición no se satisface, el deseo permanece inactivo. Este diseño impone trazabilidad: toda intención adoptada puede rastrearse a (a) las creencias que activaron la condición y (b) la norma (o cláusula) que justifica el curso de acción.

(iii) Semántica de mundos posibles (Kripke): cambio de creencias y reevaluación.

Para capturar que los argumentos pueden cambiar de estatus al variar las premisas, recurrimos a semánticas modales/híbridas basadas en Kripke [74, 75, 53]. Sea $@_w\varphi$ “ φ es válida en el mundo w ”; la posibilidad contextual se expresa como:

$$@_w\Diamond\varphi \equiv \exists v (wRv \wedge @_v\varphi), \quad (\text{III-6})$$

donde R conecta mundos compatibles (p.ej., tras incorporar un nuevo hecho o excepción). Al recibir un *ataque* que cuestiona una creencia o activa una excepción, el agente transita $w \rightarrow v$ y debe *reevaluar* sus intenciones.

En el agente. El predicado `apply_attack/3` actualiza B (equivalente a pasar de w a v en (III-6)); luego `add_arguments_to_Tomkins/1` recomputa intenciones I' según (III-2). Así, la semántica de Kripke se operacionaliza como una *transición de estado* de creencias seguida por re-selección BDI.

(iv) Estándares FIPA: interoperabilidad y vocabulario compartido. En escenarios multiagente, la comunicación exige *ACL* y ontologías controladas [44, 45, 43]. Un esquema típico de ontología FIPA puede modelarse como un cuádruple $\mathcal{O} = \langle C, P, I, A \rangle$ (clases, propiedades, instancias y axiomas), que garantiza coherencia semántica en el intercambio.

En el agente. Aunque la validación de esta tesis ejecuta una sola instancia (Tomkins), el *diseño* incorpora la noción de vocabulario controlado (clases **proposition**, **value**, **action**, **condition**) y relaciones entre ellas, de modo que la proyección multiagente (Figura III-1) pueda *compartir* significados y comparar argumentos bajo las mismas etiquetas y restricciones.

De la teoría a la implementación: tabla de trazabilidad. Para explicitar el puente, la Tabla III-2 muestra ejemplos representativos de cómo cada pilar teórico se refleja en el prototipo.

Tabla III-2: De la formalización normativa a las construcciones Mercury (extractos del Listado VI.1)

Pilar formal	Implementación y efecto
Kelsen: orden de precedencia (III-4)	<code>condition</code> priorizadas; el filtrado <code>should_add_argument/1</code> evita intenciones basadas en condiciones vencidas por normas de mayor rango.
Condicionalidad contextual (III-5)	<code>belief/unknown + condition</code> ; si $B \models \text{cond}(d)$, el <code>desire</code> que la invoca puede generar <code>argument</code> en <code>add_arguments_to_Tomkins/1</code> .
Kripke: transición $w \rightarrow v$ (III-6)	<code>apply_attack/3</code> actualiza <code>beliefs</code> ($B \rightarrow B'$) y dispara recomputación de intenciones (ecuación (III-2)).
FIPA/ontología compartida	Universo tipado: <code>proposition</code> , <code>value</code> , <code>action</code> , <code>condition</code> actúan como vocabulario controlado para futuras interacciones.

Evidencias puntuales en el Tabla VI.1.

- **Universo proposicional y creencias tipadas.** `proposition` (f1–f7) fija el alfabeto fáctico; `belief/disbelief/unknown` impone estados epistémicos explícitos (chequeados por el compilador).
- **Condiciones normativas como interfaz Kelsen/IHL.** `condition` enumera activadores normativos (propiedad, malicia, bienestar público, etc.); su evaluación concreta refleja $B \models \text{cond}(d)$.
- **Selección intencional trazable.** `add_arguments_to_Tomkins/1` usa `list.filter` con `should_add_argument/1`: si la condición de un deseo no es satisfecha por B (o queda vencida por precedencia), el argumento no pasa al conjunto de intenciones I .
- **Dinámica ante ataque.** `apply_attack/3` produce `TomkinsUpdatedAttack`; volver a llamar `add_arguments_to_Tomkins/1` materializa (III-2).

En conjunto, la formalización normativa asegura que cada intención adoptada por el agente pueda rastrearse hasta una secuencia clara de fundamentos: desde las creencias que configuran el contexto, pasando por la condición que habilita el deseo, hasta la acción normativa que se convierte en intención. Esta trazabilidad evita reduccionismos y otorga coherencia

al proceso, de modo que el prototipo ejecutado en Mercury no representa un código aislado, sino la expresión computacional de una semántica normativa previamente establecida. Con esta base, el modelo está preparado no solo para generar argumentos consistentes, sino también para confrontarlos y someterlos a procesos de ataque y defensa. Precisamente esta dimensión dinámica —la negociación y la mediación normativa— constituye el siguiente paso, abordado en la Sección 3.3.

III.3. Protocolo de negociación y mediación normativa

La formalización expuesta en la Sección 3.2 dota al agente de una base normativa trazable. Sin embargo, la deliberación jurídica no se agota en representar normas: exige un *protocolo dinámico* que permita construir, atacar y defender argumentos conforme las creencias e intenciones del agente se actualizan. Este apartado desarrolla dicho protocolo, mostrando cómo el ciclo BDI se convierte en un mecanismo de negociación y mediación normativa.

El ciclo BDI en clave jurídica. El funcionamiento del agente se articula en un ciclo deliberativo que traduce el paradigma BDI al contexto normativo. Este ciclo sigue cuatro fases:

1. **Percepción.** El agente recibe información sobre el caso —hechos, proposiciones fácticas o activación de normas— que actualiza su conjunto de creencias B .
2. **Deliberación.** Con base en B , evalúa qué deseos D se encuentran habilitados por condiciones normativas (cf. (III-5)), descartando aquellos bloqueados por jerarquías superiores (cf. (III-4)).
3. **Decisión.** El operador de construcción $\Gamma(B, D)$ selecciona los argumentos coherentes y los convierte en intenciones I (cf. (III-1)), priorizando aquellos consistentes con valores normativos.
4. **Acción.** Cada intención se materializa como una acción argumentativa (`find_ownership_established`, `discourage_fox_hunting`, etc.), que puede ser comunicada o confrontada en un escenario multiagente.

Este protocolo no es un flujo rígido, sino un proceso iterativo: cualquier cambio en las creencias provoca un nuevo ciclo, lo que refleja la naturaleza dinámica de la deliberación jurídica. En el prototipo Mercury, este comportamiento se implementa mediante la secuencia `add_arguments_to_Tomkins/1` $\rightarrow$ `apply_attack/3` $\rightarrow$ `add_arguments_to_Tomkins/1`, lo

cual garantiza que el agente no quede anclado en una decisión inicial, sino que reevalúe constantemente sus intenciones.

Construcción de argumentos normativos. Un argumento jurídico en el modelo se define como una combinación de creencias, acción normativa, metas y valores:

$$a = \langle B, \text{action}, \text{goals}, \text{values} \rangle.$$

En la implementación, este esquema se refleja en el tipo `argument(beliefs, action, goals, values)`. Este diseño responde a dos criterios: (i) *trazabilidad*, pues cada argumento puede desglosarse en los elementos normativos que lo habilitan; y (ii) *explicabilidad*, ya que la estructura hace visible cómo los valores (como `less_litigation` o `public_benefit`) orientan la selección de intenciones. En términos prácticos, el agente no genera “respuestas mecánicas”, sino que produce razonamientos anclados en principios normativos y valorativos.

Ataque y defensa de argumentos. La deliberación jurídica implica confrontación. En el modelo, un *ataque* cuestiona las creencias de base de un argumento, activa excepciones o introduce condiciones incompatibles. Esto se formaliza de manera clásica en la teoría de la argumentación [50, 8]:

$$a_1 \rightarrow a_2 \quad \Leftrightarrow \quad \text{Conclusión}(a_1) \text{ contradice una premisa de } a_2.$$

Un ataque induce una actualización $B \rightarrow B'$ en las creencias del agente; el conjunto de intenciones se recalcula entonces como:

$$I' = \Gamma(B', D).$$

De este modo, la reevaluación no depende de reglas ad hoc, sino de la semántica ya establecida: los ataques modifican el estado epistémico del agente y desencadenan un nuevo ciclo deliberativo que puede reforzar, sustituir o descartar intenciones previas.

Reflejo en el prototipo Mercury. En la implementación, esta dinámica se hace explícita en dos pasos del Listado VI.1:

```

1      % Aplica un ataque y reevalúa las creencias del agente.
2      apply_attack(TomkinsUpdated, Attack1, TomkinsUpdatedAttack),
3
4      % Recalcula intenciones después de la actualización.
```

```
TomkinsReevaluated = add_arguments_to_Tomkins(
    TomkinsUpdatedAttack),
```

Aquí, `apply_attack/3` modela la transición $B \rightarrow B'$, mientras que `add_arguments_to_Tomkins/1` ejecuta $I' = \Gamma(B', D)$. Así, lo que en el formalismo se describe como un cambio de mundo posible (cf. (III-6)) se materializa en una recomputación de argumentos que preserva trazabilidad: el compilador asegura que cada nueva intención derive de premisas tipadas y condiciones verificables.

Diferencia con un sistema experto clásico. Un sistema experto tradicional aplica reglas predefinidas de manera estática: dadas ciertas condiciones, devuelve una conclusión, pero no reevalúa sus inferencias cuando el entorno cambia. En contraste, el agente BDI propuesto exhibe tres rasgos distintivos:

- **Estados mentales explícitos.** Las creencias, deseos e intenciones se mantienen diferenciados y son actualizables en tiempo de ejecución.
- **Reevaluación dinámica.** Cada ataque o nueva información dispara un ciclo completo de deliberación, modificando intenciones previas.
- **Dimensión argumentativa.** Las acciones no son simples salidas, sino argumentos con justificación normativa y valorativa trazable.

En este sentido, el protocolo de negociación y mediación normativa trasciende la lógica de un sistema experto: no sólo “aplica reglas”, sino que modela la dinámica propia del razonamiento jurídico, donde los argumentos se construyen, se disputan y, eventualmente, se modifican.

Sobre esta base, la Sección 3.4 examinará las propiedades y criterios de evaluación del modelo, analizando en qué medida satisface los requisitos de coherencia normativa, adaptabilidad y capacidad de generar interpretaciones doctrinales múltiples.

III.4. Propiedades y criterios de evaluación

El modelo propuesto no pretende ser únicamente una demostración de viabilidad computacional, sino un marco con propiedades bien definidas que permiten valorar su coherencia y su aporte a la investigación en Inteligencia Artificial y Derecho. En esta sección se precisan los criterios bajo los cuales el modelo será evaluado, de manera que el lector cuente con una referencia clara de las dimensiones que guiarán el análisis posterior en el Capítulo 5.

Coherencia normativa. El primer criterio se refiere a la capacidad del modelo para mantener consistencia interna entre las normas jerárquicas, las condiciones contextuales y los argumentos generados. Una simulación carecería de valor si permitiera conclusiones contradictorias o intenciones derivadas de normas inferiores que desconocen la precedencia de normas superiores. La coherencia normativa implica, por tanto, que cada argumento pueda rastrearse hasta una fuente normativa válida y que, en caso de conflicto, prevalezca siempre la norma de mayor jerarquía. Este criterio se evalúa examinando los ciclos deliberativos del agente y verificando que nunca se generen intenciones incompatibles con la estructura kelseniana de referencia.

Adaptabilidad. El segundo criterio apunta a la capacidad del agente para reevaluar sus intenciones frente a cambios en el entorno argumentativo. En el razonamiento jurídico, las conclusiones rara vez son definitivas: nuevos hechos, excepciones o ataques argumentativos pueden alterar el equilibrio normativo. El modelo se somete a prueba determinando si, tras una actualización de creencias, el agente produce un nuevo conjunto de intenciones que refleje la modificación del contexto. La adaptabilidad se traduce en un proceso deliberativo iterativo, donde las decisiones no permanecen fijas, sino que se ajustan en función de la evolución del debate.

Pluralidad doctrinal. El tercer criterio considera la capacidad del modelo para representar interpretaciones jurídicas diversas. El Derecho no se caracteriza por ofrecer una única respuesta correcta, sino por albergar posturas doctrinales alternativas, cada una con fundamentos normativos y axiológicos propios. El diseño multiagente en potencia del modelo —representado por instancias como Tompkins, Livingston o Banks— busca capturar precisamente esa pluralidad. La evaluación se centra en comprobar que, aun con una misma estructura BDI, la variación de creencias y deseos iniciales conduce a conjuntos de argumentos distintos, reflejando enfoques formalistas, pragmáticos o utilitaristas. Este criterio permite demostrar que el modelo no es rígido ni monolítico, sino flexible frente a la diversidad de tradiciones jurídicas.

Estos tres criterios —coherencia normativa, adaptabilidad y pluralidad doctrinal— constituyen el marco de referencia para valorar el modelo. No se trata únicamente de verificar que el código en Mercury compila o produce resultados, sino de demostrar que la implementación preserva las propiedades que distinguen al razonamiento jurídico de cualquier otro dominio: consistencia, dinamismo y pluralidad interpretativa. En el Capítulo 5, estos criterios se retomarán como ejes de la evaluación experimental, donde se pondrá a prueba el

comportamiento del agente en escenarios concretos y se analizará hasta qué punto cumple con las exigencias planteadas aquí.

Con esta delimitación de propiedades y criterios, concluye la presentación del modelo propuesto. El siguiente capítulo (Capítulo 4) mostrará cómo las publicaciones generadas a lo largo del proceso doctoral se integran y complementan para dar solidez teórica, metodológica y aplicada a la propuesta.

Capítulo IV

Productos obtenidos

El presente capítulo reúne y contextualiza las publicaciones que se generaron a lo largo del proceso doctoral, las cuales constituyen tanto los hitos de validación académica del trabajo como una línea evolutiva del propio modelo propuesto. Cada una de ellas refleja una etapa de madurez distinta en la investigación: desde la identificación del problema y su relevancia interdisciplinaria, hasta la consolidación de un modelo formal validado en un entorno computacional ejecutable.

En primer lugar, la publicación en la revista *Komputer Sapiens*, editada por la Sociedad Mexicana de Inteligencia Artificial y registrada en el Índice de Revistas Mexicanas de Divulgación Científica y Tecnológica de CONACyT (ahora SECIHTI), representó el punto de partida del proyecto. En este trabajo se planteó por primera vez la problemática de las disputas de propiedad intelectual en el contexto de la inteligencia artificial y se exploró cómo los sistemas multiagente (SMA) podían ofrecer una base metodológica para modelar los procesos de deliberación jurídica. Este artículo permitió enmarcar el problema y formular las primeras hipótesis sobre la aplicabilidad de arquitecturas cognitivas en entornos normativos.

Posteriormente, el capítulo titulado “*Towards a Formal Language in Law*”, publicado por IGI Global en el Handbook of Research on Applied Artificial Intelligence and Robotics for Government Processes (2023, DOI 10.4018/978-1-6684-5624-8.ch004), marcó una transición hacia un plano más teórico y filosófico. Esta obra —indexada en Scopus y Web of Science— permitió examinar la necesidad de un lenguaje formal especializado para el razonamiento jurídico, sustentado en la Filosofía del Derecho y en la teoría pura de Kelsen. Allí se discutió la posibilidad de estructurar un lenguaje normativo propio, capaz de integrar valores, jerarquías y condiciones de validez dentro de un sistema multiagente. Si bien el alcance de dicho lenguaje resultó ambicioso, el trabajo sentó las bases conceptuales del modelo que más tarde se implementaría de manera computacional.

Finalmente, el artículo “*Modeling Legal Judgements with BDI Agents*”, publicado en las Lecture Notes in Networks and Systems (LNNS, vol. 1118) de Springer, como parte de los Proceedings del Computer Science Online Conference 2024 (CSOC 2024), consolidó la validación técnica del modelo. Esta serie, indexada en Scopus y Web of Science (CPCI), permitió demostrar empíricamente que el razonamiento jurídico puede modelarse computacionalmente mediante un sistema multiagente BDI, incluso utilizando un lenguaje lógico-declarativo de propósito general como *Mercury* [55]. En esta etapa se comprobó la coherencia interna del modelo, su capacidad de reevaluación normativa y su potencial para escalar hacia entornos multiagente más complejos.

En conjunto, estas tres publicaciones trazan el itinerario completo de la investigación: desde la identificación del problema y su pertinencia interdisciplinaria, pasando por la formalización teórica de un lenguaje jurídico-computacional, hasta la demostración práctica de su viabilidad técnica. Cada una de ellas refuerza una dimensión distinta del trabajo doctoral —científica, teórica y metodológica— y, al integrarlas, se obtiene una visión coherente del proceso de construcción del modelo de razonamiento jurídico basado en agentes BDI.

IV.1. Inclusión de la inteligencia artificial en disputas legales sobre propiedad intelectual | *Komputer Sapiens*

La primera publicación derivada de este trabajo, titulada “*Inclusión de la inteligencia artificial en disputas legales sobre propiedad intelectual*” [76], marcó el inicio del proyecto doctoral. En ese momento, el objetivo no era aún construir un modelo formal completo, sino entender con claridad qué estaba ocurriendo en la frontera entre el Derecho y la inteligencia artificial, y por qué las herramientas tecnológicas existentes no lograban representar adecuadamente la complejidad del razonamiento jurídico.

El artículo surgió de una preocupación muy concreta: la forma en que las disputas de propiedad intelectual estaban siendo transformadas por la automatización. Los sistemas de análisis contractual y las plataformas de búsqueda jurídica impulsadas por IA —como Kira Systems, Luminance o Casetext— comenzaban a reemplazar tareas tradicionalmente humanas [4, 5, 6, 7]. Sin embargo, lo que ofrecían era básicamente una mayor eficiencia en la clasificación de información, no una comprensión del razonamiento normativo que subyace a una disputa. En otras palabras, podían procesar grandes volúmenes de texto, pero no “razonar jurídicamente”.

Esa brecha entre el potencial técnico y el sentido normativo fue el punto de partida del texto. La publicación planteó que, para hablar verdaderamente de “inteligencia jurídica”,

era necesario dotar a los sistemas de la capacidad de argumentar, justificar y reevaluar decisiones de acuerdo con principios jurídicos explícitos. Fue allí donde comencé a explorar la idea de los sistemas multiagente (SMA) como un posible camino para representar este tipo de procesos, inspirándome en la noción de agentes BDI (*Beliefs, Desires, Intentions*) propuesta por Wooldridge y Jennings [11, 12].

La propuesta consistía en imaginar a cada agente como una “postura jurídica” dentro de una disputa: un juez formalista, un juez pragmático, o un actor con intereses económicos específicos. En conjunto, los agentes formarían un sistema capaz de deliberar, negociar y defender argumentos en torno a un mismo conflicto. De alguna manera, era una traducción computacional de la estructura misma del debate jurídico.

Sin embargo, mientras el trabajo avanzaba, se hizo evidente que el modelo multiagente, por sí solo, no resolvía la cuestión de fondo. Los SMA podían reproducir comportamientos estratégicos o coordinados, pero carecían de un lenguaje capaz de expresar con fidelidad las estructuras normativas del Derecho. Un agente podía decidir o coordinar, pero no explicar, ni mucho menos justificar jurídicamente, la validez de su decisión. Esa constatación fue clave: comprender que no bastaba con un SMA —que hacía falta dotar a los agentes de una semántica formal que les permitiera razonar como entes jurídicos, no solo como entidades racionales.

En retrospectiva, esa fue quizá la aportación más importante de este primer trabajo. El artículo no ofrecía todavía un modelo completo, pero sí estableció el “por qué” de toda la tesis: la necesidad de construir un marco donde la inteligencia artificial no solo ejecutara reglas, sino que pudiera dialogar con el lenguaje y la lógica del Derecho. En ese sentido, la publicación en *Komputer Sapiens* sirvió como una especie de cartografía inicial: identificó las limitaciones de los enfoques existentes, puso sobre la mesa la idea de los sistemas multiagente, y abrió la pregunta sobre cómo hacerlos normativamente coherentes.

A partir de esa reflexión se configuró el paso siguiente: si el problema no era solo de arquitectura computacional, sino también de lenguaje, ¿sería posible imaginar un lenguaje formal pensado desde el Derecho mismo? Esa pregunta dio origen al segundo trabajo publicado en IGI Global, donde la investigación avanzó hacia la búsqueda de un lenguaje jurídico-computacional que sirviera como fundamento formal para el modelo.

IV.2. Towards a Formal Language in Law | *IGI Global*

La segunda publicación derivada del proyecto doctoral, titulada “*Towards a Formal Language in Law*” [77], representó un punto de inflexión en la investigación. Si el primer trabajo había

planteado que los sistemas multiagente podían servir para representar disputas jurídicas de forma dinámica, este segundo texto fue el intento de ir más allá: de pensar un lenguaje que hiciera posible expresar las estructuras normativas del Derecho con la misma precisión y coherencia que un sistema formal en lógica.

En esta etapa, el proyecto ya no se limitaba a imaginar agentes que deliberaran; buscaba dotarlos de un lenguaje propio, uno que no dependiera de metáforas computacionales importadas, sino que emergiera desde la filosofía del Derecho misma. La pregunta que guiaba el capítulo era sencilla, pero profunda: ¿es posible formalizar el razonamiento jurídico sin despojarlo de su naturaleza normativa? ¿Puede un lenguaje de programación, basado en la lógica y la teoría de agentes, representar el deber ser jurídico sin reducirlo a meras condiciones booleanas?

El trabajo publicado por IGI Global exploró precisamente esa tensión. Propuso el esbozo de un lenguaje formal —al que denominamos provisionalmente “K-Language”— construido sobre tres pilares conceptuales: la jerarquía normativa de Kelsen [52, 73], la semántica de mundos posibles de Kripke [74, 75] y la estructura categórica de la teoría de los *topoi* [78, 79]. La intuición era que solo desde un marco filosófico y lógico unificado podría alcanzarse la fidelidad que el Derecho exige en su formalización.

La publicación proponía, por tanto, un diálogo entre dos tradiciones históricamente separadas: la del pensamiento jurídico normativo y la de la ingeniería de lenguajes en inteligencia artificial. La primera aportaba la noción de validez, de jerarquía y de interpretación contextual; la segunda, las herramientas de formalización y de ejecución simbólica. En conjunto, configuraban la idea de un lenguaje en el que los conceptos jurídicos pudieran tener representación formal, relaciones lógicas y, al mismo tiempo, una semántica capaz de expresar su rango y su aplicabilidad contextual.

Durante el desarrollo del capítulo, el análisis se centró en cómo los lenguajes de programación existentes para agentes BDI —como *ALIAS*, *IMPACT* o *METAMEM*— [38, 36, 12] habían abordado parcialmente el problema de la semántica, pero sin resolverlo del todo. La mayoría carecía de un tratamiento explícito de la normatividad y no distinguía entre creencias de hecho y creencias de deber, lo que en el contexto jurídico resulta fundamental. Por ello, el texto argumentó que un lenguaje genuinamente jurídico debía incorporar no solo operadores modales ($\Box$, $\Diamond$), sino también un nivel de interpretación capaz de distinguir entre lo que es, lo que debe ser y lo que está permitido ser —una idea profundamente deontica y filosóficamente kelseniana.

No obstante, al avanzar en la propuesta teórica, el trabajo reveló la magnitud del desafío. Traducir estructuras tan abstractas como las categorías de la teoría de los *topoi* o las jerar-

quías normativas de Kelsen en un lenguaje ejecutable requería un esfuerzo interdisciplinario que excedía el horizonte temporal del proyecto doctoral. Pero más allá de la complejidad técnica, el obstáculo principal era conceptual: un lenguaje así no podía diseñarse simplemente “para” el Derecho, sino “desde” el Derecho. Es decir, debía surgir de su estructura lógica interna, no de una adaptación informática.

Reconocer esa dificultad no significó un retroceso, sino un avance metodológico. En lugar de insistir en la construcción de un nuevo lenguaje desde cero —una tarea que habría requerido años de desarrollo teórico y validación formal—, la investigación adoptó una posición pragmática: aprovechar un lenguaje ya existente, pero con las propiedades lógicas necesarias para poner a prueba el modelo propuesto. Ese punto de inflexión condujo a elegir **Mercury**, un lenguaje declarativo de tipado fuerte y determinismo controlado, que ofrecía una base lo suficientemente rigurosa para validar la estructura BDI y la coherencia normativa del agente.

La publicación en IGI Global, por tanto, marcó la madurez conceptual del proyecto. Fue el momento en que el enfoque dejó de ser exploratorio para volverse arquitectónico: ya no se trataba solo de especular sobre la posibilidad de representar el razonamiento jurídico, sino de definir las condiciones formales y semánticas bajo las cuales esa representación sería posible.

El capítulo concluía con una reflexión que sigue siendo central en esta tesis: que la formalización del Derecho no es solo una cuestión de traducción técnica, sino una de fidelidad epistemológica. Formalizar no implica reducir, sino hacer visible la estructura racional del sistema jurídico. Esa convicción es la que dio origen al siguiente paso de la investigación, desarrollado en la publicación presentada en la conferencia CSOC 2024, donde se demostró que, aun sin un lenguaje creado ex profeso, era posible implementar y verificar el modelo mediante un lenguaje declarativo de propósito general.

IV.3. Modelling Legal Judgements with BDI Agents in Mercury: Analysis of the Pierson Versus Post Case | *Lecture Notes in Networks and Systems*

La tercera publicación derivada de esta investigación, presentada en la *Computer Science Online Conference 2024 (CSOC 2024)* y publicada por Springer en la serie *Lecture Notes in Networks and Systems* [?], representa el cierre natural del itinerario intelectual iniciado en los dos trabajos anteriores. Si el artículo en *Komputer Sapiens* permitió reconocer la pertinencia de los sistemas multiagente y el de IGI Global llevó esa reflexión hacia la necesidad de un lenguaje jurídico-formal, esta contribución consolidó el modelo propuesto en un entorno verificable, mostrando que las ideas podían materializarse sin perder coherencia teórica ni

rigor normativo.

En esta etapa, la investigación había alcanzado un punto de equilibrio entre la abstracción conceptual y la necesidad de validación formal. La pregunta ya no era si los agentes BDI podían representar razonamiento jurídico, sino si era posible demostrarlo de manera operativa sin comprometer los fundamentos filosóficos que lo sustentan. Para ello, se recurrió a **Mercury**, un lenguaje declarativo de tipado fuerte, determinismo controlado y semántica pura [13, 14]. Estas características ofrecían un terreno propicio para trasladar los principios de coherencia normativa, trazabilidad y explicabilidad que habían guiado el modelo desde su concepción.

El trabajo presentado en CSOC no partió de la lógica de construir un sistema funcional, sino de la necesidad de *verificar la estructura deliberativa* que el modelo postulaba. El agente Tompkins —inspirado en el caso *Pierson v. Post* (1805) [3]— fue concebido como una instancia representativa dentro de un sistema multiagente potencial, en la cual las creencias, deseos e intenciones (BDI) se articularon con base en la jerarquía normativa de Kelsen [52], la semántica de Kripke [74, 75], y los estándares de interoperabilidad FIPA [43]. El código en Mercury permitió mostrar, con precisión formal, cómo esas teorías se reflejan en procesos de argumentación y reevaluación de normas.

Durante la validación, el agente fue sometido a escenarios de disputa en los que debía evaluar proposiciones contradictorias, actualizar sus creencias y modificar sus intenciones en consecuencia. Cada actualización —una reevaluación de creencias o la aparición de un ataque argumentativo— desencadenaba un nuevo ciclo deliberativo, preservando la coherencia del razonamiento y la consistencia jerárquica de las normas aplicadas. Este proceso, expresado en funciones puras y reglas declarativas, permitió observar el comportamiento de un razonamiento jurídico *explicable* en tiempo lógico, donde cada argumento emergía de una secuencia verificable de condiciones y premisas.

Más que ofrecer un programa ejecutable, la publicación en CSOC 2024 mostró la posibilidad de representar en términos computacionales la dinámica de un razonamiento jurídico fundado en teoría normativa. Cada componente del agente —las creencias, los deseos, las condiciones normativas y los argumentos derivados— funcionó como una correspondencia directa con las categorías teóricas revisadas en los capítulos anteriores. La ejecución en Mercury no fue un fin en sí misma, sino una manera de poner a prueba la solidez del modelo: una demostración de que la teoría podía sostenerse incluso en el terreno de la lógica formal ejecutable.

En este sentido, la contribución no se limita a una validación técnica, sino que constituye un argumento en sí misma: que la inteligencia artificial puede dialogar con el Derecho desde

un plano normativo y filosófico, y no solo instrumental. Lo que se logró fue evidenciar que las estructuras deliberativas —las que Kelsen, Kripke y la teoría de agentes habían anticipado desde distintos frentes— podían coexistir en un mismo marco formal sin perder su carácter jurídico.

Finalmente, la publicación concluyó con una reflexión que sirvió también como cierre conceptual para esta fase de la tesis: que el Derecho, entendido como un sistema de razones normativas y deliberaciones racionales, puede encontrar en la lógica declarativa un espejo fiel de su propia estructura. El agente Tompkins no fue un “prototipo” en el sentido técnico del término, sino la encarnación de una tesis: que el razonamiento jurídico es susceptible de ser modelado, explicado y sometido a validación formal, siempre que el lenguaje —computacional o jurídico— respete las condiciones de validez que el propio Derecho impone.

Con esta tercera publicación, el proyecto doctoral alcanzó su madurez teórica y experimental. Los tres trabajos, en conjunto, constituyen una trayectoria coherente: del reconocimiento del problema a la construcción de un marco teórico, y de ahí a la demostración formal de su viabilidad. El siguiente capítulo retoma esa línea, pero ya desde una perspectiva evaluativa, analizando la coherencia, trazabilidad y explicabilidad del modelo propuesto frente a los criterios establecidos en la literatura especializada.

Capítulo V

Evaluación experimental y discusión

Los capítulos anteriores trazaron el itinerario conceptual y formal del modelo: desde la identificación del problema y los fundamentos teóricos hasta su materialización en un entorno lógico-declarativo. El presente capítulo aborda su evaluación, entendida no como una validación empírica en el sentido tradicional de la ingeniería de software, sino como una *evaluación epistemológica y estructural* del modelo, en tanto artefacto conceptual capaz de representar coherentemente el razonamiento jurídico en contextos normativos.

El propósito de esta fase no es resolver una disputa jurídica completa ni reproducir la totalidad del proceso deliberativo de un tribunal. Por el contrario, busca demostrar que es posible generar, justificar y reevaluar argumentos jurídicos coherentes a partir de un conjunto explícito de creencias, deseos e intenciones; es decir, que un agente BDI puede actuar conforme a un marco normativo formal, preservando trazabilidad y explicabilidad sin recurrir a heurísticas estadísticas ni a razonamientos opacos.

Esta decisión metodológica distingue con claridad entre dos planos: resolver una controversia —que implica negociación, interpretación y ponderación de valores— y formalizar el razonamiento individual de un agente normativo, que constituye el nivel elemental de toda deliberación jurídica. La construcción de un sistema multiagente, donde distintas posturas doctrinales se enfrenten, se plantea como una extensión natural de este trabajo, pero no como su propósito inmediato.

La evaluación que aquí se presenta se desarrolla en tres niveles complementarios:

1. **Conceptual:** el caso *Pierson v. Post* se adopta como entorno de prueba, al condensar tensiones entre formalismo y pragmatismo que permiten observar el funcionamiento

del razonamiento normativo.

2. **Estructural:** se examina la correspondencia entre los principios teóricos del modelo —arquitectura BDI, jerarquía kelseniana, semántica de Kripke y estándares FIPA— y su manifestación en el código Mercury.
3. **Dinámico:** se analiza el comportamiento del agente Tomkins frente a modificaciones en sus creencias, observando cómo se adaptan sus intenciones y argumentos bajo condiciones cambiantes.

En este marco, el código Mercury no constituye el objeto de evaluación, sino la evidencia experimental que permite verificar la coherencia del modelo teórico. La validez del ejercicio radica en que cada argumento generado por el agente puede rastrearse hasta su fundamento normativo y que el sistema responde racionalmente a los cambios de contexto.

La elección de limitar el experimento a un solo agente —Tomkins— obedece a una lógica de validación progresiva: antes de modelar interacciones multiagente, era necesario demostrar que la estructura deliberativa individual conserva coherencia y fidelidad normativa. Un sistema multiagente sin esa base sólida correría el riesgo de generar negociación sin fundamento o argumentación inconsistente.

Por ello, la evaluación del modelo se articula en torno a cuatro criterios, que funcionan como indicadores de su solidez conceptual:

1. **Coherencia normativa:** correspondencia interna entre normas, jerarquías y acciones derivadas.
2. **Trazabilidad lógica:** posibilidad de seguir el origen de cada argumento desde las creencias y condiciones que lo habilitan.
3. **Adaptabilidad deliberativa:** capacidad del agente para actualizar intenciones ante ataques o cambios contextuales.
4. **Explicabilidad jurídica:** transparencia en la justificación de cada decisión argumentativa.

Estos criterios no buscan medir desempeño computacional, sino examinar el grado en que el modelo mantiene fidelidad con la estructura racional del Derecho. En ese sentido, la evaluación no representa un cierre técnico, sino una confirmación conceptual: la demostración de que el razonamiento jurídico puede formalizarse y ejecutarse de manera coherente, trazable y explicable.

V.1. Diseño experimental

El diseño experimental del modelo tuvo como propósito evaluar la coherencia estructural del agente jurídico propuesto, más que resolver de forma definitiva una controversia legal. Para ello se eligió el caso *Pierson v. Post* (1805) [3], un precedente emblemático en el derecho de propiedad anglosajón que plantea de manera nítida el conflicto entre la intención y la posesión efectiva, y que por tanto ofrece un terreno fértil para analizar el razonamiento normativo en condiciones de ambigüedad jurídica.

Contexto del caso *Pierson v. Post*

En agosto de 1805, la Corte Suprema del Estado de Nueva York resolvió un caso que se convertiría en una referencia clásica en el estudio del derecho de propiedad. El litigio se originó cuando **Lodowick Post**, mientras cazaba un zorro con sus perros en tierras baldías, fue interceptado por **Jesse Pierson**, quien mató al animal y se lo llevó, a pesar de que Post se encontraba en plena persecución —lo que en la tradición inglesa se denomina *hot pursuit*.

Post demandó a Pierson argumentando que su persecución le otorgaba un derecho de propiedad sobre el zorro, mientras que Pierson sostuvo que el derecho correspondía a quien capturó físicamente al animal. El tribunal inferior falló a favor de Post, pero la Corte Suprema de Nueva York revocó esa decisión, estableciendo que la mera persecución no confiere propiedad; para adquirir dominio sobre un animal salvaje (*ferae naturae*) es indispensable la *posesión corporal* o haberlo herido mortalmente durante la caza.

La **opinión mayoritaria**, redactada por el juez Tompkins, sostuvo que el principio de ocupación derivado del Derecho romano exige control físico o efectivo sobre el bien para que exista apropiación legítima. En cambio, la **opinión disidente** del juez Livingston argumentó que la persecución intensa, cuando demuestra intención y esfuerzo razonable para apoderarse del animal, debería bastar para reclamar propiedad, pues fomenta un comportamiento más ético y deportivo entre los cazadores.

Así, el caso contrapone dos visiones jurídicas fundamentales:

- una **visión formalista**, representada por Tompkins, que exige un acto de posesión material para la existencia de un derecho;
- y una **visión pragmática o intencionalista**, representada por Livingston, que reconoce el valor jurídico de la intención y del esfuerzo racional en la consecución del objetivo.

Esta dualidad —posesión versus intención— ofrece un marco ideal para evaluar un agente

basado en la arquitectura BDI, donde las *creencias* (hechos jurídicos), los *deseos* (valores o fines normativos) y las *intenciones* (acciones o argumentos) interactúan para generar decisiones razonadas conforme a reglas explícitas.

Estrategia experimental

A partir del contexto descrito, el experimento se estructuró en tres fases complementarias:

1. **Inicialización del agente (Tomkins).** Se configuró el agente con un conjunto inicial de creencias (B), deseos (D) e intenciones (I), conforme a la estructura BDI presentada en el Capítulo 3. Estas entidades se codificaron en el programa Mercury mediante tipos fuertemente declarados (`belief`, `desire`, `argument`), lo que permitió garantizar consistencia y trazabilidad de los estados mentales del agente.
2. **Generación de argumentos.** A partir de las creencias y deseos iniciales, la función `add_arguments_to_Tomkins/1` filtró los argumentos válidos según las condiciones normativas activas (`condition`). Cada argumento resultante corresponde a una intención jurídicamente coherente, generada bajo el principio de jerarquía normativa y condicionalidad contextual.
3. **Ataque y reevaluación.** Se introdujo manualmente un ataque (`attack1a`) que modificó el conjunto de creencias del agente, simulando un cambio de contexto o la aparición de nueva evidencia jurídica. Este procedimiento obligó al agente a reevaluar sus intenciones y reconstruir sus argumentos, replicando así el ciclo deliberativo BDI:

$$B \xrightarrow{\alpha} B' \Rightarrow I' = \Gamma(B', D).$$

De esta manera, se observó cómo el modelo actualiza sus decisiones al modificarse las premisas normativas, preservando coherencia y justificabilidad.

El entorno de ejecución fue el lenguaje Mercury [13, 14], cuya semántica puramente declarativa y su tipado estático aseguran que cada relación lógica entre creencias, deseos e intenciones sea verificable antes de la ejecución. Este enfoque refuerza la validez del experimento como una *demostración estructural* del modelo.

En las siguientes secciones se presentan los escenarios de prueba, los resultados de la simulación del agente Tomkins y la discusión de los criterios de evaluación —coherencia normativa, trazabilidad lógica, adaptabilidad deliberativa y explicabilidad jurídica— que permiten valorar la fidelidad conceptual del modelo propuesto.

V.2. Escenarios de prueba

Los escenarios de prueba se diseñaron para observar el comportamiento deliberativo del agente Tomkins frente a distintos estados normativos y fácticos del caso *Pierson v. Post*. A diferencia de una validación cuantitativa, el objetivo aquí fue cualitativo: analizar cómo el modelo genera, sostiene y ajusta argumentos jurídicos cuando varían sus creencias o las condiciones normativas que las sustentan.

Definición de los escenarios

El experimento se estructuró en dos escenarios principales y un escenario transicional, los cuales reproducen los puntos de inflexión del caso:

1. **Escenario 1 – Creencias iniciales: persecución sin captura.** En este estado, las creencias del agente (B) describen la situación inicial en la que Post se encuentra en *hot pursuit* del zorro pero no ha logrado capturarlo (f1-f2). Las normas activas corresponden al nivel de *intención* (deseo) de reconocer mérito jurídico a la persecución, asociadas a los valores `less_litigation` (claridad legal) y `humaneness` (comportamiento ético en la caza). La condición relevante es `ownership_plaintiff`, que el agente evalúa positivamente. En este punto, el agente genera intenciones y argumentos orientados a la defensa de Post, consistentes con la idea de que la intención razonable y el esfuerzo sostenido deberían ser suficientes para fundar una expectativa de propiedad.
2. **Escenario 2 – Actualización normativa: posesión material.** A continuación se activa una nueva evidencia (`attack1a`) que modifica las creencias del agente: Pierson ha matado y tomado posesión física del zorro (f3). Este cambio produce una transición de estado $B \rightarrow B'$ conforme al esquema modal de Kripke descrito en el Capítulo 3, donde el mundo posible posterior contiene una nueva verdad fáctica que altera la aplicabilidad de las normas previas. En este contexto, las condiciones `no_ownership_defendant` y `no_pursuit_no_ownership` se reevalúan como verdaderas, bloqueando deseos previos y reconfigurando el conjunto de intenciones. El agente, tras la reevaluación, genera nuevos argumentos en favor de Pierson, sosteniendo que solo la ocupación efectiva constituye apropiación legítima.
3. **Escenario transicional – Reevaluación deliberativa.** Finalmente, se ejecuta el ciclo de actualización:

$$B \xrightarrow{\alpha} B' \Rightarrow I' = \Gamma(B', D),$$

donde α representa el ataque normativo. Este ciclo permite comparar los conjuntos de intenciones antes y después del cambio. La reevaluación evidencia la capacidad del agente para abandonar argumentos incompatibles con el nuevo contexto y adoptar otros acordes con la norma jerárquicamente prevalente (`no_ownership_defendant`).

Representación en Mercury

Los escenarios se implementaron directamente en el prototipo Mercury (Listado VI.1) mediante inicialización y reevaluación del agente Tomkins.

El bloque inicial del código define las creencias y deseos en los tipos `belief` y `desire`:

```

1   Tomkins = agent(
2     [belief(f1), belief(f2), belief(f3), belief(f4),
3     unknown(f5), belief(f6), unknown(f7)],
4     [desire(clear_law, less_litigation, ownership_plaintiff),
5     desire(unclear_law, less_litigation, ownership_no_possession),
6     desire(trade_restricted, economic_benefit,
7     malicious_intent_productive_activity_defendant)],
8     []
9   ),
```

Este bloque representa el *estado inicial del caso*: las creencias `f1-f7` describen los hechos, mientras que las condiciones normativas (`ownership_plaintiff`, `ownership_no_possession`) activan los deseos correspondientes. La función `add_arguments_to_Tomkins/1` construye los argumentos válidos, filtrando aquellos que satisfacen $B \models \text{cond}(d)$, según el criterio formal definido en la ecuación (III-1). El resultado es un conjunto de intenciones I coherente con la interpretación doctrinal de Tomkins (formalista moderado).

Posteriormente, el escenario transicional se desencadena con un ataque, representado por el predicado `apply_attack/3`:

```

1   apply_attack(TomkinsUpdated, Attack1, TomkinsUpdatedAttack),
2   TomkinsReevaluated = add_arguments_to_Tomkins(
      TomkinsUpdatedAttack),
```

Esta operación modela la reevaluación deliberativa del agente ante nueva evidencia. El agente abandona los argumentos incompatibles con el contexto actualizado y genera un nuevo conjunto de intenciones I' , preservando la consistencia lógica y la trazabilidad de sus justificaciones. En términos conceptuales, este proceso reproduce la dinámica de reconsideración judicial: cuando una nueva norma o hecho altera las condiciones del caso, los argumentos válidos deben ser reconstruidos, no simplemente ajustados.

Interpretación cualitativa

Desde el punto de vista de la evaluación, los escenarios muestran tres comportamientos claves del modelo:

1. **Consistencia interna:** el agente mantiene coherencia entre sus creencias, deseos e intenciones en todo momento, evitando contradicciones lógicas.
2. **Adaptabilidad:** ante la introducción del ataque, el sistema ajusta sus argumentos sin pérdida de estructura formal, lo que evidencia capacidad de respuesta racional ante cambios normativos.
3. **Explicabilidad:** cada argumento generado puede rastrearse hasta sus fundamentos normativos (**condition**) y su contexto fáctico (**beliefs**), cumpliendo con el principio de trazabilidad argumentativa.

En conjunto, estos escenarios no buscan demostrar una victoria procesal de Post o Pierson, sino exhibir que el modelo BDI implementado en Mercury puede sostener deliberaciones normativas consistentes, jerárquicamente justificadas y explicables. El caso *Pierson v. Post* no se resuelve, pero se convierte en un laboratorio conceptual donde el razonamiento jurídico se comporta como un proceso formal, verificable y auditado.

En la siguiente sección se abordan las métricas empleadas para evaluar este comportamiento, atendiendo a los cuatro criterios definidos: coherencia normativa, trazabilidad lógica, adaptabilidad deliberativa y explicabilidad jurídica.

V.3. Métricas de evaluación del modelo

La evaluación del modelo propuesto se orienta a verificar la correspondencia entre la estructura teórica —basada en la jerarquía kelseniana, la semántica de mundos posibles y la arquitectura BDI— y su comportamiento observable en la ejecución del prototipo Mercury (Listado VI.1). A diferencia de los enfoques experimentales convencionales, esta evaluación no busca cuantificar el desempeño del sistema, sino constatar su *fidelidad epistemológica*: que las inferencias generadas por el agente correspondan a los principios formales que definen el razonamiento jurídico.

El modelo se analiza conforme a cuatro criterios complementarios: **coherencia normativa**, **trazabilidad lógica**, **adaptabilidad deliberativa** y **explicabilidad jurídica**. Cada uno de ellos constituye una métrica conceptual que permite evaluar si el agente Tomkins

actúa, razona y justifica sus decisiones conforme a la estructura racional del Derecho y los fundamentos teóricos desarrollados en los capítulos anteriores.

V.3.1. Coherencia normativa

Este criterio mide la consistencia interna del modelo: que las normas, jerarquías y condiciones mantengan relaciones lógicas no contradictorias. En la ejecución de Mercury, la coherencia se observa en que el agente Tomkins nunca genera argumentos incompatibles entre sí. Por ejemplo, bajo las creencias iniciales (`belief(f1)` y `belief(f2)`), se activa la condición `ownership_plaintiff`, que orienta sus deseos hacia la acción `find_for_plaintiff`. Tras el ataque (`attack1a`), la modificación de las creencias a `belief(f3)` provoca que dicha condición se invalide y sea sustituida por `no_ownership_defendant`. En este punto, el agente elimina las intenciones previas que entrarían en conflicto con la nueva norma jerárquicamente superior, en conformidad con el principio de precedencia de Kelsen ((III-4)).

Así, la coherencia normativa se demuestra porque el agente no conserva simultáneamente argumentos basados en normas incompatibles. En términos jurídicos, esto equivale a respetar el principio de no contradicción dentro del sistema normativo: una norma que pierde validez no puede seguir fundamentando una acción.

V.3.2. Trazabilidad lógica

La trazabilidad es el criterio que permite reconstruir el encadenamiento lógico desde las creencias (B) hasta las intenciones (I). Cada argumento del agente puede ser rastreado a su contexto de creencias, a la condición normativa que lo activa y al valor jurídico que lo respalda. En el modelo Mercury, esto se logra mediante la estructura tipada del agente:

$$\text{belief} \rightarrow \text{condition} \rightarrow \text{desire} \rightarrow \text{argument}$$

Este esquema implementa el flujo formal $B \models \text{cond}(d) \Rightarrow d \in D \Rightarrow I = \Gamma(B, D)$, descrito en la Ecuación (III-1). El agente no actúa como una caja negra: cada intención es reconstruible desde los elementos que la justifican. Esto responde al ideal de transparencia normativa propuesto por Prakken y Sartor [8], donde la validez de una conclusión depende de su justificabilidad dentro del sistema de reglas explícitas.

En términos prácticos, la trazabilidad se evidencia cuando, al imprimir el conjunto de intenciones, es posible identificar la norma (tipo `condition`) y el valor (`value`) que motivaron cada acción. El proceso de deliberación es, por tanto, auditable y verificable.

V.3.3. Adaptabilidad deliberativa

Este criterio evalúa la capacidad del modelo para ajustarse racionalmente ante cambios en las creencias o en las condiciones normativas.

El mecanismo `apply_attack/3` actúa como un operador modal que transforma el conjunto de creencias $B \rightarrow B'$, representando una transición entre mundos posibles en el sentido de Kripke [74, 53].

Posteriormente, la función `add_arguments_to_Tomkins/1` recomputa las intenciones a partir del nuevo contexto, de acuerdo con la ecuación (III-2).

Este ciclo evidencia que el modelo es deliberativo y no determinista en el sentido clásico: las conclusiones del agente dependen del estado actual de sus creencias. Al introducir un ataque o evidencia nueva, el sistema no sólo reevalúa sus argumentos, sino que descarta aquellos que resultan inválidos y adopta otros compatibles con el nuevo contexto. En el dominio jurídico, esto refleja la *revisión de tesis* o reinterpretación normativa ante precedentes o pruebas sobrevenidas: un proceso racional y explicable de adaptación del razonamiento jurídico a la variación contextual.

V.3.4. Explicabilidad jurídica

Finalmente, la explicabilidad jurídica evalúa la claridad con la que el modelo permite comprender el porqué de cada decisión.

El agente Tomkins no sólo genera conclusiones, sino que mantiene una estructura de justificación en la que cada argumento está compuesto por:

1. las creencias fácticas que lo sostienen (`beliefs`),
2. la condición normativa que lo activa (`condition`),
3. la acción jurídica que propone (`action`), y
4. el valor subyacente que busca maximizar (`value`).

Esta estructura cumple con el principio de *explicabilidad por construcción* defendido en modelos de IA jurídica contemporáneos [51], donde la transparencia no depende de post-procesamiento, sino del propio diseño lógico del agente. Cada decisión, incluso las que resultan desactivadas tras un ataque, puede ser comprendida en términos normativos y valorativos, preservando la interpretabilidad del proceso deliberativo.

En conjunto, los cuatro criterios permiten afirmar que el modelo cumple con los principios de coherencia, trazabilidad, adaptabilidad y explicabilidad que se esperaban de un sistema

multiagente normativo. La validación no consiste en medir su desempeño computacional, sino en comprobar que los conceptos jurídicos —norma, jerarquía, validez, contexto y valor— son formalmente representables y operativos en un entorno lógico-declarativo.

La ejecución en Mercury demuestra, por tanto, que la arquitectura BDI puede sostener un razonamiento jurídico formalizado sin perder su fundamento normativo. El resultado es un agente capaz de construir y revisar argumentos conforme a un marco teórico explícito, confirmando que la formalización del razonamiento jurídico no sólo es posible, sino verificable en términos computacionales. En la siguiente sección se analizan los alcances y limitaciones de esta validación, así como las proyecciones hacia un entorno multiagente completo.

V.4. Resultados

Los resultados que se presentan en esta sección corresponden a la ejecución del prototipo desarrollado en el lenguaje lógico-declarativo Mercury, descrito en el Listado VI.1. Su propósito no es verificar el rendimiento computacional del sistema, sino analizar su comportamiento normativo y argumentativo frente al caso jurídico *Pierson v. Post* [3]. En particular, se evalúa la capacidad del agente Tomkins para generar, sostener y reevaluar argumentos coherentes bajo los principios formales definidos en los capítulos anteriores: la jerarquía kelseniana [52], la condicionalidad contextual [73], la semántica de mundos posibles de Kripke [74] y la estructura deliberativa BDI [27, 28, 11].

En esta fase experimental, el agente Tomkins se concibe como una *instancia operativa del modelo teórico*: un ente deliberativo capaz de vincular creencias fácticas (B), deseos normativos (D) y acciones intencionales (I) dentro de un marco de razonamiento normativo explícito. El propósito es observar cómo, ante un contexto normativo definido y un conjunto de creencias iniciales, el agente:

1. construye argumentos normativos válidos y trazables;
2. selecciona intenciones compatibles con su jerarquía de condiciones y valores;
3. reevalúa sus conclusiones ante la introducción de un ataque (modificación de creencias);
4. y mantiene coherencia entre normas, creencias y valores.

Los resultados se organizan en cinco apartados. Primero, se describe la configuración inicial del agente Tomkins (5.4.1), que representa la postura formalista del juez Tompkins en la sentencia original. En segundo lugar, se muestra la generación inicial de argumentos (5.4.2) y cómo estos reflejan el razonamiento normativo de la posición mayoritaria del tribunal. Posteriormente, se analiza la simulación del ataque y la reevaluación del agente (5.4.3),

observando los cambios en sus creencias e intenciones. La cuarta sección (5.4.4) interpreta estos resultados a la luz de las doctrinas jurídicas (formalismo, equidad, utilitarismo). Finalmente, se ofrece una síntesis general (5.4.5), donde se destaca la correspondencia entre el comportamiento del agente y los criterios teóricos de coherencia, trazabilidad, adaptabilidad y explicabilidad definidos en la Sección 5.3.

En suma, esta sección muestra que la ejecución del agente Tomkins en Mercury no constituye una mera implementación técnica, sino la verificación empírica de que el razonamiento jurídico puede ser formalizado y ejecutado como un proceso deliberativo explícito, sustentado en fundamentos normativos y lógicos bien definidos.

V.4.1. Configuración inicial del agente Tomkins

El agente *Tomkins* representa la postura formalista de la opinión mayoritaria en el caso *Pierson v. Post* [3], redactada por el juez Daniel Tompkins. En dicha opinión, se sostiene que la propiedad sobre un animal *ferae naturae* sólo puede adquirirse mediante la ocupación física o la herida mortal, rechazando la idea de que la mera persecución constituya un acto suficiente de apropiación. Esta visión, basada en una noción estricta de legalidad, enfatiza la certeza jurídica sobre la equidad circunstancial, y se ajusta naturalmente a un esquema de razonamiento deliberativo de tipo BDI (*Beliefs–Desires–Intentions*) [27, 28, 11].

En el prototipo implementado en Mercury (Código V.1), esta configuración inicial se codifica mediante tres componentes explícitos: (i) el conjunto de **creencias** *B*, que expresan los hechos jurídicos y fácticos que el agente asume como verdaderos o inciertos; (ii) el conjunto de **deseos** *D*, que reflejan los fines normativos o valorativos que el agente busca promover; y (iii) el conjunto vacío de **intenciones** *I*, que será posteriormente poblado por el motor deliberativo (`add_arguments_to_Tomkins/1`).

```

1      Tomkins = agent(
2      [belief(f1), belief(f2), belief(f3), belief(f4),
3      unknown(f5), belief(f6), unknown(f7)],
4      [desire(clear_law, less_litigation, ownership_plaintiff),
5      desire(unclear_law, less_litigation, ownership_no_possession),
6      desire(trade_restricted, economic_benefit,
7      malicious_intent_productive_activity_defendant)],
8      [],
9      ),

```

Código V.1: Configuración inicial del agente *Tomkins*.

En este estado inicial, el agente Tomkins posee un conjunto de creencias que modelan los hechos esenciales del caso:

- `belief(f1)`: Post estaba en persecución del zorro.
- `belief(f2)`: Post no había capturado ni herido al zorro.
- `belief(f3)`: Pierson mató al zorro para arruinar el deporte de Post.
- `belief(f4)`: Los zorros matan ganado.
- `belief(f6)`: Reducir el número de zorros protege al ganado.
- `unknown(f5)`, `unknown(f7)`: se desconocen los efectos de incentivar o desalentar la caza sobre el sufrimiento animal.

Estas proposiciones constituyen el espacio factual del agente, o universo proposicional B , equivalente a la base epistémica descrita en el Capítulo 3. Las creencias **f1–f3** modelan hechos del caso, mientras que **f4–f7** introducen consideraciones de interés público y bienestar animal, ampliando la perspectiva de decisión más allá del mero dominio propietario.

Por su parte, los deseos D capturan los fines normativos que orientan el razonamiento del agente:

- `desire(clear_law, less_litigation, ownership_plaintiff)`: expresa el objetivo de promover una ley clara que reduzca la litigiosidad, favoreciendo al demandante (Post).
- `desire(unclear_law, less_litigation, ownership_no_possession)`: representa la posibilidad de que la falta de claridad genere incertidumbre sobre la posesión.
- `desire(trade_restricted, economic_benefit, malicious_intent_productive_activity_defenda` modela la tensión entre conducta maliciosa y beneficio económico, como factor contextual.

Cada deseo se compone de un *nombre* (fin jurídico), un *valor* (criterio evaluativo) y una *condición normativa*, reflejando la estructura tripartita formalizada en el Capítulo 3:

$$D = \{\text{desire}(n, v, c) \mid n \in \text{Nombre}, v \in \text{Valor}, c \in \text{Condición}\}.$$

En el momento inicial, el conjunto de intenciones I está vacío ($I = \emptyset$), lo cual corresponde al estado previo al proceso deliberativo. De acuerdo con el formalismo BDI presentado en la Ecuación (III-1), las intenciones se derivarán posteriormente cuando se cumplan las condiciones normativas $\text{cond}(d)$ asociadas a los deseos activos.

Desde un punto de vista jurídico, esta configuración inicial reproduce el *estado cognitivo* de la postura formalista: el agente cree en los hechos principales (la persecución, la intervención de Pierson y los efectos en la comunidad) y orienta sus deseos hacia la certeza normativa y la reducción del litigio, más que hacia la equidad del caso individual. En otras palabras, Tomkins parte de un marco epistémico y axiológico que privilegia la estabilidad del sistema jurídico sobre la reparación moral o emocional del demandante.

En síntesis, el agente Tomkins se inicializa con un conjunto de creencias consistentes, un vector de deseos normativos con prioridad por la claridad jurídica y la estabilidad social, y un conjunto vacío de intenciones a la espera del proceso deliberativo. Este punto de partida constituye el *estado base* del modelo, a partir del cual se observarán los resultados de la generación y reevaluación de argumentos en las subsecciones siguientes.

V.4.2. Generación inicial de argumentos

Una vez establecida la configuración inicial del agente *Tomkins*, el modelo ejecuta el proceso de *deliberación inicial*, en el cual se activan los deseos cuyas condiciones normativas se satisfacen bajo las creencias actuales B , y se generan las intenciones I correspondientes. El proceso puede describirse como:

$$I = \{ a \in \Gamma(B, d) \mid d \in D, B \models \text{cond}(d) \}.$$

En la implementación en Mercury, este proceso está encapsulado en la función `add_arguments_to_tomkins/1`, mostrada parcialmente en el Código V.2, la cual aplica el constructor de argumentos Γ filtrando las combinaciones válidas de creencias y deseos.

```

1      :- func add_arguments_to_tomkins(agent) = agent.
2      add_arguments_to_tomkins(Tomkins) = TomkinsUpdated :-
3      Arguments = [arg1, arg2, arg3, arg4, arg5,
4      arg6, arg7, arg8, arg9, arg10],
5      ValidArguments = list.filter(should_add_argument(Tomkins), Arguments),
6      TomkinsUpdated = Tomkins^arguments := ValidArguments.
```

Código V.2: Función de generación inicial de argumentos en el agente *Tomkins*.

La función `should_add_argument/2` actúa como el filtro deliberativo que implementa la relación de satisfacción $B \models \text{cond}(d)$. Sólo los argumentos que se encuentran justificados por las creencias activas y los valores presentes en los deseos son incorporados al conjunto de intenciones I . Este comportamiento refleja fielmente el principio BDI de *coherencia racional*: el agente sólo se compromete con intenciones que son compatibles con sus estados mentales y su jerarquía normativa [26, 28].

Argumentos generados. Al ejecutar la actualización inicial, el agente *Tomkins* construye un conjunto de argumentos válidos, tal como se ilustra en la salida de consola (fragmento de ejecución):

```
1      = Beliefs of Tomkins agent: =
2      [belief(f1), belief(f2), belief(f3), belief(f4), unknown(f5), belief(f6),
3        unknown(f7)]
4
5      = Desires of Tomkins agent: =
6      [desire(clear_law, less_litigation, ownership_plaintiff),
7        desire(unclear_law, less_litigation, ownership_no_possession),
8        desire(trade_restricted, economic_benefit,
9        malicious_intent_productive_activity_defendant)]
10
11     = Intentions (arguments) of Tomkins agent: =
12     [arg3, arg5, arg8]
```

Código V.3: Fragmento de salida: generación inicial de intenciones.

En esta ejecución inicial, el agente selecciona tres argumentos activos: **arg3**, **arg5** y **arg8**. Cada uno refleja un tipo distinto de razonamiento jurídico dentro del marco formalista del caso *Pierson v. Post*:

- **arg3**: se apoya en las creencias **f1** (Post estaba en persecución del zorro) y **f5** (fomentar la caza reducirá el número de zorros), con la acción **find_ownership_established**. Representa el razonamiento según el cual la conducta del demandante podría interpretarse como un intento legítimo de apropiación. Sin embargo, su activación inicial obedece al principio de claridad legal más que a equidad: el agente “considera” esta posibilidad antes de descartarla bajo normas de mayor rango.
- **arg5**: basado en la creencia **f2** (Post no había capturado ni herido al zorro) y el valor **less_litigation**, conduce a la acción **find_no_ownership**. Este argumento reproduce directamente la postura del juez Tompkins: la persecución no genera derechos de propiedad y fallar en ese sentido preserva la certeza jurídica [3].
- **arg8**: sustentado en la creencia **f3** (Pierson mató al zorro para arruinar el deporte de Post) y orientado al valor **public_benefit**, se asocia con la acción **find_for_plaintiff**. Este argumento representa una inclinación moral o de interés público que, aunque presente, no domina la jerarquía normativa del agente.

El conjunto inicial de intenciones de *Tomkins* (**{arg3, arg5, arg8}**) muestra la coexistencia de tres niveles de razonamiento normativo:

1. un nivel **fáctico-normativo** (la relación entre persecución y posesión, **arg3**);
2. un nivel **doctrinal-formalista** (la exigencia de ocupación efectiva, **arg5**);
3. y un nivel **axiológico-contextual** (la valoración del comportamiento malicioso, **arg8**).

La interacción entre estos niveles confirma la operacionalización del modelo teórico: el agente no produce una única conclusión lógica, sino un conjunto de argumentos jerarquizados según su relevancia normativa. El formalismo de Kelsen [52] se evidencia en la preferencia de **arg5**, que representa la norma de mayor rango; mientras que la semántica de Kripke [74] se manifiesta potencialmente en la estructura que permite su reevaluación posterior.

En esta primera fase deliberativa, el modelo demuestra que es capaz de construir argumentos jurídicos consistentes y trazables sin recurrir a reglas heurísticas o aprendizaje estadístico. Cada intención está sustentada en una combinación explícita de creencias, deseos y valores, todos ellos tipados y verificados estáticamente por el compilador de Mercury [13, 14]. Esto constituye la primera evidencia experimental de la fidelidad normativa del modelo: el agente actúa conforme a los principios lógicos y jurídicos que definen su marco conceptual, anticipando la dinámica de reevaluación que se observará en la siguiente subsección.

V.4.3. Simulación de ataque y reevaluación de argumentos

En el razonamiento jurídico real, las conclusiones rara vez son estáticas: los argumentos se fortalecen, debilitan o transforman conforme se introducen nuevos hechos o interpretaciones. El modelo BDI implementado en *Tomkins* incorpora esta dinámica mediante el mecanismo de **ataque argumentativo**, que altera el conjunto de creencias B y desencadena un nuevo ciclo de deliberación. Este proceso corresponde a la transición formal descrita a continuación:

$$B \xrightarrow{\alpha} B' \Rightarrow I' = \{ a \in \Gamma(B', d) \mid d \in D, B' \models \text{cond}(d) \}.$$

En el prototipo, el ataque se define como un evento que cuestiona una o varias creencias del agente, activando condiciones normativas específicas. El Código V.4 muestra la definición del ataque inicial empleado en la simulación experimental.

```

1  :- func attack1a = attack.
2  attack1a = attack(
3  [belief(f1)],
4  [ownership_plaintiff],
5  [disbelief(f1)]
6  ).

```

Código V.4: Definición del ataque **attack1a** en el agente *Tomkins*.

Este ataque representa una objeción al hecho **f1** (“Post estaba en persecución del zorro”), asociada a la condición **ownership_plaintiff**. En términos jurídicos, simula la introducción de una evidencia contraria que cuestiona la validez de la persecución como fundamento de propiedad —una línea argumentativa coherente con la crítica del juez Livingston a la postura formalista de Tompkins, aunque aquí se usa como mecanismo de prueba del modelo.

El ataque se aplica al agente mediante el predicado **apply_attack/3**, mostrado en el Listado V.5. Este procedimiento implementa la actualización de creencias, sustituyendo las proposiciones afectadas por el ataque y generando un nuevo estado B' .

```

1      :- pred apply_attack(agent::in, attack::in, agent::out) is det.
2      apply_attack(!.Agent, Attack, !:Agent) :-
3      Attack = attack(QuestionedBeliefs, Conditions, ArgumentBeliefs),
4      list.foldl((pred(Belief::in, A0::in, A::out) is det :-
5      ( if belief_in_list(Belief, A0^beliefs) then
6      ( if list.member(desire(_, _, Condition), A0^desires),
7      list.member(Condition, Conditions) then
8      list.delete_all(A0^beliefs, Belief, TempBeliefs),
9      list.append(TempBeliefs, ArgumentBeliefs, NewBeliefs),
10     A = A0^beliefs := NewBeliefs
11     else
12     A = A0
13     )
14     else
15     A = A0
16     )
17     ), QuestionedBeliefs, !Agent).
```

Código V.5: Aplicación del ataque y actualización de creencias.

En este fragmento, el ataque actúa como un operador sobre el conjunto de creencias del agente ($A0^{\text{beliefs}}$), eliminando las proposiciones cuestionadas e introduciendo las contrarias. Así, la transición $B \rightarrow B'$ se ejecuta explícitamente mediante sustitución de elementos en la lista de creencias.

Tras la actualización, el sistema invoca nuevamente la función deliberativa **add_arguments_to_tomkins/1** para reconstruir el conjunto de intenciones. El fragmento de ejecución en consola (Listado V.6) muestra los resultados antes y después del ataque.

```

1      = Intentions (arguments) of Tomkins agent before attack: =
2      [arg3, arg5, arg8]
3
4      = Beliefs of Tomkins agent after reevaluation: =
5      [disbelief(f1), belief(f2), belief(f3), belief(f4),
```

```

6      unknown(f5), belief(f6), unknown(f7)]
7
8      = Intentions (arguments) of Tomkins agent after reevaluation: =
9      [arg5, arg8]

```

Código V.6: Ejecución parcial: impacto del ataque y reevaluación de argumentos.

El resultado muestra que, tras el ataque, el agente elimina la creencia **f1** (la persecución del zorro) y la reemplaza por su negación **disbelief(f1)**. Como consecuencia, el argumento **arg3** —que dependía directamente de **f1**— deja de ser válido, y el conjunto de intenciones I' se reduce a $\{\mathbf{arg5}, \mathbf{arg8}\}$.

Este cambio tiene un significado jurídico claro: al desaparecer la creencia de que Post perseguía legítimamente al zorro, se debilita cualquier argumento que intente fundamentar una pretensión de propiedad derivada de la persecución. Permanece, en cambio, el razonamiento estrictamente formalista (**arg5**) y el valor moral contextual (**arg8**), coherente con la preferencia jerárquica del agente por la certeza legal.

Desde la semántica de Kripke [74, 53], este episodio puede entenderse como una transición de mundo posible:

$$w \xrightarrow{\text{attack}1a} v,$$

donde w es el conjunto original de creencias y v el conjunto revisado B' . El conjunto de intenciones I' que emerge en v es consistente con las normas activas bajo ese nuevo estado de creencias. El modelo, por tanto, no sólo actualiza datos, sino que ejecuta una *reevaluación epistémico-normativa*, conservando coherencia lógica y fidelidad normativa (criterios de la Sección 5.3).

El experimento confirma que el modelo propuesto posee capacidad de adaptación deliberativa: ante un ataque que modifica sus creencias, el agente Tomkins no “pierde” la coherencia, sino que reconfigura su marco argumentativo conforme a los principios que rigen su jerarquía normativa. El hecho de que las intenciones se reduzcan, pero no se contradigan, demuestra la estabilidad interna del modelo BDI implementado. Además, esta reevaluación ilustra que los ataques en Mercury no operan de forma arbitraria ni estadística, sino conforme a una semántica formal controlada —lo que distingue a este enfoque de los sistemas expertos tradicionales y de los modelos basados en aprendizaje automático.

En la siguiente subsección (5.4.4), se examina la correspondencia entre estos resultados y las distintas doctrinas jurídicas (formalismo, equidad, utilitarismo), así como su relevancia para evaluar la coherencia teórico-normativa del modelo.

V.4.4. Correspondencia doctrinal y análisis de resultados

La interpretación de los resultados observados en *Tomkins* a la luz de tres líneas doctrinales presentes en *Pierson v Post* y en la literatura de IA & Derecho: (i) formalismo (mayoría; juez Tompkins), (ii) equidad/razonamiento práctico (disenso; juez Livingston), y (iii) utilitarismo (bienestar público y efectos agregados). El objetivo no es “clasificar” el sistema como uno u otro, sino mostrar que el modelo BDI y su implementación en Mercury *son capaces de expresar y adaptar* argumentos que encarnan estas doctrinas, manteniendo coherencia y trazabilidad.

En *Pierson v. Post*, la mayoría sostiene que la mera persecución no confiere propiedad: se requiere ocupación/captura (posesión efectiva) [3]. En el agente, este sesgo hacia la certeza normativa se ve reflejado en:

- el **valor** `less_litigation` como criterio de selección de intenciones [10], que privilegia reglas claras y reducción de litigio;
- las **acciones** `find_no_ownership` y `find_for_defendant` respaldadas por creencias como `f2` (no hay captura ni herida), codificadas en `arg5` y `arg7`.
- la **persistencia** de intenciones formalistas tras el ataque (Codigo V.6), donde `arg5` permanece y `arg3` (dependiente de la persecución, `f1`) desaparece al ser cuestionada.

Este comportamiento es consistente con la mayoría: ante duda sobre *hot pursuit*, el sistema recae en la norma clara (ocupación), reduciendo ambigüedad y litigio (alineado con [9, 8]).

El desacuerdo defiende que una *persecución* puede bastar para fundar un título razonable (en aras de incentivar conductas deportivas y desincentivar el oportunismo) [10]. En el prototipo, esta visión está modelada por:

- la **dependencia** del argumento `arg3` (`find_ownership_established`) de la creencia `f1` (*Post estaba en persecución*)
- la **sensibilidad** del sistema a los ataques que cuestionan `f1`, al convertirse en `disbelief(f1)`, `arg3` deja de ser una intención válida.

Con ello, el modelo no descarta la equidad: la admite cuando el *estado de creencias* lo respalda, y la retira cuando el contexto probatorio se vuelve adverso [26]. Este ajuste contextual es precisamente lo que se espera de un razonamiento práctico con compromisos intencionales explícitos [28].

La tercera línea enfatiza efectos agregados (protección del ganado, reducción de plagas, sufrimiento animal). En el código, esta dimensión aparece como **valores** y **metas** explícitos:

- `arg1` (`encourage_fox_hunting`) se apoya en la cadena `f4 → f5 → f6` y prioriza `public_benefit`;
- `arg2` (`discourage_fox_hunting`) invoca `f7` y el valor `humaneness`.

La coexistencia de `arg1` y `arg2` *no* es una contradicción, sino la evidencia de que el modelo puede **poner en tensión valores** y resolverlos mediante condiciones activas y preferencias normativas (Sección 5.3). Esta forma de “razonamiento con valores” es central en la literatura [9, 8].

LO abordado en las subsecciones anteriores sugiere tres conclusiones:

1. El **núcleo formalista** del agente (tipo Tompkins) se mantiene estable cuando hay incertidumbre sobre la persecución: prevalecen intenciones ligadas a ocupación y reducción de litigio (`arg5`, `arg7`).
2. La **equidad** se activa sólo si el contexto *B* la sostiene: `arg3` es viable en presencia de `f1` y se desactiva cuando `f1` es cuestionada coherentemente por un ataque.
3. Los **criterios utilitaristas** coexisten como líneas argumentales alternativas, moduladas por *creencias* y *valores* (p.ej., `public_benefit` vs. `humaneness`), sin romper la coherencia del sistema.

Los resultados son consistentes con una visión contemporánea del razonamiento jurídico como *construcción, evaluación y ajuste de teorías* con valores [9, 8]: el agente no aplica reglas “en seco”, sino que *justifica* y *reconfigura* intenciones con base en creencias y condiciones activas. El que `arg3` desaparezca tras el ataque y `arg5` persista no es un artefacto de programación, sino el efecto esperable de un modelo con estados mentales BDI y jerarquías normativas explícitas.

Esta correspondencia doctrinal refuerza dos hallazgos: (i) el modelo soporta **interpretaciones múltiples** de una misma controversia y (ii) la **dinámica ante ataques** mantiene la coherencia interna. En la Sección 5.5 analizamos los alcances y límites de esta evidencia: la elección deliberada de un solo agente, la naturaleza manual de los ataques y la proyección a un escenario multiagente con protocolos de negociación e interoperabilidad semántica (FIPA) [44, 45].

Capítulo VI

Conclusiones y trabajo futuro

Esta investigación condujo desde la reflexión teórica sobre el razonamiento jurídico hasta su traducción formal y su verificación computacional. En los capítulos previos, se trazó un trayecto que partió de la pregunta fundacional —si es posible modelar el razonamiento normativo sin perder su coherencia lógica ni su sentido jurídico— y culminó en una demostración experimental mediante la simulación del agente *Tomkins*.

El presente capítulo busca integrar sus resultados del modelo desarrollado y valorar su alcance como aporte a la inteligencia artificial jurídica. En este sentido, las conclusiones deben entenderse no como un punto final, sino como el cierre de un ciclo epistemológico: desde la formulación de una hipótesis sobre la racionalidad del Derecho hasta la evidencia formal de que dicha racionalidad puede expresarse, ejecutarse y observarse computacionalmente.

A diferencia de los capítulos técnicos, este cierre se articula en torno a una lectura reflexiva del modelo. Se examina su coherencia conceptual, la relación entre sus componentes teóricos y su implementación, y las implicaciones que ello tiene para el estudio del Derecho como sistema de razones, valores y normas susceptibles de representación lógica. Así, las siguientes secciones abordan, de manera integrada, la síntesis del modelo y su alcance, el desarrollo de sus componentes, las aportaciones obtenidas, las limitaciones metodológicas y las líneas futuras que emergen de este trabajo.

VI.1. Síntesis del modelo y su alcance

El modelo desarrollado a lo largo de esta investigación integró una doble aspiración: por un lado, capturar la estructura racional del razonamiento jurídico y, por otro, demostrar que dicha estructura puede representarse de manera formal, ejecutable y explicable. Su núcleo es un sistema deliberativo inspirado en la arquitectura BDI (*Belief–Desire–Intention*) [11, 28],

adaptado al contexto normativo mediante los principios de jerarquía kelseniana, condicionalidad contextual y semántica de mundos posibles. Esta integración, que hasta ahora no se había articulado en un entorno computacional operativo, constituye el eje conceptual del trabajo.

Desde el punto de vista teórico, el modelo articula cuatro niveles de racionalidad normativa. Primero, la *validez derivada* de las normas (*Grundnorm*) en la tradición de Kelsen [52] se tradujo en una jerarquía explícita de condiciones y prioridades, permitiendo resolver conflictos entre normas en función de su rango. Segundo, la condicionalidad contextual, propia de la teoría de los sistemas normativos [73], introdujo la idea de que la aplicabilidad de una norma depende de las creencias activas del agente. Tercero, la semántica de mundos posibles de Kripke [74, 53] sirvió como marco para modelar la dinámica de cambio en las creencias y, por tanto, en las intenciones. Finalmente, los estándares FIPA [44, 45] otorgaron al modelo un vocabulario semántico interoperable, preparado para la futura interacción multiagente.

En su dimensión formal, la propuesta tomó forma en el lenguaje lógico-declarativo Mercury [13, 14], elegido por sus propiedades de tipado fuerte, determinismo y trazabilidad. La elección de Mercury no respondió a una preferencia instrumental, sino a la necesidad de demostrar que la lógica jurídica puede traducirse a un entorno computacional que preserve la fidelidad normativa sin ambigüedades semánticas. La sintaxis estricta y la verificación estática de Mercury garantizan que cada creencia, deseo e intención del agente *Tomkins* corresponda a un constructo lógico explícito y verificable. De este modo, el paso de la teoría a la ejecución no implicó una pérdida de significado, sino una comprobación empírica del modelo conceptual.

La prueba de concepto se desarrolló mediante la simulación del caso *Pierson v. Post* [3], clásico del derecho de propiedad, que permitió observar cómo el agente Tomkins construye y reevalúa argumentos conforme a principios normativos. En este contexto, las creencias representaron hechos jurídicos, los deseos reflejaron finalidades o valores —como la justicia, la utilidad pública o la reducción de litigios— y las intenciones emergieron como argumentos válidos derivados de la coherencia entre ambos planos. El modelo no buscó “resolver” la disputa, sino mostrar que un agente racional puede mantener coherencia y explicabilidad en su proceso deliberativo frente a nuevas circunstancias o ataques argumentativos.

El resultado es una arquitectura normativa que combina la estructura lógica del Derecho con las dinámicas cognitivas del razonamiento intencional. Cada argumento generado por el agente puede rastrearse hasta sus fundamentos: una norma superior que le confiere validez, un contexto de creencias que la activa y un valor que le otorga sentido jurídico. Esta trazabilidad —inexistente en modelos heurísticos o estadísticos— constituye la prueba más sólida de la

fidelidad conceptual del modelo. La ejecución en Mercury, lejos de ser un ejercicio técnico, se convirtió en la validación formal de que el razonamiento jurídico puede expresarse en términos computacionales sin perder su estructura normativa ni su transparencia argumentativa.

El alcance del modelo se manifiesta, por tanto, en tres dimensiones complementarias. En el plano teórico, demuestra que la integración de teorías jurídicas clásicas y lógicas formales contemporáneas es viable y coherente. En el plano metodológico, ofrece una vía replicable para la formalización de razonamientos normativos mediante lenguajes declarativos. En el plano aplicado, sienta las bases para el desarrollo de sistemas multiagente deliberativos capaces de negociar, justificar y reinterpretar normas con fundamento lógico. En conjunto, estos resultados confirman que el modelo no sólo representa una contribución técnica, sino también epistemológica: una demostración de que el Derecho, como sistema de razones, puede pensarse y modelarse bajo los principios de la inteligencia artificial simbólica.

VI.2. Desarrollo y articulación de los componentes del modelo

El desarrollo del modelo fue resultado de un proceso de integración progresiva entre niveles conceptuales, formales y experimentales. Cada uno de sus componentes respondió a una pregunta distinta sobre la posibilidad de representar el razonamiento jurídico de manera computacional, pero todos convergieron en una misma hipótesis: que la estructura normativa del Derecho puede ser modelada como un sistema de agentes racionales que deliberan, justifican y revisan sus propias creencias.

El punto de partida fue la definición de un marco multiagente inspirado en la arquitectura BDI, que ofrece una ontología mental clara —creencias, deseos e intenciones— para modelar procesos deliberativos [11, 28]. Este enfoque permitió trasladar la racionalidad práctica del Derecho al plano formal: las *creencias* representaron hechos y contextos jurídicos; los *deseos*, fines normativos o valores; y las *intenciones*, las acciones o conclusiones derivadas de esa estructura motivacional. Con ello se alcanzó el primer propósito del trabajo: disponer de un marco formal que capturara la dinámica de la deliberación jurídica como un proceso racional continuo.

El segundo componente correspondió a la formalización del razonamiento normativo. En esta etapa, el modelo incorporó principios derivados de la lógica deóntica y de la semántica de Kripke [74, 53], junto con los postulados de la jerarquía kelseniana [52] y de la teoría contextual de las normas [73]. La combinación de estos marcos permitió expresar la validez de una norma no sólo como una relación estática entre reglas, sino como una función de

contexto: una norma se activa o se desactiva en función de las creencias del agente. Así, la noción de *coherencia normativa* emergió como propiedad verificable: las normas y valores del sistema podían ser aplicados sin contradicciones internas, incluso ante cambios en el entorno o en la información disponible.

A partir de esta base teórica, se desarrolló la tercera fase: la traducción computacional del modelo. El uso del lenguaje Mercury [13, 14] fue decisivo, ya que su estructura lógico-declarativa permitió representar los vínculos entre creencias, deseos e intenciones como predicados verificables. La definición tipada de entidades —**belief**, **desire**, **intention**, **condition**— garantizó consistencia semántica, y su sistema de modos y determinismo proporcionó una correspondencia directa entre la lógica formal y la ejecución. Esta fase materializó la tercera meta del proyecto: demostrar que es posible implementar un razonamiento jurídico normativo dentro de un marco declarativo sin pérdida de trazabilidad ni ambigüedad interpretativa.

La cuarta dimensión del modelo se centró en la dinámica de la deliberación: la capacidad del agente para reevaluar sus decisiones ante la introducción de nuevos argumentos o ataques. Inspirado en los enfoques de argumentación formal [?, 50], el modelo incorporó la noción de transición epistémica: cada “ataque” actualiza las creencias del agente, forzando una nueva selección de intenciones. Este mecanismo reproduce, en términos computacionales, el comportamiento de un jurista que revisa su posición frente a nuevas evidencias o interpretaciones. En consecuencia, el modelo no se limita a derivar conclusiones, sino que mantiene una capacidad reflexiva, alineada con las teorías del razonamiento práctico [80, 81].

Finalmente, la validación conceptual se realizó a través del caso *Pierson v. Post* [3], cuya estructura factual y argumentativa permitió verificar la coherencia del modelo frente a un conflicto clásico de derechos de propiedad. El agente *Tomkins*, al representar una de las posturas jurídicas, fue capaz de generar argumentos trazables basados en su jerarquía normativa, sus valores y su contexto de creencias. Al someterlo a ataques simulados, el sistema mostró cómo las intenciones se reconfiguraban de acuerdo con la lógica formal del modelo, cumpliendo así con el propósito de demostrar la capacidad del agente para mantener coherencia y explicabilidad durante su deliberación.

De esta manera, el desarrollo del modelo puede entenderse como un proceso de convergencia: cada uno de sus componentes —teórico, formal, computacional y experimental— se ensambló para confirmar la hipótesis central de la investigación. El resultado no es un sistema experto que dicta veredictos, sino un agente racional capaz de sostener procesos argumentativos consistentes con los principios del razonamiento jurídico. En este punto, el modelo se revela no como un sustituto del jurista, sino como una demostración formal de

que el razonamiento normativo puede representarse, ejecutarse y evaluarse bajo las mismas exigencias de coherencia y racionalidad que definen al Derecho.

VI.3. Aportaciones teóricas, metodológicas y aplicadas

Las contribuciones derivadas de esta investigación pueden entenderse en tres planos complementarios —teórico, metodológico y aplicado— que, en conjunto, conforman una propuesta coherente para la representación computacional del razonamiento jurídico. Cada plano responde a una dimensión distinta del problema abordado: comprender cómo razona el Derecho, cómo formalizar ese razonamiento y cómo ponerlo a prueba dentro de un entorno lógico de ejecución.

Aportación teórica: una arquitectura unificada del razonamiento jurídico

En el plano teórico, la tesis propone una síntesis conceptual inédita entre cuatro tradiciones: la jerarquía normativa de Kelsen [52], la condicionalidad contextual de los enfoques IHL (*Institutional Hierarchy Logic*), la semántica modal de Kripke y la arquitectura cognitiva BDI. Esta integración no es una unión de teorías, sino una correspondencia estructural: la jerarquía kelseniana determina la validez normativa [74]; la lógica contextual define la aplicabilidad de las normas según las creencias; la semántica de Kripke permite representar el cambio epistémico; y la arquitectura BDI organiza estos procesos dentro de un ciclo racional de deliberación [28].

El resultado es un modelo que trasciende los sistemas expertos tradicionales y las aproximaciones estadísticas en IA jurídica, al restituir al razonamiento jurídico su carácter intencional, justificativo y explícitamente normativo. La tesis demuestra que el Derecho puede ser modelado no como un conjunto de reglas fijas, sino como un sistema dinámico de razones, capaz de actualizar sus conclusiones a partir de sus propias condiciones de validez.

Aportación metodológica: de la lógica formal a la validación ejecutable

En el plano metodológico, la investigación aporta una ruta de traducción desde el nivel lógico-formal hasta el nivel computacional ejecutable, garantizando fidelidad semántica en cada transición. El uso del lenguaje lógico-declarativo Mercury [13, 14] permitió conservar la estructura inferencial del modelo —creencias, deseos, intenciones y condiciones normativas— sin recurrir a procedimientos heurísticos ni a simplificaciones algorítmicas. Este enfoque demuestra que la representación de razonamiento jurídico no necesita depender de redes

neuronales ni modelos opacos: puede lograrse mediante sistemas simbólicos formalmente trazables.

La metodología desarrollada integra tres etapas: (i) la formalización teórica del razonamiento normativo; (ii) la construcción de un modelo BDI jurídico con jerarquía de normas y semántica de mundos posibles; y (iii) la validación en Mercury como entorno lógico de ejecución. Esta cadena metodológica establece un precedente replicable para futuros proyectos de modelado jurídico, en los que la coherencia teórica y la precisión computacional puedan coexistir sin sacrificios de interpretabilidad.

Aportación aplicada: demostración del modelo en un caso jurídico clásico

En el plano aplicado, la tesis ofrece una prueba de concepto basada en el caso *Pierson v. Post* (1805), utilizado históricamente como un ejemplo de conflicto entre propiedad, intención y comportamiento moral. La implementación del agente *Tomkins* permitió observar cómo un sistema racional puede generar argumentos jurídicos coherentes, evaluando hechos, valores y normas de manera explícita. Más allá de la ejecución técnica, el experimento constituye una demostración conceptual: el agente mostró la capacidad de revisar sus creencias y reformular sus intenciones frente a ataques, preservando la coherencia de su razonamiento.

Esta capacidad de *reevaluación deliberativa* distingue al modelo propuesto de los sistemas expertos tradicionales, que operan bajo reglas fijas. El modelo BDI jurídico no busca “resolver” el caso, sino representar el proceso de razonamiento que un jurista seguiría para justificar su posición y adaptarla frente a nuevas condiciones normativas. En ese sentido, la aplicación del modelo en Mercury sirvió no sólo como prueba empírica, sino como evidencia epistémica de que el razonamiento jurídico puede expresarse mediante estructuras computacionales que preservan el sentido racional y explicativo del Derecho [8, 9, 10].

En conjunto, las tres dimensiones de la contribución articulan un marco general de *IA jurídica simbólica*, donde la formalización lógica y la deliberación racional convergen en un mismo plano operativo. Teóricamente, la tesis consolida la posibilidad de una arquitectura normativa coherente. Metodológicamente, propone un itinerario de implementación replicable. Y, en el plano aplicado, demuestra la viabilidad de una simulación explicable del razonamiento jurídico. Así, el modelo no sólo constituye un avance técnico, sino una respuesta conceptual al desafío contemporáneo de construir sistemas de inteligencia artificial que razonen con —y no sólo sobre— el Derecho.

VI.4. Limitaciones y proyección del trabajo futuro

El desarrollo del modelo propuesto se concibió bajo la noción de un *sistema multiagente deliberativo*, en el que distintas instancias jurídicas podrían interactuar bajo marcos normativos diferenciados. Sin embargo, la implementación realizada en esta tesis se limitó intencionalmente a un solo agente —*Tomkins*— como paso inicial para validar la coherencia teórica y formal del modelo. Esta decisión no representa una omisión, sino una elección metodológica: antes de modelar la interacción entre agentes, era necesario consolidar un marco deliberativo internamente consistente, capaz de sostener procesos argumentativos trazables y coherentes con los principios normativos establecidos.

Desde esta perspectiva, el modelo actual constituye el *núcleo deliberativo* de un sistema multiagente. Su arquitectura, su tipificación en Mercury y la integración de jerarquías normativas permiten proyectar la incorporación de nuevos agentes —*Livingston*, *Tropolle*, *Banks*— que representarían distintas posturas doctrinales o jurisdicciones. Una futura extensión del modelo permitirá que dichos agentes interactúen mediante protocolos de negociación normativa y mecanismos de contraargumentación formal, siguiendo líneas como las desarrolladas en los modelos de Carneades [?] o ASPIC+ [8].

Otra limitación deliberada fue el alcance del caso de estudio. La simulación de *Pierson v. Post* no pretendió resolver el conflicto jurídico en su totalidad, sino verificar que el agente fuera capaz de generar, justificar y reevaluar argumentos de manera consistente. Este nivel de validación, si bien acotado, es el necesario para demostrar la *viabilidad epistemológica* del modelo, antes de avanzar hacia escenarios multiagente o de aprendizaje adaptativo.

En el contexto contemporáneo, esta aproximación adquiere un valor especial. La irrupción de los modelos de lenguaje de gran escala (LLMs) —como ChatGPT o Claude— ha redefinido la percepción pública de la inteligencia artificial, pero también ha puesto en evidencia una tensión conceptual: dichos sistemas, aunque potentes en generación lingüística, carecen de estructura lógica, intencionalidad y trazabilidad argumentativa. En contraste, el modelo aquí presentado se inscribe en la tradición de la *IA simbólica explicable*, donde cada inferencia puede ser reconstruida y justificada. Mientras los LLMs ofrecen fluidez discursiva, el enfoque BDI-normativo ofrece *razonamiento estructurado*. En un futuro cercano, la convergencia entre ambas aproximaciones —modelos generativos para el análisis textual y arquitecturas simbólicas para la deliberación racional— puede abrir una vía hacia sistemas jurídicos híbridos, capaces de unir comprensión contextual con coherencia normativa.

Entre las líneas de trabajo futuras destacan cuatro direcciones principales:

- **Extensión multiagente:** implementación de agentes adicionales con marcos norma-

tivos y valores distintos, para simular procesos de negociación y mediación jurídica.

- **Contraargumentación automatizada:** desarrollo de un protocolo de diálogo formal que permita la generación y evaluación dinámica de ataques y defensas normativas.
- **Aplicación en disputas de propiedad intelectual:** adaptación del modelo a casos reales contemporáneos, utilizando bases normativas actualizadas y ontologías FIPA extendidas.
- **Integración con modelos de lenguaje:** exploración de arquitecturas híbridas que combinen la formalidad del razonamiento simbólico con la flexibilidad lingüística de los modelos generativos actuales.

En síntesis, las limitaciones del presente trabajo no constituyen debilidades, sino fronteras metodológicas abiertas. El modelo logró demostrar que el razonamiento jurídico puede ser formalizado y ejecutado de manera coherente en un entorno lógico-declarativo. El paso siguiente es llevar esa coherencia a la interacción entre múltiples agentes normativos, abriendo así la posibilidad de construir sistemas jurídicos computacionales que no sólo procesen normas, sino que realmente *deliberen* sobre ellas.

El desarrollo de este trabajo se inscribe en la intersección de tres disciplinas tradicionalmente distantes: la filosofía, el derecho y la computación. Desde la filosofía, se recuperó la noción de *razonamiento práctico* y la idea de que toda decisión jurídica implica una estructura de justificación. Desde el derecho, se asumió la exigencia de coherencia normativa, jerarquía y validez que caracteriza a los sistemas jurídicos positivos. Y desde la computación, se aportó el rigor formal y la capacidad de representar dinámicamente esas estructuras en un entorno lógico-declarativo. La convergencia de estos tres campos permitió formular una propuesta donde el razonamiento jurídico no sólo se explica, sino que se *ejecuta* de manera verificable.

El caso de estudio *Pierson v. Post* sirvió como laboratorio conceptual para poner a prueba esta integración. Más allá de su valor histórico, el caso permitió modelar un conflicto normativo clásico —la tensión entre propiedad, intención y moralidad— dentro de un marco deliberativo formal. El agente *Tomkins*, aunque limitado a una sola instancia, mostró que un sistema simbólico puede construir, justificar y reevaluar argumentos de forma coherente con los principios jurídicos que lo sustentan. El resultado no pretende resolver disputas, sino mostrar que el razonamiento jurídico puede ser formalizado sin perder su racionalidad ni su fundamento normativo.

En el contexto contemporáneo, donde los modelos de lenguaje masivo dominan el discurso sobre inteligencia artificial, esta tesis ofrece una perspectiva complementaria. Mientras los

sistemas generativos producen respuestas lingüísticamente verosímiles pero opacas en su justificación, el modelo propuesto se apoya en principios trazables, jerarquías explícitas y relaciones lógicas verificables. No busca imitar el pensamiento jurídico, sino demostrar que es posible representarlo con fidelidad y transparencia.

El valor de esta investigación reside, por tanto, en haber articulado un puente metodológico entre tres tradiciones: la reflexión filosófica sobre la norma, el análisis jurídico de la validez y la implementación computacional de la coherencia. Si bien la validación se limitó al dominio del derecho de propiedad intelectual, la arquitectura desarrollada es lo suficientemente general para extenderse a otros contextos normativos: ética algorítmica, políticas públicas automatizadas o mediación institucional. En todos ellos, la necesidad de agentes capaces de razonar con normas, justificar sus decisiones y adaptarse al cambio normativo es cada vez más evidente.

Así, este trabajo no representa un punto final, sino un punto de partida. En una era dominada por inteligencias que producen sin comprender, este modelo ofrece una alternativa: una inteligencia que *razona*, que explica, y que se compromete con la estructura lógica y ética del Derecho. Un pequeño aporte, pero un aporte fundamentado, hacia la construcción de sistemas artificiales que no sólo procesen información, sino que comprendan el sentido normativo de sus propias decisiones.

Referencias

- [1] J. G. Corvalán, “Inteligencia artificial: retos, desafíos y oportunidades - prometea: la primera inteligencia artificial de latinoamérica al servicio de la justicia,” *Revista de Investigações Constitucionais*, 2018.
- [2] D. Restrepo Amariles and P. M. Baquero, “Promises and limits of law for a human-centric artificial intelligence,” *Computer Law & Security Review*, vol. 48, p. 105795, 2023.
- [3] N. Y. C. of Appeals, “Pierson v. post, 3 cai. r. 175 (n.y. 1805),” 1805. Disponible en https://www.nycourts.gov/reporter/archives/pierson_post.htm (última consulta: noviembre 2024).
- [4] Kira Systems, “Kira Systems - AI for Contract Analysis,” 2024. Última consulta: noviembre 2024.
- [5] Luminance Technologies, “Luminance - AI-Powered Legal Document Review,” 2024. Última consulta: noviembre 2024.
- [6] Casetext, “Casetext - Advanced Legal Research and AI Tools,” 2024. Última consulta: noviembre 2024.
- [7] CaseMine, “CaseMine - Legal Research with Artificial Intelligence,” 2024. Última consulta: noviembre 2024.
- [8] H. Prakken and G. Sartor, “Law and logic: A review from an argumentation perspective,” *Artificial Intelligence*, vol. 227, pp. 214–245, 2015.
- [9] T. Bench-Capon and G. Sartor, “A model of legal reasoning with cases incorporating theories and values,” *Artificial Intelligence*, vol. 150, no. 1, pp. 97–143, 2003.
- [10] K. Atkinson and T. Bench-Capon, “Legal case-based reasoning as practical reasoning,” *Artificial Intelligence and Law*, vol. 13, pp. 93–131, 2005.

- [11] M. Wooldridge, *An introduction to multiagent systems*. John Wiley & Sons, 2009.
- [12] N. Jennings, K. Sycara, and M. Wooldridge, “A roadmap of agent research and development,” *Autonomous Agents and Multi-Agent Systems*, vol. 1, pp. 7–38, 03 1998.
- [13] Z. Somogyi, F. Henderson, and T. Conway, “The execution algorithm of mercury, an efficient purely declarative logic programming language,” *The Journal of Logic Programming*, vol. 29, no. 1, pp. 17–64, 1996. High-Performance Implementations of Logic Programming Systems.
- [14] F. Henderson, T. Conway, Z. Somogyi, D. Jeffery, P. Schachte, S. Taylor, C. Speirs, T. Dowd, R. Becket, and M. Brown, “The mercury language reference manual,” 1996.
- [15] V. Aleven, “Using background knowledge in case-based legal reasoning: A computational model and an intelligent learning environment,” *Artificial Intelligence*, vol. 150, no. 1, pp. 183–237, 2003.
- [16] F. Dignum, “Autonomous agents with norms,” *Artificial Intelligence and Law*, vol. 7, no. 1, pp. 69–79, 1999.
- [17] O. M. de la Propiedad Intelectual, “Otro año record en las solicitudes de títulos de propiedad intelectual,” Dec. 2018.
- [18] I. M. de la Propiedad Industrial, “Actúa el impi a favor de los derechos de propiedad intelectual,” Nov. 2018.
- [19] H. Alessa, “The role of artificial intelligence in online dispute resolution: A brief and critical overview,” *Information & Communications Technology Law*, vol. 31, no. 3, pp. 319–342, 2022.
- [20] A. Lodder and J. Zeleznikow, “Developing an online dispute resolution environment: Dialogue tools and negotiation support systems in a three-step model. harvard negotiation law review 10,” vol. 10, 08 2007.
- [21] Judicial Commission of New South Wales, “Argumentation in legal reasoning,” 2021. Accessed: November 2023.
- [22] C. C. Stav Fainshmidt, George O. White III, “Legal distance, cognitive distance, and conflict resolution in international business intellectual property disputes,” *Journal of International Management*, 2014.

- [23] J. Wan, “Application of game theory in a patent dispute negotiation,” *les Nouvelles*, 2015.
- [24] L. Machina, “Legal analytics.” Website, Oct. 2018.
- [25] M. Wooldridge, *The logical modelling of computational multi-agent systems*. PhD thesis, The University of Manchester, 1992.
- [26] P. R. Cohen and H. J. Levesque, “Intention is choice with commitment,” *Artificial Intelligence*, vol. 42, no. 2, pp. 213–261, 1990.
- [27] A. S. Rao and M. P. Georgeff, “Bdi agents: From theory to practice,” in *International Conference on Multiagent Systems*, 1995.
- [28] J.-J. Meyer, J. Broersen, and A. Herzig, “BDI logics,” in *Handbook of Logics for Knowledge and Belief* (H. van Ditmarsch, J. Y. Halpern, W. van der Hoek, and B. Kooi, eds.), vol. Chapter10, pp. 453–493, College Publications, 2015.
- [29] A. García-Camino, J. A. Rodríguez-Aguilar, C. Sierra, and W. Vasconcelos, “Constraint rule-based programming of norms for electronic institutions,” *Autonomous agents and multi-agent systems*, vol. 18, no. 1, pp. 186–217, 2009.
- [30] C. S. Marc Esteva, Julian Padget, “Formalizing a language for institutions and norms,” *Intelligent Agents VIII*, vol. 2333, pp. 348–366, June 2002.
- [31] N. Fornara and M. Colombetti, “Specifying and enforcing norms in artificial institutions,” in *International Workshop on Declarative Agent Languages and Technologies*, pp. 1–17, Springer, 2008.
- [32] F. M. Wagdi Alrawagfeh, “Utilizing permission norms in bdi practical normative reasoning,” *Coordination, Organizations, Institutions, and Norms in Agent Systems X*, vol. 9372, pp. 1–18, Nov. 2015.
- [33] W. W. Vasconcelos, A. García-Camino, D. Gaertner, J. A. Rodríguez-Aguilar, and P. Noriega, “Distributed norm management for multi-agent systems,” *Expert Systems with Applications*, vol. 39, no. 5, pp. 5990–5999, 2012.
- [34] K. S. Marina De Vos, Tina Balke, “Combining event-and state-based norms,” *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, vol. International Foundation for Autonomous Agents and Multiagent Systems, pp. 1157–1158, 2013.

- [35] G. Governatori and A. Rotolo, “Bio logical agents: Norms, beliefs, intentions in defeasible logic,” *Autonomous Agents and Multi-Agent Systems*, vol. 17, no. 1, pp. 36–69, 2008.
- [36] A. Ciampolini, E. Lamma, P. Mello, F. Toni, and P. Torroni, “Cooperation and competition in alias: A logic framework for agents that negotiate,” *Ann. Math. Artif. Intell.*, vol. 37, pp. 65–91, 01 2003.
- [37] V. Mascardi, M. Martelli, and L. Sterling, “Logic-based specification languages for intelligent software agents,” *Theory and Practice of Logic Programming*, vol. 4, 12 2003.
- [38] V. Mascardi, D. Demergasso, and D. Ancona, “Languages for programming bdi-style agents: an overview.,” pp. 9–15, 01 2005.
- [39] Y. Shoham and M. Tennenholtz, “On the synthesis of useful social laws for artificial agent societies (preliminary report),” in *AAAI*, pp. 276–281, 1992.
- [40] R. Conte and C. Castelfranchi, “Understanding the functions of norms in social groups through simulation,” *Artificial societies: the computer simulation of social life*, vol. 1, pp. 252–267, 1995.
- [41] T. Yu and L. Zhang, “An agent-based model of social norm and cooperation,” *Procedia Computer Science*, vol. 17, pp. 951–957, 2013.
- [42] P. Ribino, Lodato, Lopes, Sabatucci, V. Seidita, M. Cossentino, C. Lodato, S. Lopes, L. Sabatucci, and V. Seidita, “Ontology and goal model in designing bdi multi-agent systems,” vol. 1099, 12 2013.
- [43] IEEE Foundation for Intelligent Physical Agents (FIPA), “Design Process Documentation Template.” <http://www.fipa.org/>, 2001.
- [44] T. F. Yannis Labrou, “Semantics for an agent communication language,” *Intelligent Agents IV Agent Theories, Architectures, and Languages*, vol. 1365, pp. 209–214, 1997.
- [45] M. M. K. Shujun Li, *Agent Communication Language*. Springer, Sept. 2012.
- [46] P. M. Nowotko, “Ai in judicial application of law and the right to a court,” *Procedia Computer Science*, vol. 192, pp. 2220–2228, 2021.
- [47] J. A. Blair, “Walton’s argumentation schemes for presumptive reasoning: A critique and development,” *Argumentation*, vol. 15, pp. 365–379, 2001.

- [48] K. D. Ashley, “Tutorial on ai and legal reasoning,” 2005.
- [49] E. L. Rissland, K. D. Ashley, and R. Loui, “Ai and law: A fruitful synergy,” *Artificial Intelligence*, vol. 150, no. 1, pp. 1–15, 2003. AI and Law.
- [50] G. R. Simari and R. P. Loui, “A mathematical treatment of defeasible reasoning and its implementation,” *Artificial Intelligence*, vol. 53, no. 2, pp. 125–157, 1992.
- [51] J. Collenette, K. Atkinson, and T. Bench-Capon, “Explainable ai tools for legal reasoning about cases: A study on the european court of human rights,” *Artificial Intelligence*, vol. 317, p. 103861, 2023.
- [52] M. S. Green, “Hans kelsen and the logic of legal systems,” *Alabama Law Review*, vol. 53, pp. 365–413, 2003.
- [53] T. Braüner and V. de Paiva, “Intuitionistic hybrid logic,” *Journal of Applied Logic*, vol. 4, no. 3, pp. 231–255, 2006. Methods for Modalities 3 (M4M-3).
- [54] Mercury Language Development Team, “Notes on the design of the mercury implementation,” 2023. Accessed: November 2023.
- [55] Mercury Language Page, “Mercury language reference manual,” 2023. Accessed: November 2023.
- [56] W. F. Clocksin and C. S. Mellish, *Programming in Prolog: Using the ISO Standard*. Springer, 5 ed., 2003.
- [57] J. Wielemaker, “Swi-prolog.” <http://www.swi-prolog.org>, 2011.
- [58] P. Moura, *Logtalk - Object-Oriented Logic Programming Language*, 2011.
- [59] G. Nadathur and D. Miller, “An overview of λ prolog,” in *Fifth International Logic Programming Conference*, pp. 810–827, MIT Press, 1988.
- [60] M. Hanus, “A unified computation model for functional and logic programming,” in *Proceedings of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp. 80–93, ACM, 1997.
- [61] P. Hudak, *The Haskell school of expression: learning functional programming through multimedia*. Cambridge University Press, 2000.
- [62] J. Armstrong, *Programming Erlang: Software for a Concurrent World*. Pragmatic Bookshelf, 2007.

- [63] M. Odersky, L. Spoon, and B. Venners, *Programming in Scala*. Artima Publishing, 3 ed., 2016.
- [64] B. S. Henry Kautz, “Planning as satisfiability,” 1992.
- [65] C. Z. V. S. Subrahmanian, *Relating stable models and ai planning domains*, vol. 12 of *Proceedings of the Twelfth International Conference on Logic Programming*. MIT Press, 1995.
- [66] D. E. Wilkins, “Domain-independent planning representation and plan generation,” *Artificial Intelligence*, vol. 22, no. 3, pp. 269–301, 1984.
- [67] T. L. T. S. Christian Anger, Kathrin Konczak, “A glimpse of answer set programming.,” *KI*, vol. 19, no. 1, p. 12, 2005.
- [68] Y. Lierler, “Handout on answer set programming,” 2017.
- [69] R. C. Cardoso and A. Ferrando, “A review of agent-based programming for multi-agent systems,” *Computers*, vol. 10, no. 2, 2021.
- [70] S. Batsakis, G. Baryannis, G. Governatori, I. Tachmazidis, and G. Antoniou, “Legal representation and reasoning in practice: A critical comparison.,” in *JURIX*, pp. 31–40, 2018.
- [71] B. P. Allen, L. Stork, and P. Groth, “Knowledge engineering using large language models,” 2023.
- [72] S. Franklin and A. Graesser, “Is it an agent, or just a program?: A taxonomy for autonomous agents,” in *International Workshop on Agent Theories, Architectures, and Languages*, pp. 21–35, Springer, 1996.
- [73] C. Braga, E. H. Haeusler, and J. S. Santos, “Modeling normative multi-agent systems from a kelsenian perspective,” *arXiv preprint*, 2018.
- [74] J.-J. C. Meyer, R. J. Wieringa, *et al.*, “Deontic logic in computer science normative system specification,” *HeinOnline*, vol. 3, p. 93, 1993.
- [75] D. van Dalen, *Logic and Structure*. Universitext, Springer London, 2012.
- [76] R. Vidal-López and J. AHernández-Servín, “Inclusión de la inteligencia artificial en disputas legales sobre propiedad intelectual,” *Komputer Sapiens*, vol. 2, 2019.

- [77] R. Vidal-López, J. L. González-Ruíz, Marcial-Romero, J. Raymundo, and J. A. Hernández-Servín, *Towards a Formal Language in Law*, p. 16. IGI Global, 2023.
- [78] S. Mac Lane and I. Moerdijk, “Sheaves in geometry and logic a first introduction to topos theory,” 1992.
- [79] G. E. Reyes, “A topos-theoretic approach to reference and modality,” *Notre Dame Journal of Formal Logic*, vol. 32, no. 3, pp. 359–391, 1991.
- [80] M. Bratman, *Intention, plans, and practical reason*, vol. 10. Harvard University Press Cambridge, MA, 1987.
- [81] J. Broome, “Normative practical reasoning: John broome,” in *Aristotelian Society Supplementary Volume*, vol. 75, pp. 175–193, Wiley Online Library, 2001.

Anexos

Inclusión de la Inteligencia Artificial en Disputas Legales sobre Propiedad Intelectual

por Rodrigo Vidal-López y José Antonio Hernández-Servín

La Importancia de las Disputas Legales sobre Propiedad Intelectual

Las disputas legales sobre la propiedad intelectual son un recurso utilizado por los inventores como medio de protección de sus creaciones [1]. Estas disputas legales se complican cuando la infracción en propiedad intelectual es cometida en territorios con procesos legales distintos, generalmente, internacionales. La importancia de estas disputas se remonta a que tan sólo en 2015 la solitu-

des de patentes creció un 9.2 % en los Estados Unidos, el cual es el crecimiento más rápido en los últimos 18 años [2]. Para 2017, la Organización Mundial de la Propiedad Intelectual, reportó un récord en solicitudes y registro de patentes, pero esta vez en China [3]. Existen dos formas para llevar a cabo una disputa legal, una de ellas es la negociación y la otra el litigio. La primera de ellas, es un común acuerdo que se da entre las partes involucradas en mutuo beneficio y el litigio por otro lado es el conflicto entre las partes involucradas en un juicio.

Las disputas legales son un recurso utilizado por los inventores como medio de protección de sus creaciones.

De acuerdo a Fainshmidt *et al.* [1], existen dos conceptos importantes que influyen en la toma de decisiones para la resolución de los conflictos, los cuales son: **la distancia cognitiva** y **la distancia legal**. La **distancia cognitiva** se refiere a las malas interpretaciones que pueden surgir de entre las partes involucradas, es decir, la educación y los contextos sociales de las partes involucradas son diferentes, por lo cual refuerza la idea de una resolución de conflictos mediante un litigio y no una negociación. La **distancia legal**, se asocia de manera positiva a llevar una negociación como resolución de conflictos en lugar de un litigio. Como se puede observar, estas dos alternativas generan a su vez otro conflicto entre sí, de acuerdo a las definiciones y referencias actuales ofrecen poca información en la elección de la solución de conflictos.

Los litigios como resolución de conflictos

Los litigios son la parte medular de los conflictos ya que existen temas de rivalidad y contrademandas, lo que en ocasiones llevan demasiado tiempo en resolverse, sin embargo, se impone a la idea de negociar un acuerdo fuera de los tribunales de justicia, como sugiere la distancia legal, ya que este acuerdo puede ser considerado como un juego de suma-cero en el que “el ganador se lleva todo” [2] y por supuesto que ninguna de las partes involucradas quiere perder.

Por tal razón, en el estudio realizado por Fainshmidt *et al.* [1] se examinaron las controversias internacionales en materia de propiedad intelectual, resueltas en tribunales de los Estados Unidos. Los datos analizados se

obtuvieron de la base de datos de 94 tribunales federales, teniendo como resultados una correlación de la distancia legal y cognitiva, donde a mayores distancias legales y cognitivas darán lugar a litigios, en vez de negociaciones, porque la distancia cognitiva refuerza el efecto negativo de la distancia legal sobre elegir la negociación como una estrategia de resolución de conflictos. Al mismo tiempo, la distancia legal se asocia positivamente con la probabilidad de obtener una negociación como resolución, porque las partes involucradas no quieren tener un conflicto legal, al no saber las leyes con las que se rige la parte anfitriona, lo que podría considerarse una ventaja para esta, es decir, el lugar y con las leyes donde se lleva a cabo la disputa legal.

Inteligencia Artificial (IA) en Disputas Legales

La IA se ha interesado en los años recientes en el problema que se presenta, al tratar de obtener una resolución de conflictos en las disputas legales, específicamente en las que están relacionadas con la propiedad intelectual.

Existen dos ramas de la IA con las que se ha tratado el problema de la resolución de conflictos, como son: la **Teoría de Juegos** y la **Minería de Datos**. La Teoría de Juegos se ha utilizado en litigios para negociar la solución a la controversia que logre resultados favorables para los oligopolistas (un grupo pequeño de involucrados). La solución a la controversia con teoría de juegos se hace mediante un análisis del comportamiento y to-



CARTA DE ACEPTACIÓN

México. D.F, 28 de junio de 2019

Rodrigo Vidal López and José Antonio Hernández Servín

Estimados colegas

A nombre de la Sociedad Mexicana de Inteligencia Artificial (SMIA) tengo el placer de informarles que como resultado del proceso de arbitraje, su artículo

Inclusión de la Inteligencia Artificial en Disputas Legales sobre Propiedad Intelectual

ha sido aceptado por el Comité Editorial para su publicación en la revista de divulgación Komputer Sapiens. El artículo será publicado en el año 11 volumen 2, correspondiente al periodo mayo-agosto de 2019, cuya temática es "Tema Libre".

Para la preparación de su manuscrito, les pedimos sean tan amables de atender las instrucciones que hemos enviado por correo electrónico.

Agradecemos el tiempo dedicado a la preparación de su manuscrito y los invitamos a seguir colaborando con esta revista.

Atentamente

Dra. Laura Cruz Reyes
Editor en Jefe

Dra. Karina Mariela Figueroa Mora
Editora Científica

Chapter 4

Towards a Formal Language in Law

Rodrigo Vidal-López

Universidad Autónoma del Estado de México, Mexico

Jacobo Leonardo González-Ruíz

Universidad Autónoma del Estado de México, Mexico

José Raymundo Marcial-Romero

Universidad Autónoma del Estado de México, Mexico

J. A. Hernández-Servín

 <https://orcid.org/0000-0001-8951-0063>

Universidad Autónoma del Estado de México, Mexico

ABSTRACT

In today's globalized society, the generation of digital information grows exponentially along with the technological innovations in all aspects of everyday life. In order to cope with gathering, administering, sharing information, planning, and automatization of many human activities including some human reasoning tasks, multi-agent systems in the form of language programming scripting have become an important aspect in dealing with those problems. Thus, the agent-based programming language has become an important line of research in artificial intelligence from a theoretical and practical point of view. In this chapter, the authors review some already implemented languages and theoretical aspects toward a formal proposal of a language that deals with human reasoning in law. In particular, the authors present as an example a language, based on topoi theory, that deals with the problem of defining ontology as specified by FIPA as a requirement for agent-based programming language.

DOI: 10.4018/978-1-6684-5624-8.ch004



eEditorial Discovery

Supporting Your Editorial Experience

Dear Rodrigo Vidal-López,

Thank you for submitting your chapter proposal, "Towards a formal language in Law," for the upcoming book, "Applied Artificial Intelligence and Robotics for Government Processes." After reviewing your recent proposal, I believe the topic of your proposed chapter is consistent with the theme of this book and could make a significant contribution to this project. I therefore encourage you to begin preparing your full chapter and submit a copy before Wednesday, April 20, 2022 at the following URL:

<https://www.igi-global.com/submission/submit-chapter/?projectid=bf7fd529-08d4-43d6-ac2f-e20eefe15ff2>

Please note that you will be asked to create an account prior to uploading your chapter to the system. This is to ensure the security of your work and to assist you in organizing your materials for submission, receiving and providing peer reviews, and making any necessary revisions to your chapter. For information on creating and accessing your Web account, please see our tutorial at www.igi-global.com/publish/contributor-resources/book-submission-system/video-guide/#creating-a-user-account.

The following documents are provided by the publisher, IGI Global (www.igi-global.com), to aid in the writing of your chapter:

Ensure your chapter has been professionally copy edited

It is crucial that professional copy editing is conducted prior to submission to ensure proper use of the English language, proper grammatical structure, and correct spelling and punctuation. Submitted chapters are considered to be in their final form and ready for publication as is. We recommend eContent Pro for copy editing (**currently offering 10% discount**): <https://www.econtentpro.com/copyediting>.

Details to keep in mind for your full chapter submission:

www.igi-global.com/publish/contributor-resources/before-you-write/

Image guide:

<http://www.igi-global.com/publish/resources/image-guide.pdf>

Your adherence to the guidelines provided in these documents is very important. Should you have any questions regarding your proposed chapter, please do not hesitate to contact me, David Valle-Cruz, at davacr@uaemex.mx.

I look forward to receiving your chapter!

IGI Global
eEditorial Discovery

You have received this email because you are associated with a project in the IGI Global eEditorial Discovery® system. Adjust where notifications are sent by adding or updating your primary email address at <https://www.igi-global.com/account/e-mail/> (login required). Please contact cust@igi-global.com for assistance.



Modelling Legal Judgements with BDI Agents in Mercury: Analysis of the Pierson Versus Post Case

Rodrigo Vidal-López and J. A. Hernández-Servín^(✉)

Universidad Autónoma del Estado de México, Toluca, Estado de México, Mexico
{rvidall,joseph.servin}@uaemex.mx

Abstract. This paper implements Belief-Desire-Intention operators in the Mercury Language based on Atkinson’s formalism to simulate legal judgements, specifically analyzing the Pierson versus Post case. By showcasing how a BDI agent dynamically adapts its belief and desire system through arguments and attacks, the research underscores the critical adaptability required for legal reasoning. The interplay between law, computer science, and philosophy is examined, and technical challenges in legal decision-making automation. The purpose of the paper also aims to short the distance between the logical multiagent formalization and practical applications in artificial intelligence within the law.

1 Introduction

The integration of artificial intelligence into the legal arena is not a new phenomenon. Since Gardner’s early work in 1987 [12], which explored artificial intelligence and its application in legal reasoning, to contemporary studies on expert systems and legal argumentation, there has been a steady development in how technology can serve the law. Gardner already highlighted the potential of AI to simulate cognitive processes associated with legal reasoning, setting a precedent for future research.

In 1992, Wooldridge [23] introduced fundamental concepts on multi-agent systems, which have been crucial for understanding how autonomous agents can interact within complex systems, including legal ones. This work laid the groundwork for the subsequent development of agent-based systems capable of simulating the dynamics of reasoning and legal disputes.

Logic and argumentation in modelling legal reasoning, emphasising how logical formalisation can capture the structure and process of legal debate [6]. The representation of legal knowledge and the generation of legal arguments through AI has inspired the development of tools that can support legal professionals in their daily work [4].

The formalisation of legal reasoning through programming languages and multi-agent systems, as proposed in [21], suggests a promising approach to addressing the complexity of law. However, the effectiveness of these systems crucially depends on their ability to faithfully reflect legal reasoning and adapt

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
R. Silhavy and P. Silhavy (Eds.): CSOC 2024, LNNS 1118, pp. 442–456, 2024.
https://doi.org/10.1007/978-3-031-70285-3_33

LETTER OF ACCEPTANCE

13th Computer Science On-line Conference 2024

Dear J. A. Hernandez-Servin,

Organizing & Program Committee is pleased to announce that your paper:

Modelling Legal Judgements with BDI Agents in Mercury: Analysis of the Pierson versus Post Case (Paper ID: 113007)

Author(s): Hernandez-Servin J. A., Vidal-Lopez Rodrigo,

was Accepted

for the 13th Computer Science On-line Conference 2024.

For finishing your registration follow instruction, which has been already sent by e-mail to all authors of accepted papers (or follow instruction on <https://csoc.openpublish.eu>)

CSOC2024 is held on-line from 4/25/2024 to 4/28/2024.

Conference organization (sponsored by): OpenPublish.eu

Organization Committee Chair:

Radek Silhavy, Ph.D.

A handwritten signature in black ink, appearing to read 'Radek Silhavy', with a large, stylized 'S' at the end.

Radek Silhavy, Ph.D.

Organizing Committee Chair

Aplicación del modelo del caso *Pierson v. Post* en Mercury

```
1      :- module test1.
2      :- interface.
3      :- import_module io.
4
5      % Prototipos de funciones y tipos.
6      :- pred main(io::di, io::uo) is det.
7
8      :- implementation.
9      :- import_module list, bool, string.
10     :- import_module pair.
11
12     % Definición de proposiciones utilizadas en el caso legal.
13     :- type proposition --->
14     f1; % Post estaba en persecución del zorro.
15     f2; % Post no había capturado ni herido al zorro.
16     f3; % Pierson mató al zorro para arruinar el deporte de Post.
17     f4; % Los zorros matan ganado.
18     f5; % Fomentar la caza reducirá el número de zorros.
19     f6; % Reducir el número de zorros protege al ganado.
20     f7. % Si se desalienta la caza, no se inflige sufrimiento innecesario a
        los animales.
21
22     % Las creencias pueden ser positivas, negativas o desconocidas respecto a
        las proposiciones.
23     :- type belief --->
24     belief(proposition);
25     disbelief(proposition);
26     unknown(proposition).
27
28     % Condiciones bajo las cuales los deseos son válidos.
29     :- type condition --->
30     ownership_plaintiff;
31     no_ownership_defendant;
32     no_ownership_no_possession;
33     ownership_no_possession;
34     no_ownership_plaintiff;
35     no_ownership_possession;
36     malicious_intent_productive_activity_defendant;
37     malicious_intent_plaintiff;
38     malicious_intent_defendant;
39     fewer_foxes_farmers_protected;
```

```

40     no_fewer_foxes_no_farmers_protected;
41     reduced_animal_suffering;
42     no_reduced_animal_suffering;
43     ownership_pursuit;
44     no_pursuit_no_ownership.
45
46     % Valores que un agente puede intentar maximizar o mantener.
47     :- type value --->
48     less_litigation;
49     economic_benefit;
50     public_benefit;
51     humaneness.
52
53     % Nombres de deseos que reflejan estrategias o metas legales.
54     :- type name --->
55     clear_law ;
56     unclear_law ;
57     trade_restricted ;
58     malice_condemned ;
59     malice_condoned ;
60     less_threat_to_others ;
61     more_threat_to_others ;
62     more_suffering ;
63     less_suffering .
64
65     % Acciones que un agente puede tomar basadas en el escenario legal.
66     :- type action --->
67     encourage_fox_hunting ;
68     discourage_fox_hunting ;
69     find_ownership_established ;
70     find_no_ownership ;
71     find_for_plaintiff ;
72     find_for_defendant ;
73     not_find_for_defendant .
74
75     % Los deseos combinan nombres, valores y condiciones.
76     :- type desire --->
77     desire(name, value, condition).
78
79     % Los argumentos son combinaciones de creencias, acciones, metas y
      valores.
80     :- type argument --->
81     argument(

```

```

82     beliefs :: list(belief),
83     action :: action,
84     goals :: list(proposition),
85     values :: list(value)
86 ).
87
88 % Los ataques están definidos para desafiar las creencias de los agentes.
89 :- type attack --->
90     attack(
91         question :: list(belief),
92         conditions :: list(condition),
93         argument :: list(belief)
94     ).
95
96 % Estructura del agente actualizada.
97 :- type agent --->
98     agent(
99         beliefs :: list(belief),
100        desires :: list(desire),
101        arguments :: list(argument)
102    ).
103
104 % Función para añadir argumentos válidos a un agente basándose en
105     creencias y deseos.
106 :- func add_arguments_to_\textit{Tomkins}(agent) = agent.
107 add_arguments_to_\textit{Tomkins}(\textit{Tomkins}) = \textit{Tomkins}
108     Updated :-
109 Arguments = [arg1, arg2, arg3, arg4, arg5, arg6, arg7, arg8, arg9, arg10
110 ],
111 ValidArguments = list.filter(should_add_argument(\textit{Tomkins}),
112     Arguments),
113 \textit{Tomkins}Updated = \textit{Tomkins}^arguments := ValidArguments.
114
115 % Punto de entrada para el programa Mercury.
116 main(!IO) :-
117     % Configuración inicial del agente con creencias y deseos.
118     \textit{Tomkins} = agent(
119         [belief(f1), belief(f2), belief(f3), belief(f4), unknown(f5), belief(f6),
120             unknown(f7)],
121         [desire(clear_law, less_litigation, ownership_plaintiff), desire(
122             unclear_law, less_litigation, ownership_no_possession), desire(
123                 trade_restricted, economic_benefit,
124                 malicious_intent_productive_activity_defendant)],

```



```

151     io.print("\n\n", !IO),
152
153     io.print("=_Intenciones_(argumentos)_de_\textit{Tomkins}_después_de_la_
      reevaluación:_=\n", !IO),
154     io.print(\textit{Tomkins}ReevaluatedArguments, !IO),
155     io.print("\n\n", !IO).

```

Código VI.1: Aplicación del modelo del caso *Pierson v. Post* en Mercury

***Answer Set Programming* como una argumentación formal**

En este apartado se presenta el resultado de una participación en el Coloquio del Posgrado del año 2020, CICTTACA. Este trabajo refleja una fase exploratoria en la búsqueda de soluciones computacionales para la argumentación legal, donde se investigó el potencial del *Answer Set Programming* (ASP) para estructurar argumentos lógicos y claros, libres de ambigüedades, en el ámbito legal.

El documento ofrece una perspectiva valiosa sobre cómo las técnicas de programación lógica-declarativa pueden contribuir al campo de la resolución de conflictos legales, incluso cuando más adelante en la investigación principal se prefirió la adopción de Mercury por sus características y ventajas específicas. Se elige incluir esta publicación en los anexos para:

1. Reconocer las contribuciones iniciales y el trabajo preliminar que informó el desarrollo posterior de la tesis.
2. Mostrar la evolución del enfoque metodológico desde ASP hacia Mercury, proporcionando una visión completa de la trayectoria de la investigación.
3. Valorar la capacidad de ASP para la formalización de argumentos, mientras se subraya la decisión de transición hacia Mercury por su idoneidad más marcada en el contexto de la investigación actual.

La participación en el coloquio sirvió como una plataforma para probar y discutir ideas iniciales, lo que eventualmente llevó a la refinación de los métodos y la elección de herramientas más apropiadas para los objetivos de esta tesis. La elección de Mercury sobre ASP fue motivada por [motivos específicos, por ejemplo, la representación de conocimiento más rica, mejor manejo de incertidumbre, estructuras de datos más complejas, etc.], características que lo hacían más adecuado para el modelado avanzado y la simulación de juicios legales que se buscaba alcanzar.

A pesar de que ASP no fue la herramienta finalmente elegida para el desarrollo del modelo de esta tesis, la publicación adjunta en este anexo ilustra el valor de la diversidad en enfoques computacionales para el razonamiento legal y ofrece un contraste útil con el trabajo realizado con Mercury. El aprendizaje obtenido de la experiencia con ASP ha contribuido a un entendimiento más profundo de las demandas de la argumentación formal y ha enriquecido la capacidad de elección de herramientas computacionales para tareas de razonamiento complejas.

Answer Set Programming como una argumentación formal

Vidal-López Rodrigo¹, Marcial-Romero José Raymundo², Hernández-Servín José Antonio³

¹⁻³Facultad de Ingeniería

Universidad Autónoma del Estado de México, Cerro de Coatepec S/N. Ciudad Universitaria. C.P. 50100. Toluca, Estado de México

¹rvidall@uaemex.mx, ²jrmarcialr@uaemex.mx, ³xoseahernandez@uaemex.mx

Resumen– Los seres humanos constantemente se involucran o son partícipes en alguna discusión. En estas discusiones, deben ser capaces de defender la postura que tienen respecto al tema en cuestión. Incluso, poder influir en el sistema de creencias de los demás participantes. Esto es posible a través de la argumentación. Una barrera que puede existir es que los argumentos contruidos contengan ambigüedades. Esto trae como consecuencia que los argumentos sean débiles y fácilmente atacados. Por lo tanto, generar argumentos mediante un lenguaje lógico-declarativo (*Answer Set Programming*, ASP), formaliza y permite obtener argumentos válidos en una discusión.

Palabras clave: ASP; argumentación; formalización; creencias; lógico-declarativo

Abstract– The human being, constantly gets involved or participates in discussions. In these discussions, they have to be capable of defending the posture they have in that matter. Even, they have to be capable of having influence in the beliefs of other participants. This is possible through the arguments. One of the obstacles we can find is that these arguments are ambiguous. In consequence, these arguments can be easily attacked and meaningless. Therefore, if we create arguments through logical-declarative and assertive language (*Answer Set Programming*, ASP), we can formalize and get an accurate and acceptable argument in a discussion.

Keywords: ASP; argumentation; beliefs; logical-declarative; formalization

áreas como la lógica [2]. La lógica acepta la validez de un argumento si la conclusión es una consecuencia de las premisas, es decir, si las premisas son verdaderas, entonces la conclusión también lo es [3].

Las ciencias computacionales han utilizado desde mediados de los 90's tratamientos para desarrollar una argumentación mediante la lógica con enfoques existentes para el razonamiento [2] utilizando la negación por fallo¹.

A través de marcos dialécticos, los cuales mediante una especificación sucinta y flexible se han propuesto formalismos para la argumentación [4]. Aunado a estos marcos dialécticos utilizando la Programación de Conjuntos de Respuestas (*Answer Set Programming*, ASP) se pueden obtener predicciones de aridad delimitada a través de diferentes instancias².

Los argumentos toman una mayor relevancia en disputas legales, debido a la construcción que debe tener el argumento, no tener ambigüedades y debe ser lo suficientemente sólido para que no sea fácilmente refutable [5].

Por estas razones, se pueden construir argumentos válidos utilizando las ciencias computacionales y un software de programación lógica declarativa como ASP, para después utilizarlos en disputas legales y de esta manera obtener una resolución de conflictos sobre esa disputa legal. En este documento se presenta la argumentación de un caso de estudio sobre una disputa legal [6].

1 Introducción

La comunicación existente entre los seres humanos es algo muy común y natural. Mediante esta, dos o más participantes puede intercambiar mensajes o simplemente como una forma de expresión [1]. Sin embargo, la comunicación en diversas ocasiones no tiene la efectividad que se desea. La argumentación, es un método que nos permite mejorar la comunicación mediante razonamientos que están en relación con diferentes

2 Metodología

Los programas en ASP son programas bien formados los cuales para lograr argumentos válidos se utiliza el principio lógico de la extensión por definición [7]. Fanndino et al. [7], proporciona una forma normal para programas lógicos similares a la lógica clásica. Teniendo esta forma normal, se modela mediante ASP y obtener los modelos estables³.

¹Se considera falso aquello que no exista evidencia clara para demostrar que es verdadero

²versiones de programa, diferentes hechos

³conjuntos de respuestas o soluciones al problema



Figura 1: Un programa en ASP se compone de 4 conjuntos

En la Figura 1 se puede observar que un programa en ASP es una cuadrupla (F, C, D, I) donde F es un conjunto de HECHOS, C un conjunto de OPCIONES/ELECCIONES, D un conjunto de REGLAS DEFINIDAS y finalmente I un conjunto de RESTRICCIONES DE INTEGRIDAD.

Loa HECHOS son aquellas proposiciones que se consideran verdaderas. Las REGLAS DEFINIDAS permiten analizar y/o extender la solución candidata, estas derivadas de los HECHOS. Después, las OPCIONES/ELECCIONES reflejan las elecciones reales para la solución del problema. Finalmente, estas opciones son evaluadas por las RESTRICCIONES DE INTEGRIDAD, resultando en los modelos estables que son solución al problema [4].

Planteado lo anterior, la metodología para poder establecer un problema argumentado en lógica, para después expresarlo en ASP consta de 4 pasos:

1. Identificar los hechos.
2. Establecer reglas definidas, es decir, estas reglas no cambian, conocidas como “reglas duras”.
3. Las reglas consideradas reglas de elección, para obtener los modelos candidatos a solución.
4. Evaluar que las reglas de elección cumplan con las restricciones o reglas de integridad.

Identificando y estableciendo estas 4 reglas, es suficiente para plantear un problema en ASP y así obtener los modelos solución.

En el algoritmo 1 se presenta en términos de ASP el problema de un Circuito Hamiltoniano, el cual se describe en el campo matemático de la teoría de grafos, es un camino de un grafo, una sucesión de aristas adyacentes, que visita todos los vértices del grafo una sola vez. Si además el último vértice

visitado es adyacente al primero, el camino es un ciclo hamiltoniano [8].

Algorithm 1: Circuito Hamiltoniano

```

1: node(1..4). start(1).
2: edge(1,2).
3: edge(2,3). edge(2,4).
4: edge(3,1). edge(3,4).
5: edge(4,1). edge(4,3).
6: { hc(V,U) } :- edge(V, U).
7: reached(V) :- hc(S, V), start(S).
8: reached(V) :- reached(U), hc(U, V).
9: :- node(V), not reached(V).
10: :- hc(V, U), hc(V, W), U != W.
11: :- hc(U, V), hc(W, V), U != W.

```

Recurriendo a la metodología planteada, lo primero que se debe de identificar en el algoritmo 1 son los hechos, los cuales podemos observar en las líneas (1-5), ya que en estas líneas se forma el grafo para el cual se quiere encontrar un camino o circuito. Es claro observar, como el grafo construido consta de 4 nodos y empieza en el nodo 1, además esta conectado como se muestra en la Figura 2.

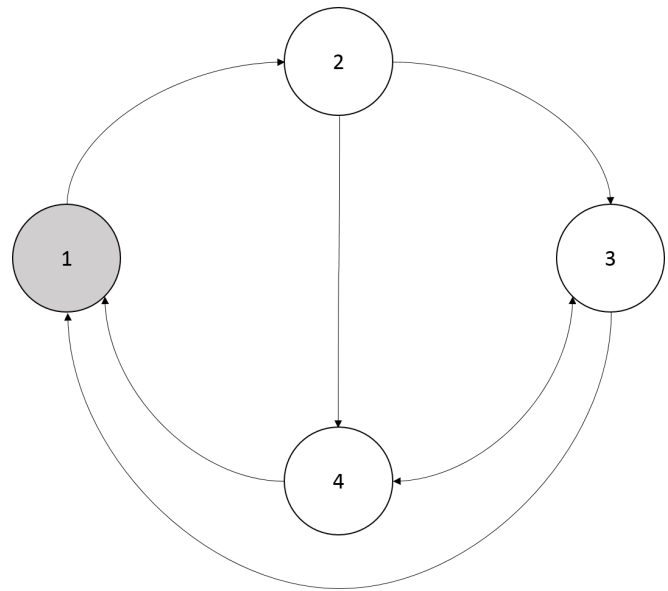


Figura 2: Grafo instantiado para obtener un camino/circuito Hamiltoniano

Las línea (6) de 1 es una regla de elección, donde las $\{ \}$ simbolizan que la regla puede ser ejecutada o no, según los modelos propuestos por ASP como un circuito hamiltoniano.

Las reglas que se encuentran en las líneas (7-8) son reglas definidas, en las que dan como resultados los vértices o nodos que pueden se alcanzados.

Finalmente, en las líneas (9-11) son restricciones, las cuales eliminarán los candidatos a ser modelos solución que no cumplan con dichas restricciones, considerando así solo aquellos modelos derivados de los hechos, que cumplan con las reglas establecidas y que no infrinjan las restricciones dictadas.

Por otra parte, existen herramientas como ASPARTIX [9] para encontrar formalizaciones aceptables en un marco argu-

mentativo. Tomando una instancia como entrada y así obtener los modelos estables.

ASPARTIX aplica la argumentación en un caso legal de 2002 conocido como “Popov v. Hayashi”, el cual en pocas palabras describe una pelota de beisbol atrapada por Popov, que después fue tacleado por otros sujetos, perdiendo la bola, Hayashi, la encuentra, la mete a su bolsa y se hace de la posesión. Mediante recopilación de hechos y testigos, se puede modelar el caso en una serie de argumentos como un gráfico mostrado en la Figura 3.

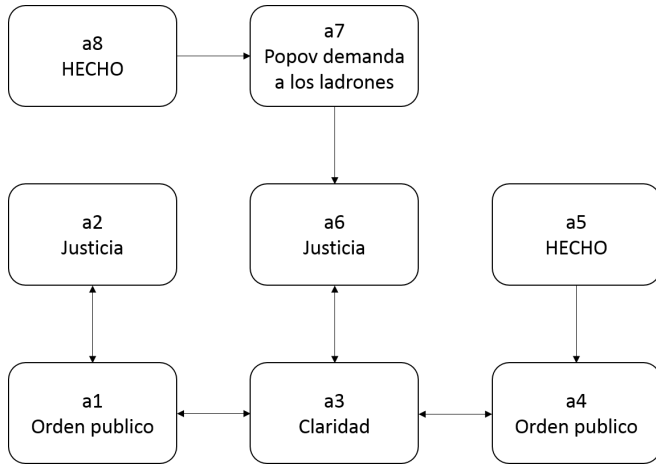


Figura 3: Gráfico de instancia del caso Popov v Hayashi

El modelado que se muestra en la Figura 3, se expresa en ASP como se muestra en la Figura 4.

```

prefex. arg(a1). arg(a2). ... arg(a8).
attacks(a1,a2). attacks(a1,a3). attacks(a2,a1). attacks(a3,a1).
attacks(a3,a4). attacks(a3,a6). attacks(a4,a3). attacks(a5,a4).
attacks(a6,a3). attacks(a7,a6). attacks(a8,a7).

```

Figura 4: Caso Popov v Hayashi en ASP

3 Resultados: Caso de estudio *Pierson v Post*

El caso de estudio *Pierson v Post* es un caso muy famoso que se le presenta a los estudiantes de Derecho en Estados Unidos [6]. Básicamente, tiene una gran similitud con el caso Popov v Hayashi solo que en vez de estar en disputa una pelota de beisbol, es por la posesión de un zorro al estarlo cazando, sin embargo intervienen más participantes, lo cual complica el modelado.

El caso *Pierson v Post*, después de los puntos expresados por cada participante se describe en 7 hechos.

- F1: Post estaba persiguiendo al zorro.
- F2: Post no había capturado ni herido al zorro (no tenía posesión del zorro).
- F3: Pierson mató al zorro para arruinar el deporte de Post (Pierson tenía intenciones maliciosas).
- F4: Los zorros matan al ganado.

- F5: Fomentar la caza reducirá el número de zorros.
- F6: Reducir el número de zorros protege el ganado de los granjeros.
- F7: Si se desalienta la caza, no se inflige sufrimiento innecesario a los animales.

Pero, en este caso participan cuatro individuos, los cuales son: Tompinks, juez del caso (T), Livingston, abogado defensor (L), Banks, diputado en contra de la caza (B), A. Tropolle, escritor a favor de la caza (A).

En la Figura se muestra la postura de cada participante en relación a los hechos, donde 1, representa que cree en el hecho, 0 no tiene argumentos suficientes para creer o no, -1 definitivamente no cree en el hecho.

Agent	F1	F2	F3	F4	F5	F6	F7
L	1	1	0	1	1	1	0
T	1	1	1	1	0	1	0
B	1	1	-1	0	0	0	1
A	1	1	1	1	-1	-1	0

Figura 5: Creencias de los participantes sobre los hechos

Después de abstraer algunos hechos del caso en cuestión, se debe de ir integrando en términos de ASP, sin embargo falta considerar otros aspectos, razón por la cual no se ha avanzado a la siguiente etapa de la metodología.

4 Discusión y conclusiones

Existen diferentes metodologías para lograr plasmar un problema de manera lógica en ASP. Sin embargo, la metodología propuesta por Fandinno et al. [7], permite de alguna manera trasladar la complicación de expresar argumentos en ASP a un paradigma similar a la programación estructurada.

El tener una metodología que te permita conocer que debes de clasificar el problema en: hechos, reglas, decisiones y restricciones ayuda a clarificar el problema. Sin embargo, expresar dichas reglas en ASP no es una tarea trivial y si hay que tener cierto nivel de conocimiento sobre ASP.

Por otra parte, el plantear problemas de manera lógica, eliminando ambigüedades hace que se obtengan resultados mejores que los que se obtendrían al manejar a conveniencia ciertos argumentos en una disputa legal.

Referencias

- [1] Tomasello Michael. *Los orígenes de la comunicación humana*. Katz Editores, 2013.
- [2] Francesca Toni and Marek Sergot. Argumentation and answer set programming. In *Logic programming, knowledge representation, and nonmonotonic reasoning*, pages 164–180. Springer, 2011.
- [3] JC Beall, Greg Restall, and Gil Sagi. Logical consequence. 2005.

- [4] Gerhard Brewka, Martin Diller, Georg Heissenberger, Thomas Linsbichler, and Stefan Woltran. Solving advanced argumentation problems with answer set programming. *Theory and Practice of Logic Programming*, 20(3):391–431, 2020.
- [5] Carole Cangioni Stav Fainshmidt, George O. White III. Legal distance, cognitive distance, and conflict resolution in international business intellectual property disputes. *Journal of International Management*, 2014.
- [6] Katie Atkinson and Trevor Bench-Capon. Legal case-based reasoning as practical reasoning. *Artificial Intelligence and Law*, 13(1):93–131, 2005.
- [7] Jorge Fandinno, Seemran Mishra, Javier Romero, and Torsten Schaub. Answer set programming made easy preliminary report.
- [8] Andrew G Thomason. Hamiltonian cycles and uniquely edge colourable graphs. In *Annals of Discrete Mathematics*, volume 3, pages 259–268. Elsevier, 1978.
- [9] Uwe Egly, Sarah Alice Gaggl, and Stefan Woltran. Aspartix: Implementing argumentation frameworks using answer-set programming. In *International Conference on Logic Programming*, pages 734–738. Springer, 2008.