



GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO DESARROLLO DE UN *TOOLKIT* PARA LA AUTOMATIZACIÓN Y OPTIMIZACIÓN DE TAREAS LEGALES BASADO EN IA

Autor: María Castilla Montes

Director: Dr. Carlos Miguel Vallez Fernández

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
Desarrollo de un *toolkit* para la automatización y optimización de tareas legales basado en
IA.

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2024/25 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.



Fdo.: María Castilla Montes

Fecha: ...15/ 06.../2025

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Carlos Miguel Vallez Fernández

Fecha: ..15../ 06../ 2025



GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO DESARROLLO DE UN *TOOLKIT* PARA LA AUTOMATIZACIÓN Y OPTIMIZACIÓN DE TAREAS LEGALES BASADO EN IA

Autor: María Castilla Montes

Director: Dr. Carlos Miguel Vallez Fernández

Madrid

“Great things are done by a series of small things brought together.”

— Vincent van Gogh

DESARROLLO DE UN TOOLKIT PARA LA AUTOMATIZACIÓN Y OPTIMIZACIÓN DE TAREAS LEGALES BASADO EN IA

Autor: Castilla Montes, María.

Director: Vallez Fernández, Carlos Miguel.

Entidad Colaboradora: ICAI-Universidad Pontificia de Comillas.

RESUMEN DEL PROYECTO

En este proyecto se desarrolla un ecosistema LegalTech modular basado en Inteligencia Artificial (“IA”) orientado a la modernización de procesos jurídicos. El sistema integra dos componentes principales: un asistente de redacción de contratos basado en generación contextual y un asistente conversacional jurídico, con el objetivo de mejorar la eficiencia y calidad de la práctica profesional.

Palabras clave: *LegalTech*, Inteligencia Artificial, Automatización Jurídica, Procesamiento de Lenguaje Natural, Asistente Legal.

1. Introducción

Este proyecto parte de una idea sencilla: en Derecho, la tecnología no debería limitarse a digitalizar PDFs ni a copiar plantillas. Debería ayudar a pensar mejor, escribir más rápido y evitar errores que cuestan tiempo y credibilidad. Pero pocas herramientas realmente lo consiguen. Muchas son demasiado rígidas, otras no entienden el lenguaje jurídico, y la mayoría no están pasadas para cómo trabaja un abogado en su día a día.

Themis nace para cubrir ese hueco. Es una plataforma *Legaltech* modular que se ha diseñado e implementado para asistir de forma real a los profesionales del Derecho en tareas que consumen tiempo y requieren precisión: redactar contratos desde cero, comprobar que los datos tienen sentido y responder a dudas jurídicas sin depender de búsquedas complejas. Frente a soluciones basadas en formularios, repositorios cerrados o software genérico de gestión documental, este proyecto propone una herramienta con capacidad de adaptarse al lenguaje jurídico, trabajar en tiempo real y mantener el control humano en todo momento. La clave está en combinar de forma coherente varias tecnologías: procesamiento de lenguaje natural, verificación semántica, anonimización y modelos de lenguaje. Con ellas se ha construido algo útil y funcional, no una prueba de concepto.

El resultado es una herramienta sencilla de usar pero técnicamente compleja por dentro. Una que no impone un modo de trabajo, sino que se adapta a cómo trabajan los profesionales. El tipo de solución que, si estuviera disponible comercialmente, sí cambiaría la forma de abordar ciertas tareas en un despacho.

2. Definición del proyecto

Themis se ha diseñado como un sistema *Legaltech* que combina tres ejes fundamentales: asistencia legal generativa, validación semántica de contratos y preservación de la privacidad desde el núcleo del flujo de trabajo. Todo el planteamiento técnico parte de esa idea: que redactar contratos o consultar dudas legales debe poder hacerse de forma más rápida, más coherente y sin comprometer ni el control del profesional ni los datos de sus clientes.

A nivel de enfoque, este proyecto se diferencia de soluciones existentes en varios aspectos:

- No depende de formularios ni plantillas rígidas. El usuario escribe con libertad y es el sistema el que se adapta a lo que redacta.
- No requiere experiencia técnica. Todo se realiza desde una interfaz simple que opera con lenguaje natural.
- No delega la privacidad a capas externas. La anonimización de los datos se realiza antes de cualquier envío del modelo de Inteligencia Artificial.

El sistema se estructura en módulos independientes, pero interconectados, lo que permite añadir funcionalidades sin romper el flujo principal. Se ha priorizado la arquitectura modular, la escalabilidad técnica y la posibilidad real de uso profesional en despachos.

Esta definición no parte de una visión teórica. Cada decisión se ha tomado desde la práctica: ¿cómo se evita que un contrato contenga un DNI mal escrito? ¿Cómo se sugiere una cláusula sin obligar a rellenar 15 campos antes? ¿cómo se evita que se expongan datos cuando se consulta a una IA? *Themis* responde a todas esas preguntas con soluciones implementadas, no con promesas.

3. Descripción de la herramienta

Themis está compuesto por dos módulos principales que trabajan de forma integrada: el Copilot de Contratos, orientado a la redacción legal asistida, y el Asistente Legal, orientado a la resolución de dudas jurídicas mediante lenguaje natural. Ambos comparten una misma arquitectura cliente-servidor, diseñada para ser modular, escalable y segura, con componentes comunes como el motor de anonimización, los validadores semánticos y el acceso a modelos de lenguaje generativo.

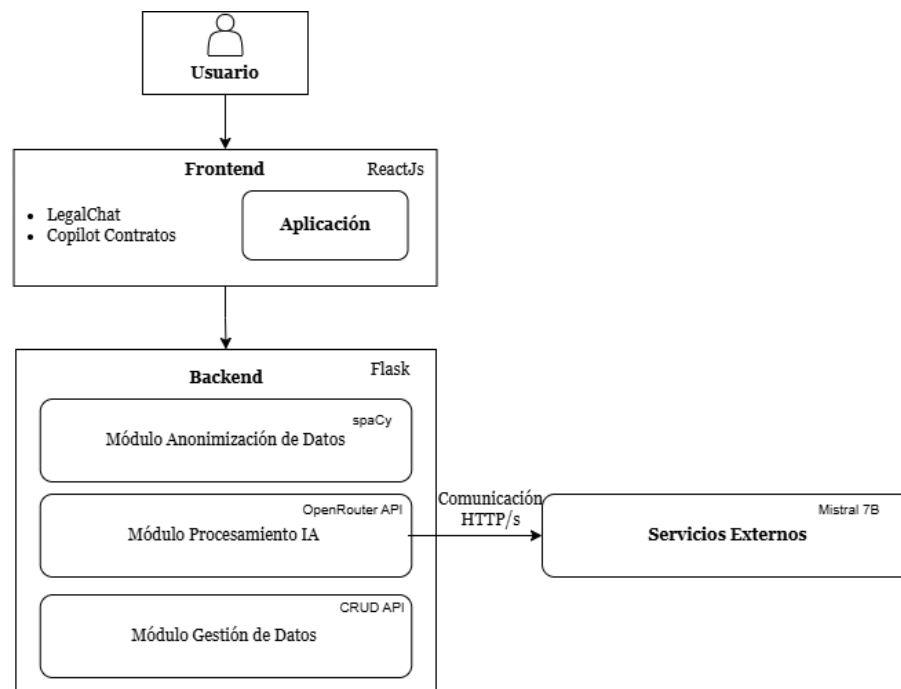


Figura 1. -Arquitectura general del ecosistema *Themis*

La estructura del sistema se organiza en tres capas claramente diferenciadas:

- En el *frontend*, desarrollado en React, el usuario interactúa directamente con los módulos. Desde aquí puede redactar contratos en lenguaje natural, visualizar errores detectados en tiempo real, consultar dudas jurídicas y descargar el documento final en formato PDF. La interfaz está pensada para respetar el flujo de trabajo del jurista: no impone plantillas, sino que permite trabajar con libertad y recibir asistencia cuando se necesita.
- El *backend*, implementado en Flask, es el núcleo del sistema. Se encarga de procesar el contenido introducido, anonimizar los datos personales antes de enviarlos al modelo de IA, validar la coherencia del documento y coordinar todas las operaciones del sistema. Este componente gestiona además las respuestas del asistente legal y la reconstrucción exacta de la información anonimizada.
- La capa de servicios externos integra dos motores fundamentales: spaCy, responsable del reconocimiento y análisis de entidades jurídicas relevantes (como nombres, direcciones, DNIs o referencias catastrales), y Mistral 7B, que genera texto jurídico contextualizado y responde a las consultas del usuario mediante la API de OpenRouter.

La Figura 1 muestra la comunicación asíncrona entre módulos, que permite mantener una experiencia fluida y eficiente. La redacción, la verificación y la consulta jurídica se gestiona sin bloqueos, con una respuesta casi inmediata, incluso cuando intervienen modelos externos de IA. Todo ocurre en el mismo entorno, sin necesidad de cambiar de aplicación, copiar contenidos o realizar procesos manuales adicionales.

4. Resultados y conclusiones

El desarrollo de *Themis* ha permitido implementar una solución *Legaltech* funcional que automatiza y mejora tareas clave en el ámbito jurídico, combinando tecnologías avanzadas en una arquitectura sólida y preparada para ser utilizada en contextos reales. Durante las pruebas del sistema, realizadas con contratos reales, consultas comunes y flujos jurídicos habituales, se han alcanzado resultados tangibles que demuestran su valor.

El Copilot de Contratos ha reducido significativamente el tiempo necesario para redactar documentos jurídicos. Esta ganancia se debe a las sugerencias contextuales generadas por el modelo, también a la eliminación de errores gracias al sistema de verificación semántica automática.

En paralelo, el proceso de anonimización previa y restitución posterior ha funcionado de forma robusta, asegurando que los datos personales no se expongan durante la generación de texto por parte de la IA. Esta capa de protección garantiza la compatibilidad del sistema con el Reglamento General de Protección de Datos (RGPD), sin interrumpir la experiencia de uso.

El Asistente Legal, por su parte, ha sido capaz de responder con coherencia a consultas jurídicas formuladas en lenguaje natural, gracias a la integración con modelos avanzados. La comunicación asíncrona con OpenRouter y el diseño modular del *backend* han permitido mantener una experiencia fluida, sin bloqueos, incluso en contextos complejos.

Estos resultados no solo demuestran la visibilidad técnica del sistema, sino también su aplicabilidad real. *Themis* no es una maqueta, ni una prueba conceptual. Es una herramienta completa que asiste al profesional en tareas reales, desde el primer borrador hasta el contrato final.

En resumen, este proyecto ha confirmado que la inteligencia artificial puede integrarse en el ámbito jurídico sin renunciar a la precisión, la privacidad ni el control. Y que, cuando se diseña con criterio y conocimiento del sector, la tecnología no reemplaza al jurista, sino que lo acompaña y amplifica su capacidad.

5. Referencias

[1] Katz, D. M., Bommarito, M. J., & Blackman, J. (2017). A general approach for predicting the behavior of the Supreme Court of the United States. PLoS ONE, 12(4), e0174698.

<https://doi.org/10.1371/journal.pone.0174698>

[2] Chandrasekaran, S. Gurusamy, S.; Kannan, R. "Advancing Legal Research Using Natural Language Processing." International Journal of Information Management, vol. 56, 2021.

<https://www.sciencedirect.com/science/article/abs/pii/S0268401220314304?via%3Dihub>

[3] Susskind, R. Tomorrow's Lawyers: An Introduction to Your Future. Oxford University Press, 2nd ed., 2019.

DESIGN AND DEVELOPMENT OF AN AI-BASED TOOLKIT FOR THE AUTOMATION AND OPTIMIZATION OF LEGAL TASKS

Author: Castilla Montes, María.

Supervisor: Vallez Fernández, Carlos Miguel.

Collaborating Entity: ICAI-Universidad Pontificia de Comillas

ABSTRACT

In this project, a modular LegalTech ecosystem based on Artificial Intelligence is developed, aimed at modernization of legal processes. The system integrates two main components: a contract drafting assistant based on contextual generation, a legal conversational assistant, all contributing to improving efficiency and quality in professional legal practice.

Keywords: LegalTech, Artificial Intelligence, Legal Automation, Natural Language Processing, Legal Assistant.

1. Introduction

The ongoing digital transformation has brought about a radical change across multiple sectors. This project starts from a simple idea: in the legal field, technology should not be limited to digitizing PDFs or copying templates. It should help professionals think more clearly, write faster, and avoid errors that cost time and credibility. Yet few tools truly achieve this. Many are too rigid, others fail to understand legal language, and most are not designed for how lawyers actually work in their day-to-day practice.

Themis was created to fill that gap. It is a modular LegalTech platform designed and developed to provide real assistance to legal professionals in tasks that are time-consuming and require precision: drafting contracts from scratch, checking that the data makes sense, and answering legal questions without relying on complex searches. Compared to solutions based on rigid forms, closed repositories, or generic document management software, this project proposes a tool capable of adapting to legal language, operating in real time, and keeping the professional in control at all times. The key lies in combining several technologies coherently: natural language processing, semantic validation, data anonymization, and language models. With these components, something truly useful and functional has been built—not a proof of concept.

The result is a tool that is simple to use, but technically complex under the hood. One that doesn't force a particular workflow but instead adapts to how professionals already operate. The kind of solution that, if available commercially, could genuinely change the way certain legal tasks are approached in a law firm.

2. Project Definition

Themis has been designed as a LegalTech system built on three core pillars: generative legal assistance, semantic contract validation, and privacy preservation embedded at the core of the workflow. The entire technical approach stems from a simple premise: drafting contracts or consulting legal questions should be faster, more coherent, and should not compromise either the professional's control or their clients' data.

From a design standpoint, this project sets itself apart from existing solutions in several key ways:

- It does not rely on rigid forms or templates. The user writes freely, and the system adapts to what is being drafted.
- It does not require technical expertise. Everything operates through a simple interface that works with natural language.
- It does not outsource privacy to external layers. Data anonymization is handled before any content is sent to the AI model.

The system is structured around independent but interconnected modules, which allows new features to be added without breaking the overall workflow. The focus has been placed on modular architecture, technical scalability, and real-world applicability in legal practice.

This definition is not theoretical. Every design decision was made from a practical perspective: how do you ensure a contract doesn't contain a mistyped ID number? How can a clause be suggested without forcing the user to fill in fifteen predefined fields? How do you prevent data from being exposed when querying an AI model? Themis answers all of these questions with implemented solutions—not with promises.

3. System description

Themis is composed of two main modules that operate in an integrated manner: the Contract Copilot, focused on assisted legal drafting, and the Legal Assistant, designed to resolve legal queries expressed in natural language. Both share a common client-server architecture, built to be modular, scalable, and secure, with shared components such as the anonymization engine, semantic validators, and access to generative language models.

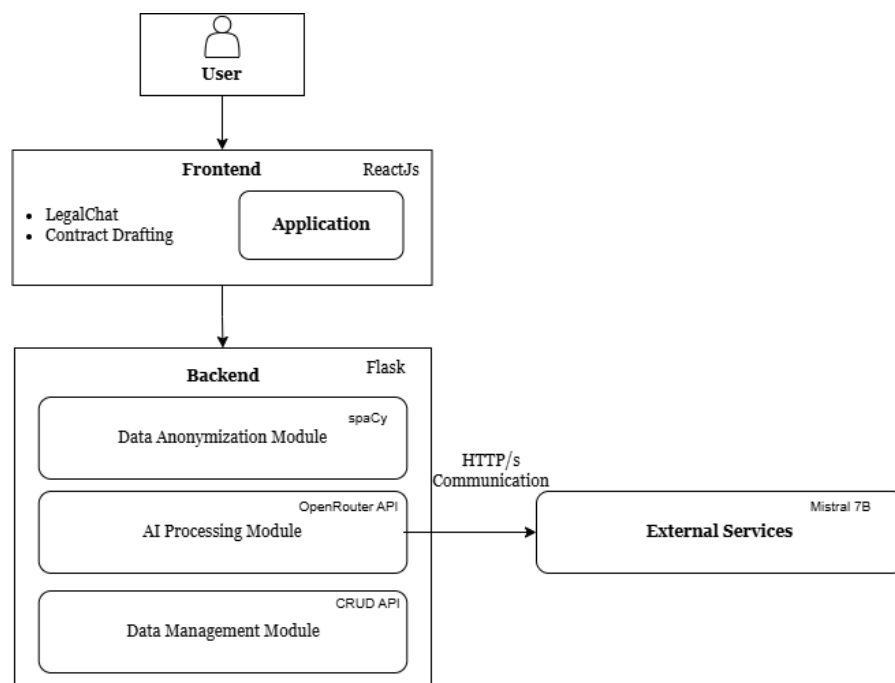


Figure 1- General Architecture of the Themis Ecosystem

The system architecture is organized into three clearly differentiated layers:

- The frontend, developed in React, is where the user interacts directly with the modules. From this interface, contracts can be drafted in natural language, detected

errors are displayed in real time, legal queries can be submitted, and the final document can be downloaded in PDF format. The interface is designed to respect the jurist's workflow: it does not enforce templates but rather allows free-form writing with intelligent assistance when needed.

- The backend, implemented in Flask, forms the core of the system. It processes the input content, anonymizes personal data before sending it to the AI model, validates the internal coherence of the document, and coordinates all system operations. It also manages the responses from the Legal Assistant and the exact reconstruction of anonymized information.
- The external services layer integrates two key engines: spaCy, responsible for identifying and analyzing relevant legal entities (such as names, addresses, national ID numbers or cadastral references), and Mistral 7B, which generates contextualized legal text and answers user queries via the OpenRouter API.

Figure 1 illustrates the asynchronous communication between modules, which ensures a fluid and efficient experience. Drafting, verification, and legal consultation are handled without interruptions, with near-instantaneous responses—even when external AI models are involved. Everything happens within the same environment, with no need to switch applications, copy content, or perform additional manual tasks.

4. Results and Conclusions

The development of *Themis* has resulted in a functional LegalTech solution that automates and enhances key tasks within the legal field, combining advanced technologies within a robust architecture ready for real-world use. During system testing—conducted with real contracts, common legal queries, and typical legal workflows—tangible results were achieved that clearly demonstrate the platform's value.

The Contract Copilot significantly reduced the time required to draft legal documents. This gain is attributed not only to the contextual suggestions generated by the model, but also to the elimination of errors thanks to the automatic semantic validation system.

In parallel, the pre-anonymization and post-reconstruction process operated reliably, ensuring that personal data was never exposed during interaction with the AI model. This layer of protection ensures the platform's compliance with the General Data Protection Regulation (GDPR), without disrupting the user experience.

The Legal Assistant was likewise able to respond coherently to legal queries expressed in natural language, thanks to its integration with advanced language models. Asynchronous communication via OpenRouter, combined with the backend's modular design, allowed for a fluid, uninterrupted experience—even in more complex contexts.

These results demonstrate not only the technical feasibility of the system, but also its practical applicability. *Themis* is not a mock-up or a proof of concept; it is a complete tool that supports legal professionals throughout the entire drafting process—from the initial outline to the final contract.

In summary, this project confirms that artificial intelligence can be integrated into the legal domain without compromising precision, privacy, or professional control. When designed with insight and sector-specific understanding, technology does not replace the legal expert—it supports and amplifies their capabilities.

5. References

- [1] Katz, D. M., Bommarito, M. J., & Blackman, J. (2017). A general approach for predicting the behavior of the Supreme Court of the United States. PLoS ONE, 12(4), e0174698.
<https://doi.org/10.1371/journal.pone.0174698>
- [2] Chandrasekaran, S. Gurusamy, S.; Kannan, R. "Advancing Legal Research Using Natural Language Processing." International Journal of Information Management, vol. 56, 2021.
<https://www.sciencedirect.com/science/article/abs/pii/S0268401220314304?via%3Dihub>
- [3] Susskind, R. Tomorrow's Lawyers: An Introduction to Your Future. Oxford University Press, 2nd ed., 2019.

Índice de la memoria

Capítulo 1. Introducción	14
1.1 Motivación del proyecto.....	15
Capítulo 2. Estado del Arte.....	16
2.1 Evolución tecnológica en el ámbito jurídico	16
2.2 Procesamiento de lenguaje natural en Derecho.....	17
2.3 Modelos generativos y su aplicación en LegalTech.....	19
2.4 Soluciones actuales en el mercado LegalTech	20
2.4.1 Plataformas Tradicionales	21
2.4.2 Proyectos emergentes y especializados.....	24
Capítulo 3. Diseño y Desarrollo de Themis.....	29
3.1 Objetivos de Themis.....	29
3.2 Nombre y Diseño de Marca: Themis.....	31
3.3 Arquitectura general del sistema	31
3.4 Flujo de datos en la plataforma	34
3.5 Descripción funcional de los módulos de Themis.....	36
3.5.1 Copilot de Contratos	36
3.5.2 Asistente Legal.....	61
Capítulo 4. Tecnologías utilizadas.....	75
4.1 Lenguajes de programación y entorno base	75
4.2 Backend y estructura del servidor	76
4.3 Frontend	77
4.4 Procesamiento del lenguaje natural	77
4.5 Validación, seguridad y control de datos	79
4.6 Infraestructura y herramientas de desarrollo	79
4.7 Instalación y dependencias clave.....	80
4.8 Bases de datos y persistencia de información	81
4.9 Selección del modelo de Lenguaje: Mistral 7B vía OpenRouter	82
Capítulo 5. Metodología y Planificación	83
5.1 Principios aplicados del enfoque Agile	83

5.2	Comunicación directa con el usuario final	84
5.3	Diagrama iterativo Ciclos de trabajo (Sprints)	85
5.3.1	<i>Sprint 1 – Definición de requisitos y backlog</i>	86
5.3.2	<i>Sprint 2 – Diseño de prototipos iniciales y validación</i>	87
5.3.3	<i>Sprint 3 – Desarrollo iterativo y pruebas semánticas</i>	87
5.3.4	<i>Sprint 4 – Ajustes, personalización y documentación</i>	88
5.4	Estimación de costes: desarrollo de un prototipo funcional open source	89
5.4.1	<i>Alcance del prototipo</i>	90
5.4.2	<i>Coste estimado del desarrollo del prototipo</i>	90
5.4.3	<i>Costes adicionales para su conversión en producto final</i>	91
Capítulo 6.	<i>Resultados Funcionales y visualización del sistema</i>	93
6.1	Introducción.....	93
6.2	Caso de uso completo: Redacción asistida de contratos.....	94
6.2.1	<i>Redacción contextual asistida</i>	95
6.2.2	<i>Verificación automática de datos</i>	97
6.2.3	<i>Informe jurídico Automatizado</i>	98
6.2.4	<i>Anonimización y procesamiento externo</i>	99
6.2.5	<i>Resultado generado y exportación a pdf</i>	101
6.3	Caso de uso: Consulta jurídica en lenguaje natural	103
6.3.1	<i>Pregunta del usuario</i>	103
6.3.2	<i>Respuesta del sistema</i>	104
6.3.1	<i>Leyes citadas</i>	106
6.4	Análisis de flujo de interacción	106
6.5	Demostración grabada	107
Capítulo 7.	<i>Conclusiones y análisis de resultados</i>	108
7.1	Qué aporta Themis al mercado actual	108
7.2	Trabajo futuro.....	111
7.1	Limitaciones y consideraciones técnicas clave	112
7.1.1	<i>Privacidad por diseño y uso de modelos externos</i>	113
7.1.2	<i>Anonimización previa y reconstrucción exacta</i>	113
7.1.3	<i>Capacidad del asistente legal para mantener contexto</i>	113
7.1.4	<i>Libertad vs. estructura en la redacción</i>	113

7.1.5 Afirmaciones categóricas y verificación legal preliminar	114
7.1.6 Uso para usuarios no expertos	114
Capítulo 8. Bibliografía.....	115
ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS	117
ANEXO II-FRAGMENTO LEGAL ALMACENADO.....	119

Índice de figuras

Figura 1- Logotipo del proyecto <i>Themis</i>	31
Figura 2-Arquitectura general de la plataforma <i>Themis</i>	33
Figura 3-Diagrama de flujo general del procesamiento de solicitudes en <i>Themis</i>	34
Figura 4-Página de inicio de <i>Themis</i> con acceso a los dos módulos principales.....	36
Figura 5-Interfaz del Copilot de Contratos con sugerencias jurídicas asistidas	37
Figura 6-Flujo completo del proceso de autocompletado adaptativo en el Copilot.	40
Figura 7-Sugerencia jurídica generada por el Copilot tras analizar el contexto.....	45
Figura 8-Verificación automática de datos sensibles	46
Figura 9- Errores detectados en la verificación de datos sensibles.	50
Figura 10-Verificación jurídica preliminar mediante modelo de lenguaje.....	51
Figura 11-Informe jurídico generado por <i>Themis</i> (contrato incompleto).	52
Figura 12-Flujo de proceso completo de renderizado y exportación de contrato	54
Figura 13-Resultado de renderizado de un contrato generado por <i>Themis</i>	55
Figura 14-Ejemplo de exportación a PDF de un contrato generado por <i>Themis</i>	57
Figura 15- Consulta respondida con normativa explicativa.	62
Figura 16-Flujo interno del procesamiento de consultas en el Asistente Legal	63
Figura 17- Obtención y vectorización de normativa jurídica.	67
Figura 18-Pregunta realizada por el usuario.....	69
Figura 19- Parte de la respuesta con explicación del problema.	70
Figura 20-Mención de las leyes utilizadas.	70
Figura 21-Parte de la respuesta generada que se basa en la mención de palabras clave.	70
Figura 22-Botón “Buscar Jurisprudencia”.....	72
Figura 23-Fragmento del informe generado.	73
Figura 24-Aviso legal incluido en el pie del documento PDF generado.....	74
Figura 25-Ciclos iterativos de desarrollo del sistema <i>Themis</i>	85
Figura 26-Contrato con errores intencionados para demostración del Copilot.	95
Figura 27-Editor de contratos con autocompletado en tiempo real.....	96
Figura 28-Panel de sugerencias contractuales en tiempo real.	96

Figura 29- Panel de errores detectados durante la redacción.	97
Figura 30- Informe jurídico automatizado con recomendaciones de mejora.	99
Figura 31-Esquema del proceso de anonimización y reconstrucción de datos sensibles. .	101
Figura 32-Botón vista previa para renderizar contrato.	102
Figura 33-Botón guardar contrato PDF para exportar el contrato.	102
Figura 34-Contrato renderizado con estructura legal clara.	103
Figura 35-Fragmento final del contrato exportado con cláusula de jurisdicción.	103
Figura 36-Selector de modo de respuesta en el asistente legal.	104
Figura 37-Consulta formulada en lenguaje natural sobre el impago del alquiler.	104
Figura 38-Respuesta explicativa del asistente legal (impago de alquiler).	105
Figura 39-Respuesta experta del asistente legal (impago de alquiler).	105
Figura 40-Listado de leyes y artículos legales citados por el asistente jurídico.	106
Figura 41-Esquema del flujo de interacción en <i>Themis</i>	107

Índice de tablas

Tabla 1-Lenguajes y entornos utilizados.	76
Tabla 2-Tecnología empleadas en el <i>backend</i>	76
Tabla 3-Tecnologías empleadas en el <i>frontend</i>	77
Tabla 4-Tecnologías de procesamiento de lenguaje natural.	78
Tabla 5-Herramientas empleadas para la detección y control de datos sensibles	79
Tabla 6-Herramientas de soporte y entorno de desarrollo.	80
Tabla 7-Tecnologías empleadas para la gestión de bases de datos.	81
Tabla 8-Principios ágiles adaptados al desarrollo del sistema <i>Themis</i>	84
Tabla 9- Distribución realista de horas dedicadas durante el desarrollo del prototipo.	91
Tabla 10-Ejemplo de diccionario de anonimización utilizado.	100
Tabla 11- Comparativa funcional entre <i>Themis</i> y soluciones LegalTech existentes.	110

Índice de Código

Código 1-Endpoint encargado de la generación de sugerencias jurídicas.....	44
Código 2-Función central de verificación semántica del contrato.. ..	49
Código 3- <i>Prompt</i> estructurado utilizado para el análisis jurídico del contrato.	52
Código 4-Endpoint de renderizado de contratos en HTML a partir de texto plano.	56
Código 5-Función de <i>frontend</i> para generación y exportación de contratos.	59
Código 6-Ejemplo de como se anonimizan los datos sensibles y como se guardan	59
Código 7-Función principal de anonimización automática en contratos	60
Código 8-Ejemplo de fragmento legal almacenado	68
Código 9-Fragmento de ley de evaluación ambiental fragmentada en formato json.	120

Índice de Ecuaciones

Ecuación 1- Modelo de generación probabilística de cláusulas.....	39
Ecuación 2- Función indicadora basada en expresión regular.	47
Ecuación 3- Distancia euclídea aplicada sobre sus embeddings.	65
Ecuación 4- Normalización L2 del vector antes de su indexación.....	66
Ecuación 5-Estimación de costes	91

Diccionario de Abreviaciones

Sigla	Significado
-------	-------------

AI	Artificial Intelligence (<i>Inteligencia Artificial</i>)
----	--

API	Application Programming Interface (<i>Interfaz de Programación de Aplicaciones</i>)
-----	---

CRUD	Create, Read, Update, Delete (<i>Operaciones básicas en bases de datos</i>)
------	---

CSS	Cascading Style Sheets (<i>Hojas de estilo en cascada</i>)
-----	--

CENDOJ	Centro de Documentación Judicial
--------	----------------------------------

DNI	Documento Nacional de Identidad
-----	---------------------------------

DOM	Document Object Model (<i>Modelo de objetos del documento</i>)
-----	--

FAISS	Facebook AI Similarity Search (<i>Índice de búsqueda vectorial</i>)
-------	---

GDPR	General Data Protection Regulation (<i>Reglamento General de Protección de Datos</i>)
------	---

HTML	HyperText Markup Language
------	---------------------------

IA	Inteligencia Artificial
----	-------------------------

JS	JavaScript
----	------------

JSON	JavaScript Object Notation
------	----------------------------

Sigla	Significado
LLM	Large Language Model (<i>Modelo de lenguaje de gran tamaño</i>)
NLP	Natural Language Processing (<i>Procesamiento de Lenguaje Natural</i>)
NER	Named Entity Recognition (<i>Reconocimiento de Entidades Nombradas</i>)
NLTK	Natural Language Toolkit (<i>librería de PLN en Python</i>)
PDF	Portable Document Format
PLN	Procesamiento de Lenguaje Natural
POST	Método HTTP para envío de datos (<i>en API REST</i>)
RAM	Random Access Memory (<i>Memoria de acceso aleatorio</i>)
RAG	Retrieval-Augmented Generation (<i>Generación aumentada por recuperación</i>)
ReLU	Rectified Linear Unit (<i>Función de activación en redes neuronales</i>)
REST	Representational State Transfer (<i>Arquitectura de APIs web</i>)
RGPD	Reglamento General de Protección de Datos (<i>versión española de GDPR</i>)
UI	User Interface (<i>Interfaz de Usuario</i>)
URL	Uniform Resource Locator (<i>Dirección web</i>)
UX	User Experience (<i>Experiencia de Usuario</i>)

Diccionario de Tecnologías Utilizadas

Tecnología/Librería	Tipo	Rol en el proyecto
Python	Lenguaje	<i>Backend</i> , scraping, NLP y lógica de negocio.
JavaScript	Lenguaje	Desarrollo del <i>frontend</i> con React.
Node.js	Entorno de ejecución	Ejecución y gestión de dependencias del <i>frontend</i> .
HTML + CSS	Lenguajes de marcado/estilo	Renderizado de documentos legales y maquetación visual.
React	Framework <i>frontend</i>	Construcción de interfaz gráfica del usuario.
jsPDF	Librería JS	Exportación de contratos a PDF desde el <i>frontend</i> .
DOMParser / Blob / FileSaver.js	Librerías JS nativas	Conversión de HTML a PDF y gestión de descargas en navegador.
Flask	Framework <i>backend</i>	API REST ligera para el servidor.
OpenRouter API	Servicio externo	Acceso a modelos de lenguaje (Mistral 7B).
spaCy	Librería NLP (Python)	Anonimización de entidades en contratos.

Tecnología/Librería	Tipo	Rol en el proyecto
re / json / os / logging	Librerías estándar Python	Validación, parsing, logging y utilidades varias.
SentenceTransformers	Librería NLP (Python)	Generación de embeddings semánticos multilingües.
FAISS	Librería de búsqueda vectorial	Recuperación eficiente por similitud semántica.
Transformers (Hugging Face)	Librería NLP general	Base para modelos como BERT; empleada indirectamente.
BeautifulSoup4	Web scraping (Python)	Extracción del BOE para generar corpus legal.
Git	Control de versiones	Trazabilidad y colaboración durante el desarrollo.
npm	Gestor de paquetes JS	Instalación de librerías <i>frontend</i> .
Visual Studio Code	Entorno de desarrollo	IDE principal usado.
Celery	Cola de tareas (Python)	Ejecución de tareas en segundo plano como notificaciones.
Authlib	Seguridad (Python)	Gestión de autenticación y autorización.
FastAPI	Framework <i>backend</i> alternativo	Mencionado en anexos como opción alternativa a Flask.

Tecnología/Librería	Tipo	Rol en el proyecto
MongoDB	Base de datos NoSQL	Persistencia flexible para corpus y contratos.

Capítulo 1. INTRODUCCIÓN

La transformación digital ha supuesto un cambio profundo en múltiples sectores, incluyendo al sector legal, tradicionalmente más conservador en la adopción de nuevas tecnologías. Aunque la evolución en este ámbito ha sido más lenta que en otros, la necesidad de optimizar flujos de trabajo, mejorar la accesibilidad y aumentar la eficiencia operativa se ha convertido en una prioridad para despachos y profesionales del Derecho.

A día de hoy, buena parte de la práctica legal cotidiana sigue desarrollándose de forma manual, desde la redacción de documentos hasta la organización de información jurídica, lo que incrementa el riesgo de errores y prolonga los tiempos de trabajo innecesariamente. Frente a esta situación, la integración de herramientas basadas en Inteligencia Artificial y Procesamiento de Lenguaje Natural ofrece una alternativa real para modernizar estos procesos de manera eficiente y segura. Como advierte Susskind (2019), la profesión jurídica “se verá inevitablemente transformada por la adopción de sistemas inteligentes que automaticen parte de sus funciones tradicionales”, lo que refuerza la necesidad de impulsar esta evolución tecnológica en el sector legal.

En este contexto surge *Themis*, un ecosistema LegalTech modular diseñado para asistir al profesional jurídico en la optimización de sus procesos habituales, mediante la automatización inteligente de la redacción documental y la resolución de consultas jurídicas en lenguaje natural, todo ello garantizando el cumplimiento normativo en materia de protección de datos personales.

Themis no es simplemente un prototipo o una maqueta conceptual, sino un sistema funcional y probado que integra tecnologías avanzadas para la automatización jurídica. Combina mecanismos robustos de anonimización automática de datos sensibles mediante técnicas de procesamiento de lenguaje natural, un modelo generativo que ofrece redacción legal contextualizada, y un sistema de exportación profesional a formatos legales estándar. Además, su desarrollo ha incluido validaciones prácticas con profesionales del Derecho,

asegurando que las funcionalidades cumplen con criterios rigurosos de calidad, coherencia y cumplimiento normativo. Estas características posicionan a *Themis* como una solución tecnológica sólida y escalable, preparada para su aplicación en entornos jurídicos reales y para futuras ampliaciones.

1.1 MOTIVACIÓN DEL PROYECTO

La motivación principal del proyecto *Themis* nace de una experiencia personal cercana al mundo jurídico combinada con un fuerte interés por la evolución tecnológica. Crecer en un entorno familiar vinculado al Derecho, a través de la profesión de mi padre como abogado, me permitió observar desde una edad temprana cómo se desarrolla el trabajo diario en este sector. Al mismo tiempo, mi interés por las nuevas tecnologías me llevó a estar en contacto continuo con las últimas innovaciones, especialmente en desarrollo web y en el campo de la Inteligencia Artificial.

Esta doble perspectiva me hizo ser consciente de la resistencia que existía en el sector jurídico frente a la adopción de herramientas tecnológicas avanzadas. En concreto, pude observar cómo incluso profesionales experimentados mostraban reticencias iniciales ante el uso de recursos como ChatGPT, a pesar de las posibilidades evidentes que ofrecen para optimizar tareas repetitivas, reducir tiempos y mejorar la gestión de conocimientos.

Frente a esta realidad, *Themis* surge con el objetivo de facilitar la incorporación de tecnologías de Inteligencia Artificial y Procesamiento de Lenguaje Natural al trabajo jurídico diario. El proyecto pretende ofrecer una solución accesible, práctica y adaptada a las necesidades reales de los profesionales del Derecho, contribuyendo así a cerrar la brecha existente entre el potencial tecnológico actual y su aplicación efectiva en el ejercicio profesional.

Capítulo 2. ESTADO DEL ARTE

2.1 EVOLUCIÓN TECNOLÓGICA EN EL ÁMBITO JURÍDICO

La evolución tecnológica en el sector jurídico ha ido, históricamente, más lenta que en otras áreas profesionales. Durante décadas, el ejercicio del Derecho se basó casi exclusivamente en el trabajo manual: consulta física de códigos y jurisprudencia, redacción de documentos en papel y gestión administrativa sin apoyo digital. No obstante, a finales del siglo XX comenzaron a surgir las primeras herramientas de apoyo tecnológico que marcarían el inicio del proceso de transformación.

La aparición de bases de datos jurídicas electrónicas (como *Lexis*, en 1970, y *Westlaw*, en 1975) supuso un hito fundamental, permitiendo la búsqueda digital de la legislación, jurisprudencia y doctrina. Estas plataformas permitieron optimizar el acceso a la información, pero se centraban exclusivamente en la recuperación documental, sin incorporar procesos de razonamiento o generación automática de contenido legal.

En paralelo, el desarrollo de soluciones de gestión de despachos, como los primeros CRM jurídicos¹ y los sistemas de gestión documental (DMS)², comenzó a introducirse en grandes firmas, mejorando la organización interna pero sin alterar la forma en que se producían los documentos legales o se realizaba la interpretación jurídica.

¹ CRM jurídico (*Customer Relationship Management* jurídico): herramienta de gestión enfocada a la organización, seguimiento y optimización de la relación con clientes dentro del sector legal.

² Sistemas de Gestión Documental (DMS): herramientas tecnológicas que permiten la captura, almacenamiento, organización y control de documentos electrónicos para facilitar su acceso, gestión y conservación a lo largo del tiempo.

La aparición de tecnologías basadas en Procesamiento de Lenguaje Natural (PLN)³ y modelos generativos a partir de la década de 2010 abrió nuevas posibilidades para transformar procesos jurídicos, permitiendo no solo la búsqueda de información, sino también la generación de textos legales coherentes y contextualizados. Como plantea Katz, Bommatito y Blackman **¡Error! No se encuentra el origen de la referencia.**, *“la Inteligencia Artificial redefine las tareas jurídicas consideradas valiosas, orientando el foco hacia el rediseño de procesos más que hacia una simple automatización”*.

Esta evolución ha permitido el surgimiento de nuevas herramientas LegalTech que no se limitan a organizar información sino que participan activamente en tareas como la predicción de sentencias o la generación de estrategias legales preliminares. La llegada de las tecnologías como GPT-3 (2020) y GPT-4 (2023), junto con modelos jurídicos especializados como Harvey o MEL, ha acelerado exponencialmente esta tendencia, marcando un punto de inflexión en la adopción de soluciones basadas en IA dentro del sistema jurídico.

2.2 PROCESAMIENTO DE LENGUAJE NATURAL EN DERECHO

El Procesamiento de Lenguaje Natural (PLN) constituye una de las áreas más relevantes de la Inteligencia Artificial aplicada al ámbito jurídico. Su objetivo es dotar a los sistemas informáticos con la capacidad de comprender, interpretar y generar lenguaje humano de manera contextualizada, permitiendo así automatizar tareas que tradicionalmente requerían intervención humana directa.

En el sector legal, el PLN ha permitido avances significativos en actividades como la búsqueda avanzada de información jurídica, el análisis automático de sentencias, la extracción de cláusulas relevantes de contratos, la clasificación documental y la generación asistida de textos legales. Estas aplicaciones no solo incrementan la eficiencia, sino que

³ Procesamiento de Lenguaje Natural (PLN): disciplina de la inteligencia artificial que estudia la interacción entre los ordenadores y el lenguaje humano, facilitando la comprensión, interpretación y generación automática de texto y habla.

también reducen la carga de trabajo repetitivo, liberando tiempo para actividades de mayor valor añadido.

Según Chandrasekaran, Gurusamy y Kannan [1], *“el PLN aplicado al Derecho facilita la transformación de la investigación jurídica tradicional en un proceso más dinámico y accesible, reduciendo los tiempos de respuesta y aumentando la precisión en la recuperación de información relevante”*.

Entre las tareas específicas donde el PLN ha demostrado mayor impacto destacan:

- **Análisis semántico de documentos legales:** mediante técnicas de reconocimiento de entidades (NER)⁴ y relaciones, es posible identificar automáticamente partes, fechas, obligaciones y excepciones dentro de contratos o resoluciones judiciales.
- **Redacción asistida de textos jurídicos:** sistemas basados en PLN pueden sugerir formulaciones jurídicas estándar en función del contexto normativo aplicable.
- **Clasificación y organización de documentos:** el PLN permite organizar automáticamente grandes volúmenes de documentación legal, facilitando su posterior localización y análisis.
- **Generación automática de resúmenes:** mediante técnicas de resumen extractivo, los sistemas pueden sintetizar contratos extensos o sentencias legales largas, agilizando su revisión.

Sin embargo, a pesar de los avances logrados, la aplicación práctica del PLN en entornos jurídicos reales presenta desafíos importantes. Entre los principales retos destacan la necesidad de adaptar los modelos a la terminología y estructura propias del ámbito jurídico, la dificultad para garantizar interpretaciones precisas ante un lenguaje ambiguo o complejo, y las exigencias en materia de protección de datos al procesar documentos sensibles.

⁴ Reconocimiento de Entidades (NER): técnica de Procesamiento de Lenguaje Natural que identifica y clasifica automáticamente elementos relevantes en un texto, como nombres propios, fechas, ubicaciones o cifras, facilitando su análisis y extracción.

Estas limitaciones explican por qué, a pesar de las capacidades tecnológicas disponibles, la adopción del PLN en el sector jurídico ha sido hasta ahora limitada a nichos o aplicaciones específicas. Proyectos como *MEL* (IIC-UAM, 2024) evidencian el esfuerzo que se está realizando por superar estas barreras mediante el entrenamiento de modelos especializados en Derecho, capaces de interactuar de manera más controlada y fiable.

2.3 MODELOS GENERATIVOS Y SU APLICACIÓN EN LEGALTECH

Los modelos generativos de lenguaje, conocidos como *Large Language Models* (LLMs⁵) han transformado profundamente el panorama tecnológico en distintos sectores, incluido el jurídico. Estos modelos, basados en arquitectura de redes neuronales profundas, son capaces de comprender contextos complejos, generar texto coherente en lenguaje natural y adaptar su producción a distintos estilos y dominios.

El punto de inflexión en la adopción masiva de LLMs se produjo con el lanzamiento de GPT-3 en 2020 por *OpenAI*, que demostró que era posible generar textos indistinguibles de los producidos por humanos en una amplia variedad de contextos. Posteriormente GPT-4 (2023) consolidó esta tendencia, mejorando la precisión, la capacidad de razonamiento y la adaptación a tareas específicas. Según Bommarito y Katz **¡Error! No se encuentra el origen de la referencia.**, “*los LLMs representan un avance cualitativo en la automatización de tareas cognitivas complejas, incluyendo aquellas relacionadas con el análisis jurídico y la generación de documentos legales*”.

En el ámbito jurídico, los modelos generativos han abierto la puerta a nuevas aplicaciones en las que destacan intentos básicos de automatizar la redacción de contratos o de proporcionar respuestas a consultas jurídicas formuladas en lenguaje natural. Sin embargo, la mayoría de estas soluciones actuales se basan en enfoques genéricos o plantillas

⁵ LLM: Modelo de Lenguaje de Gran Tamaño (*Large Language Model*), es un sistema de inteligencia artificial entrenado con grandes volúmenes de texto para comprender y generar lenguaje natural de manera contextualizada.

prediseñadas, que carecen de una adaptabilidad real al contexto jurídico específico del usuario y no garantizar un control riguroso sobre la calidad y coherencia del contenido generado.

Esta falta de personalización y de control de calidad representa una limitación relevante en los modelos generalistas aplicados al Derecho, y evidencia la necesidad de soluciones específicas que integran capacidades de generación adaptativas, verificación semántica y cumplimiento normativo.

2.4 SOLUCIONES ACTUALES EN EL MERCADO LEGALTECH

El mercado LegalTech actual presenta una amplia oferta de soluciones que buscan dar respuesta a las necesidades de eficiencia, organización y automatización en el ejercicio jurídico. En la actualidad, el mercado combina plataformas tradicionales, centradas en la recuperación de información jurídica, con nuevas aplicaciones basadas en modelos de lenguaje natural. Sin embargo, no todas estas soluciones logran adaptarse de forma efectiva a las necesidades reales de los profesionales del Derecho, ya sea por su enfoque excesivamente genérico, su dependencia de plantillas rígidas o la falta de integración con los flujos de trabajo cotidianos.

Para abordar el análisis de las soluciones actualmente disponibles, se diferenciarán dos grandes categorías. Por un lado, se revisarán las plataformas tradicionales de referencia consolidadas en el sector por su trayectoria y estabilidad. Por otro, se analizarán proyectos emergentes impulsados por tecnologías de Inteligencia Artificial, que introducen nuevos enfoques y capacidades, aunque en muchos casos todavía en fases iniciales de adopción.

Esta clasificación permitirá identificar tanto los avances alcanzados como las limitaciones persistentes en el sector, proporcionando el contexto necesario para posicionar el desarrollo de *Themis* dentro de la evolución tecnológica del sector jurídico.

2.4.1 PLATAFORMAS TRADICIONALES

2.4.1.1 LexisNexis

LexisNexis es una de las plataformas pioneras en el ámbito de la tecnología jurídica, lanzada en 1970 como una base de datos electrónica de legislación, jurisprudencia y doctrina. A lo largo de las décadas esta herramienta ha ido evolucionando hasta convertirse en un ecosistema integral que facilita la investigación jurídica y la gestión de conocimiento. Recientemente ha añadido funcionalidades asistidas por Inteligencia Artificial.

Entre sus principales servicios destaca *Lexis+*, que integra búsqueda avanzada en lenguaje natural, análisis de jurisprudencia, verificación automática de citas y sugerencias contextuales durante la redacción de documentos jurídicos. Además, *LexisNexis* ha incorporado soluciones como *Lexis Create*, un plugin para Microsoft Word que permite acceder a funciones de análisis jurídico directamente desde el entorno de trabajo del abogado.

A pesar de la amplitud de sus recursos, LexisNexis presenta ciertas limitaciones relevantes en relación con las necesidades actuales de automatización jurídica:

- **Enfoque orientado principalmente a la recuperación de información:** aunque ofrece funcionalidades de asistencia, su núcleo sigue basado en la búsqueda documental más que en la generación adaptiva de contenido.
- **Dependencia de estructuras de consultas tradicionales:** las mejoras en lenguaje natural son recientes y, en muchos casos aún una formulación estructurada para obtener resultados óptimos.
- **Limitada personalización de la generación textual:** las sugerencias o recomendaciones ofrecidas no se adaptan dinámicamente al contexto específico de cada cliente o expediente, sino que responden a patrones generales de redacción jurídica.

2.4.1.2 *Westlaw*

Westlaw, constituye una de las principales plataformas de investigación jurídica avanzada a nivel global. Lanzado en 1975, *Westlaw* no solo ofrece acceso a legislación jurisprudencia y doctrina, sino que ha consolidado su posición gracias a funcionalidades especializadas que permiten analizar y validar la información jurídica de forma más estructurada.

Entre sus herramientas más destacadas se encuentra *KeyCite*, un sistema de verificación automática de citas que permite comprobar la vigencia y autoridad de los precedentes judiciales utilizados. Asimismo, *Westlaw* incorpora funcionalidades de análisis jurisprudencial visual, que permite entender gráficamente las relaciones entre distintas resoluciones y detectar tendencias judiciales en determinados ámbitos del Derecho.

Pese a su robustez en cuanto a investigación y análisis documental, *Westlaw* presenta ciertas limitaciones en términos de automatización avanzada:

- **Foco en validación documental** más que en generación adaptativa de contenidos jurídicos.
- **Ausencia de generación textual contextualizada** adaptada a expedientes o clientes específico.
- **Entorno poco flexible** para su integración directa con sistema de gestión de flujos de trabajo jurídicos internos.

2.4.1.3 *Clio*

Clio es una plataforma de gestión integral de despachos jurídicos, orientada principalmente a la organización administrativa y operativa del ejercicio profesional. Fundada en 2008, *Clio* representa uno de los ejemplos más consolidados de software basado en la nube específicamente diseñado para el sector legal.

Su ecosistema, compuesto por soluciones como *Clio Manage* y *Clio Grow*, permite a los despachos jurídicos gestionar expedientes, organizar calendarios de audiencias, controlar tiempos de facturación y generar informes de productividad y seguimiento financiero. Además, integra funcionalidades de generación de informes de productividad y seguimiento

financiero adaptándose tanto a firmas más pequeñas como a departamentos legales de gran tamaño.

A pesar de su éxito en el ámbito de la gestión administrativa, *Clio* presenta limitaciones significativas desde la perspectiva de la automatización del contenido jurídico:

- **Enfoque centrado en la gestión operativa:** *Clio* optimiza la gestión de clientes y expedientes, pero no aborda la generación de documentos legales adaptados al contexto normativo.
- **Ausencia de procesamiento de lenguaje natural:** la plataforma carece de capacidades de interpretación semántica o de generación textual inteligente.
- **Dependencia de integraciones externas:** para funcionalidades más avanzadas de automatización documental o análisis jurídico, *Clio* requiere la integración de aplicaciones de terceros, lo que puede generar fragmentación en el flujo de trabajo.

2.4.1.4 *Contract Express*

Contract Express es una de las herramientas más reconocidas en el ámbito de la automatización documental jurídica tradicional. Su enfoque principal consiste en la generación automática de documentos legales a partir de plantillas inteligentes, que se completan mediante formularios estructurados.

El funcionamiento de *Contract Express* se basa en la creación previa de plantillas personalizadas que incorporan variables y reglas lógicas. A través de cuestionarios dinámicos, el usuario responde a una serie de preguntas y el sistema genera automáticamente el documento adaptado en función a las respuestas proporcionadas. Esta metodología permite estandarizar la producción documental, minimizar errores de redacción y mejorar la eficiencia en procesos repetitivos como la elaboración de contratos, acuerdos de confidencialidad o escrituras públicas.

No obstante, aunque *Contract Express* representa un avance respecto a la mera redacción manual de documentos, también presenta limitaciones significativas en comparación con enfoques más dinámicos basados en procesamientos de lenguaje natural:

- **Dependencia de plantillas predefinidas:** el sistema requiere una programación previa detallada de las plantillas, lo que limita la flexibilidad para adaptarse a contextos no previstos.
- **Interacción estructurada:** el proceso de generación se basa en responder formularios, lo que puede resultar insuficiente en escenarios dinámicos y poco intuitivo en situaciones jurídicas complejas o cambiantes.
- **Falta de generación adaptativa en lenguaje natural:** *Contract Express* no genera contenido jurídico libremente contextualizado, sino que ensambla fragmentos preprogramados, restringiendo la variabilidad expresiva de los documentos resultantes.

2.4.1.5 Conclusión sobre plataformas tradicionales

La revisión de las principales plataformas tradicionales de referencia evidencia su contribución al proceso de digitalización del sector jurídico, así como las mejoras alcanzadas en materia de gestión y acceso a la información. Sin embargo, su enfoque permanece vinculado a la estandarización, sin integrar plenamente capacidades de generación adaptativa o procesamiento semántico avanzado.

Además, otras herramientas orientadas a la gestión operativa, como MyCase o DocuSign, complementan funciones administrativas pero no abordan los retos de automatización contextualizada que necesita el ejercicio jurídico a día de hoy.

2.4.2 PROYECTOS EMERGENTES Y ESPECIALIZADOS

En paralelo al desarrollo de las plataformas tradicionales, los últimos años han surgido nuevos proyectos que buscan aplicar técnicas avanzadas de Inteligencia Artificial al ámbito jurídico. Estos desarrollos, impulsados por el auge de los LLMs y el procesamiento semántico avanzado, representan un cambio cualitativo respecto a las soluciones anteriores al centrarse en la generación dinámica de contenido, el análisis predictivo y la asistencia jurídica adaptativa.

A diferencia de las plataformas consolidadas, que optimizan fundamentalmente la organización de información y la recuperación documental, estos proyectos emergentes buscan explorar la automatización inteligente de tareas jurídicas complejas, proponiendo nuevas formas de interacción entre el profesional jurídico y las tecnologías emergentes.

2.4.2.1 Harvey (Allen & Overy-actualmente denominado A&O Shearman)

Harvey es una plataforma de asistencia jurídica desarrollada en colaboración con el bufete internacional *Allen & Overy* (actualmente denominado A&O Shearman) y basada en LLMs adaptados específicamente al sector legal. Presentado públicamente en 2023, *Harvey* se diseñó con el objetivo de automatizar tareas como la generación de borradores de contratos, el análisis de documentos y la elaboración de estrategias preliminares en distintos ámbitos del Derecho.

La propuesta de *Harvey* se apoya en la utilización de modelos de lenguaje personalizados, entrenados sobre bases de datos jurídicas internas y normativas específicas, con el fin de mejorar la precisión y la contextualización de las respuestas ofrecidas. Entre sus principales funcionalidades se encuentran:

- **Generación de borradores jurídicos** preliminares basados en indicaciones en lenguaje natural.
- **Análisis de documentos** para detectar inconsistencias o riesgos contractuales.
- **Asistencia en tareas de investigación jurídica** mediante consultas directas al modelo.

No obstante, a pesar de su enfoque innovador, *Harvey* presenta también ciertas limitaciones:

- Dependencia de bases de datos internas específicas: su rendimiento óptimo depende de la calidad y amplitud de los datos legales internos sobre el que ha sido entrenado, lo que puede limitar su aplicabilidad en ámbitos jurídicos diferentes o menos documentados.

- Alto coste de implementación y mantenimiento: el entrenamiento personalizado y la infraestructura necesaria suponen una barrera de entrada significativa para despachos de menor tamaño.
- Riesgo de *alucinaciones*⁶ jurídicas: aunque el modelo ha sido adaptado al lenguaje jurídico, persisten riesgos de generación de contenido incorrecto o no fundamentado si las consultas no están bien formuladas o si el modelo extrapola fuera de su dominio de conocimiento.

Harvey representa uno de los primeros intentos serios de adaptar modelos de lenguaje avanzados al entorno jurídico real. Sin embargo, su enfoque sigue estando limitado por su dependencia de infraestructuras de alta capacidad y de entrenamientos específicos costosos, abriendo la puerta a soluciones más accesibles, modulares y adaptativas.

2.4.2.2 MEL (IIC-UAM)

MEL (Modelo Especializado Legal) es una iniciativa impulsada por el Instituto de Ingeniería del Conocimiento (IIC) de la Universidad Autónoma de Madrid (UAM) que busca ofrecer una alternativa específica a los grandes modelos generativos generalistas aplicados al sector jurídico.

A diferencia de enfoques como *Harvey*, que adaptan modelos preexistentes de propósito general, *MEL* ha sido concebido desde el principio como un modelo de lenguaje especializado en Derecho. Su entrenamiento se basa en corpus jurídicos cuidadosamente seleccionados, incorporando no solo textos legales, sino también resoluciones judiciales, doctrinas y contratos, con el objetivo de garantizar una comprensión más precisa del lenguaje jurídico español.

⁶ Alucinaciones en modelos de lenguaje (LLMs): fenómenos en los que un modelo genera información incorrecta, falsa o inventada, a pesar de que su salida puede parecer coherente y plausible.

Entre sus características principales destacan:

- **Enfoque en precisión y control:** prioriza la fiabilidad de las respuestas frente a la creatividad generativa descontrolada.
- **Sistema híbrido:** combinas técnicas de procesamiento de lenguaje natural con sistemas expertos basados en reglas, buscando limitar los riesgos de alucinaciones o interpretaciones incorrectas.
- **Orientación a la asistencia en tareas específicas:** como análisis de contratos, extracción de obligaciones legales o respuestas a consultas normativas cerradas.

No obstante, *MEL* presenta también algunas limitaciones en relación con los retos de automatización integral:

- **Enfoque predominantemente reactivo:** su funcionamiento está orientado a responder consultas o analizar documentos ya existentes, sin abordar de manera activa la generación dinámica de nuevos contenidos jurídicos
- **Rigidez inherente al sistema híbrido:** aunque mejora el control, la incorporación de reglas estrictas puede limitar la adaptabilidad a contextos jurídicos cambiantes o situaciones no previstas en los entrenamientos.

MEL representa así un avance importante en la especialización de modelos jurídicos en español, orientado principalmente a la asistencia documental y al análisis de obligaciones legales.

2.4.2.3 Otros proyectos emergentes

Además de los desarrollos de mayor consolidación, como *Harvey* o *MEL*, han surgido en los últimos años otros proyectos que exploran la aplicación de tecnologías de Inteligencia Artificial al ámbito jurídico. Entre ellos se encuentra *Solvens* (Comillas-ICADE), orientado a la optimización de procesos en despachos de abogados y departamentos legales mediante soluciones de automatización y análisis de la información.

Asimismo, iniciativas impulsadas por startups tecnológicas y universidades, aunque en fases más tempranas de madurez, reflejan el creciente interés en transformar la práctica jurídica a través de nuevas herramientas digitales. Aunque su impacto práctico es aún limitado, estos proyectos contribuyen a expandir el ecosistema LegalTech y anticipan tendencias futuras de modernización del sector.

Capítulo 3. DISEÑO Y DESARROLLO DE *THEMIS*

A partir del análisis de las soluciones existentes en el mercado LegalTech y de las limitaciones detectadas en cuanto a automatización adaptativa y asistencia jurídica contextualizada surge *Themis* como una propuesta integral orientada a modernizar el flujo de trabajo jurídico.

Themis es un ecosistema modular que combina dos componentes principales: un asistente inteligente de redacción de contratos y un asistente legal conversacional basado en técnicas de Procesamiento de Lenguaje Natural. Su objetivo es proporcionar al profesional jurídico herramientas accesibles, adaptativas y centradas en la eficiencia del trabajo cotidiano, todo ello respetando los principios de privacidad, rigor normativo y calidad documental.

A lo largo de este capítulo se describirá la concepción, el diseño arquitectónico y funcional de *Themis*, así como las principales decisiones tecnológicas y los módulos que componen su estructura.

3.1 OBJETIVOS DE *THEMIS*

El objetivo principal de *Themis* es diseñar e implementar un *toolkit* modular basado en Inteligencia Artificial que facilite la automatización y optimización de procesos jurídicos, mejorando la eficiencia en la gestión documental, la búsqueda de información y la reducción de tareas repetitivas en despachos de abogados y departamentos legales de empresas. La plataforma busca integrar en un único entorno capacidades de asistencia en redacción y consulta jurídica en lenguaje natural, todo ello garantizando la protección de datos sensibles y la facilidad de uso para el profesional jurídico.

Para alcanzar este objetivo general, se establecen una serie de objetivos secundarios específicos:

Copilot de Contratos

- **Optimizar el flujo de sugerencias legales:** desarrollar un sistema de autocompletado inteligente que proponga cláusulas jurídicas relevantes en tiempo real, adaptadas al tipo de contrato y al contexto normativo, evitando la rigidez propia de plantillas estáticas tradicionales.
- **Verificación y análisis de datos sensibles:** implementar mecanismos de detección automática de datos personales (nombres, DNIs, teléfonos, direcciones...) utilizando técnicas de procesamiento de lenguaje natural, asegurando su correcta anonimización y posterior restitución sin pérdida de integridad.
- **Preservar la integridad de los datos anonimizados:** diseñar un flujo de anonimización y reconstrucción que garantice que los datos sensibles no se vean alterados, corrompidos o duplicados durante el procesamiento externo, cumpliendo así con principios de privacidad por diseño.

Asistente Legal Conversacional

- **Ofrecer consultas jurídicas fundamentadas:** desarrollar un asistente capaz de interpretar preguntas en lenguaje natural y proporcionar respuestas basadas en fundamentos jurídicos claros, evitando respuestas genéricas o inconsistentes.
- **Adaptar las respuestas al contexto conversacional:** lograr que el asistente pueda mantener la coherencia entre múltiples interacciones, adaptando las respuestas en función del historial de la conversación.
- **Mejorar la eficiencia en la búsqueda de información jurídica:** facilitar el acceso inmediato a la legislación, la jurisprudencia o la doctrina relevante mediante preguntas formuladas en lenguaje sencillo, reduciendo los tiempos de búsqueda manual.

3.2 NOMBRE Y DISEÑO DE MARCA: THEMIS



Figura 1- Logotipo del proyecto *Themis*

El nombre *Themis* se inspira en la figura de la diosa griega homónima, símbolo de la justicia, el orden y el Derecho natural. En la mitología, *Themis* representaba la encarnación de las leyes divinas y la equidad, y es tradicionalmente representada con una balanza y los ojos vendados como emblemas de imparcialidad y objetividad.

Esta elección busca reflejar la vocación del proyecto: poner la tecnología al servicio del Derecho desde una perspectiva ética, accesible y rigurosa.

En cuanto al diseño visual, la Figura 1 muestra un logotipo estilizado que representa, según la interpretación del observador, una balanza en equilibrio o una figura humana de pie. Esta ambivalencia gráfica simboliza la unión entre el Derecho (equilibrio y justicia) y el ser humano (el abogado como figura central del ejercicio jurídico), reforzando así la idea de que la tecnología debe ser un apoyo y no un sustituto del profesional jurídico.

3.3 ARQUITECTURA GENERAL DEL SISTEMA

La plataforma *Themis* se ha diseñado siguiendo el modelo arquitectónico clásico de cliente-servidor, en el que la interfaz de usuario (cliente) y el procesamiento de datos (servidor) operan de manera separada pero coordinada a través de comunicaciones seguras.

Dentro de esta estructura, *Themis* implementa además una arquitectura modular basada en el principio de separación de responsabilidades, diferenciando tres niveles funcionales principales:

- **Capa de presentación:** corresponde al *frontend* desarrollado en React, encargado de gestionar la interacción directa con el usuario. Desde esta capa se visualizan y controlan los módulos de redacción de contratos, gestión de tareas y asistente legal, enviando las solicitudes de procesamiento al servidor de manera segura.
- **Capa de negocio:** gestionada a través del *backend* desarrollado en Flask, donde reside la lógica de procesamiento. Esta capa se encarga de aplicar reglas, coordinar la anonimización de datos, estructurar las llamadas a servicios externos de generación de texto (OpenRouter) y organizar el flujo de información entre el usuario y la base de datos.
- **Capa de persistencia:** formada por una base de datos relacional que almacena de manera estructurada los datos persistentes del sistema. Actualmente, el sistema almacena únicamente registros de usuario y fragmentos legales preprocesados para su consulta. Además, *Themis* incorpora una base jurídica fragmentada, en la que diferentes normativas y fuentes legales han sido divididas en unidades de contenido más manejables (fragmentos), y mapeadas en un espacio vectorial mediante técnicas de *embeddings*⁷ semánticos. Esta estructura permite realizar consultas jurídicas no solo basadas en coincidencias literales, sino también en similitudes conceptuales, mejorando la precisión y relevancia de las respuestas ofrecidas por el Asistente Legal Conversacional.

Esta doble estructuración del modelo cliente-servidor a nivel global y separación de capas a nivel interno facilita la escalabilidad, el mantenimiento y el control riguroso del tratamiento de los datos sensibles en todas las etapas del sistema. La arquitectura general del sistema se

⁷ Los *embeddings* son representaciones vectoriales del lenguaje que permiten traducir frases o textos en números, de forma que se puedan comparar computacionalmente según su significado. En este proyecto, se utilizan para asociar consultas jurídicas formuladas en lenguaje natural con artículos legales relevantes por similitud semántica, incluso si no comparten exactamente las mismas palabras.

representa en la Figura 2, donde se muestra el flujo de interacción entre el usuario, el cliente, el servidor y los servicios externos de procesamiento.

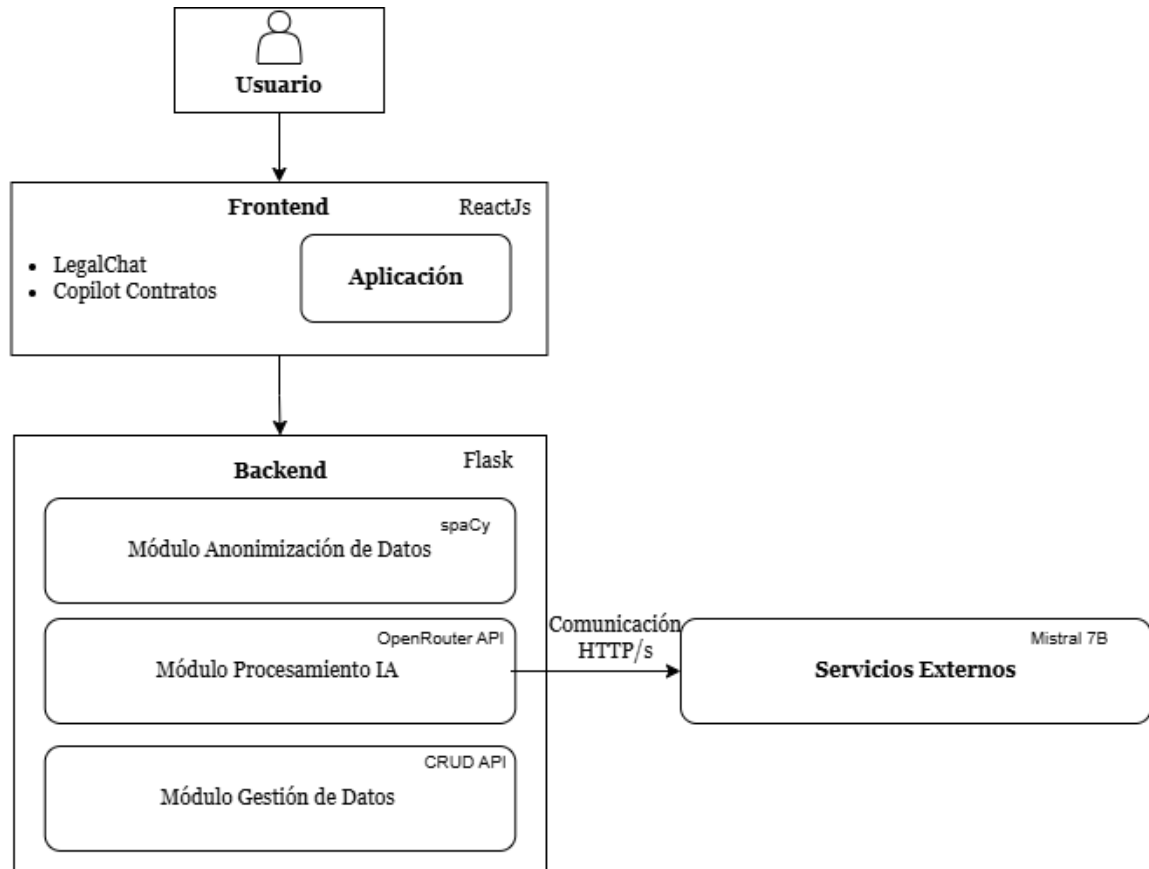


Figura 2-Arquitectura general de la plataforma *Themis*

3.4 FLUJO DE DATOS EN LA PLATAFORMA

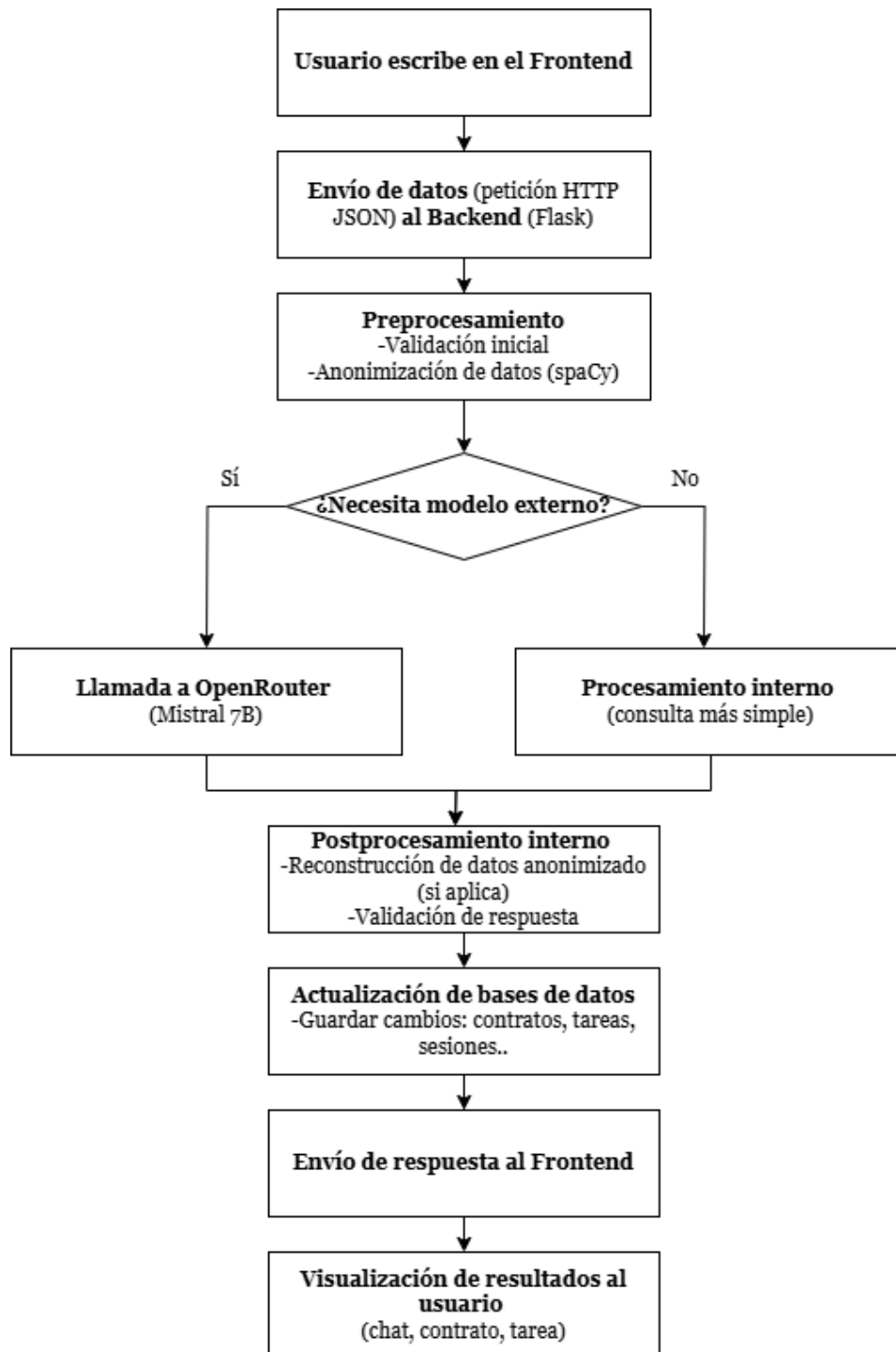


Figura 3-Diagrama de flujo general del procesamiento de solicitudes en *Themis*.

La Figura 3 muestra el funcionamiento interno de *Themis*, que sigue un flujo de procesamiento estructurado, que garantiza la trazabilidad de cada interacción y la protección de los datos sensibles. El recorrido estándar de una solicitud típica (como la redacción asistida de un contrato o una consulta legal) puede describirse en las siguientes etapas:

1. **Interacción del usuario en el *frontend*:** el usuario accede a uno de los módulos (Copilot o Asistente Legal) y realiza una acción como escribir un fragmento de contrato o plantear una pregunta jurídica.
2. **Envío de solicitud al *backend*:** la información introducida por el usuario se envía inmediatamente mediante una solicitud HTTP segura (POST o GET) al servidor *backend*. El intercambio de datos se realiza en formato JSON estructurado.
3. **Preprocesamiento en el *backend*:** al recibir la solicitud, el *backend* realiza una primera validación de formato y contenido. En el caso de textos jurídicos se aplica un proceso de anonimización automática utilizando spaCy, que identifica y sustituye datos sensibles por *tokens* neutros.
4. **Procesamiento externo o interno:** si la solicitud implica generación de texto jurídico o respuesta a consulta legal, el *backend* construye un *prompt* adecuado y lo envía a los modelos de lenguaje accesibles via OpenRouter (como Mistral 7B).
5. **Recepción y postprocesamiento:** una vez recibida la respuesta del modelo, el *backend* reconstruye el texto original (en el caso de haber anonimización previa) y verifica la coherencia de la respuesta antes de reenviarla al cliente.
6. **Actualización de la base de datos:** si la operación implica la creación o modificación de registros (nueva tarea, nuevo contrato, actualización de estado), el *backend* registra la información en la base de datos persistente, garantizando la trazabilidad completa de cada interacción.
7. **Visualización en el *frontend*:** finalmente, la respuesta procesada se envía de vuelta al cliente de React, que la presenta al usuario de forma intuitiva, actualizando el estado de la interfaz según corresponda.

3.5 DESCRIPCIÓN FUNCIONAL DE LOS MÓDULOS DE THEMIS

La plataforma *Themis* presenta una interfaz unificada desde la que el usuario puede acceder de forma directa a dos módulos principales: el Copilot de Contratos y el Asistente Legal Conversacional.



Figura 4-Página de inicio de *Themis* con acceso a los dos módulos principales.

Desde la página de inicio (Figura 4), el profesional jurídico puede seleccionar el tipo de acción que desea realizar en función de su necesidad concreta: redactar un documento legal o realizar una consulta normativa. Esta disposición modular permite una navegación rápida y centrada en tareas, facilitando la adopción de la herramienta sin necesidad de formación técnica previa.

La estructura de la interfaz ha sido diseñada siguiendo criterios de simplicidad visual y accesibilidad, presentando un conjunto limitado de opciones iniciales que derivan en funcionalidades más específicas conforme el usuario avanza en cada módulo.

3.5.1 COPILOT DE CONTRATOS

La Figura 5 muestra la interfaz principal del Copilot de Contratos, desde la que el usuario puede redactar libremente mientras recibe sugerencias generadas en tiempo real por el sistema. Este módulo se inspira en los sistemas de asistencia a la programación conocidos como "*copilots*", ampliamente utilizados en entornos de desarrollo de software. La idea

principal de estas herramientas y la clave de los mismos es el hecho de que estos sistemas no sustituyen la escritura manual del programador, sino que ofrecen sugerencias inteligentes en tiempo real que ayudan a mejorar la productividad, reducir errores y acelerar el proceso creativo.

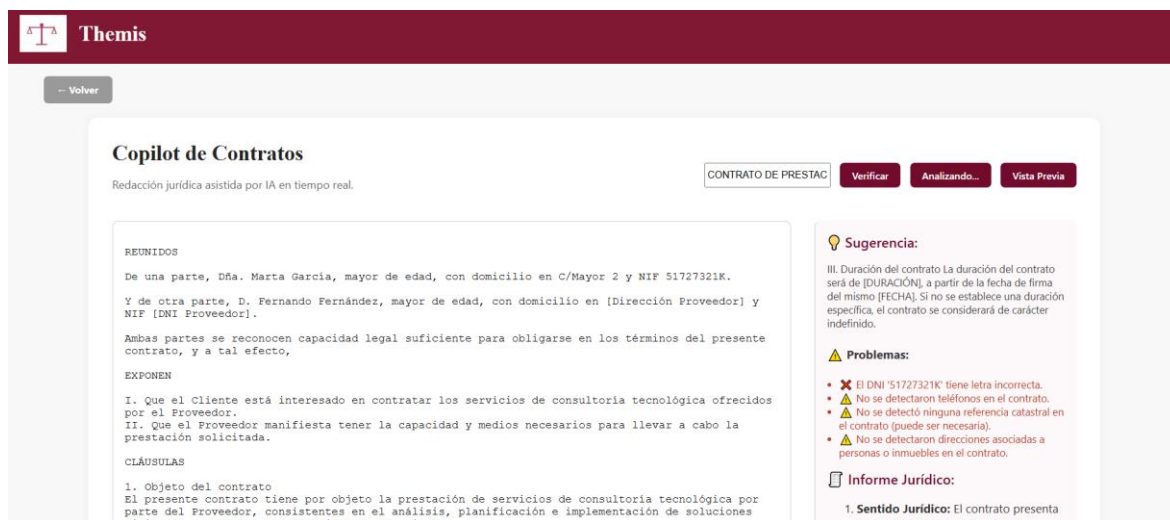


Figura 5-Interfaz del Copilot de Contratos con sugerencias jurídicas asistidas

Aplicando esta filosofía al ámbito jurídico, el Copilot de *Themis* nace con el objetivo de proporcionar asistencia adaptativa y en tiempo real en la redacción de contratos, manteniendo siempre el control creativo en manos del profesional jurídico. A diferencia de las herramientas tradicionales basadas en formularios cerrados o plantillas prediseñadas, *Themis* permite que el abogado escriba libremente su propio contrato, mientras el sistema analiza el contexto y propone sugerencias jurídicas relevantes y coherentes para completar o enriquecer el contenido.

Esta aproximación busca lograr varios objetivos fundamentales:

- **Mantener y dar valor a la flexibilidad creativa:** permitiendo que cada contrato se ajuste exactamente a las necesidades del caso concreto, sin verse encorsetado en estructuras rígidas.

- **Ahorrar tiempo de redacción:** reduciendo la necesidad de consultar manualmente modelos o referencias externas, mediante sugerencias que se integran dinámicamente en el flujo de trabajo.
- **Minimizar errores de redacción o lagunas legales:** gracias al análisis semántico y jurídico previo a la exportación del contrato.

El Copilot de Contratos pretende así potenciar la capacidad productiva del profesional, actuando como un asistente que acompaña el proceso de redacción, sugiere mejoras y contribuye a garantizar la calidad jurídica del documento, sin imponer restricciones en la estructura o el estilo del contrato redactado. Su característica clave y novedosa es la flexibilidad.

3.5.1.1 Autocompletado legal adaptativo

Como hemos mencionado, uno de los componentes más innovadores de este módulo es su sistema de autocompletado jurídico adaptativo, diseñado para acompañar la redacción en tiempo real sin imponer estructuras rígidas ni limitar la creatividad del profesional. A medida que el usuario escribe en el editor, el sistema monitoriza la entrada y aplica un retardo inteligente de un segundo. Este retardo permite identificar si el usuario ha pausado la escritura antes de lanzar una solicitud innecesaria, optimizando de esta forma el rendimiento del sistema y evitando interrupciones en el flujo natural del trabajo.

Este flujo de procesamiento se resume en la **¡Error! No se encuentra el origen de la referencia.**, que representa todas las etapas involucradas desde la entrada de texto hasta la presentación de la sugerencia final.

Desde el punto de vista probabilístico, la generación de la siguiente cláusula legal puede modelarse como un problema de predicción de secuencias condicionado al contexto. El modelo de lenguaje busca predecir la cláusula siguiente c_{t+1} , a partir de todas las anteriores $(c_1, \dots, c_t)$, generando palabra por palabra en función del historial lingüístico y semántico del contrato. Esta lógica puede representarse mediante la siguiente ecuación:

$$P(c_{t+1} | c_1, \dots, c_t) = \prod_{k=1}^m P(w_k^{(t+1)} | w_1^{(1)}, \dots, w_{k-1}^{(t+1)}, c_1, \dots, c_t)$$

Ecuación 1- Modelo de generación probabilística de cláusulas.

Esta expresión formaliza la probabilidad de generar la cláusula como secuencia de palabras, cada una condicionada no solo al historial del contrato (cláusulas anteriores), sino también al contenido parcial de la propia cláusula en construcción. De esta forma, el sistema garantiza que el texto generado mantenga coherencia jurídica tanto a nivel local (frase) como global (estructura contractual), sin recurrir a plantillas fijas.

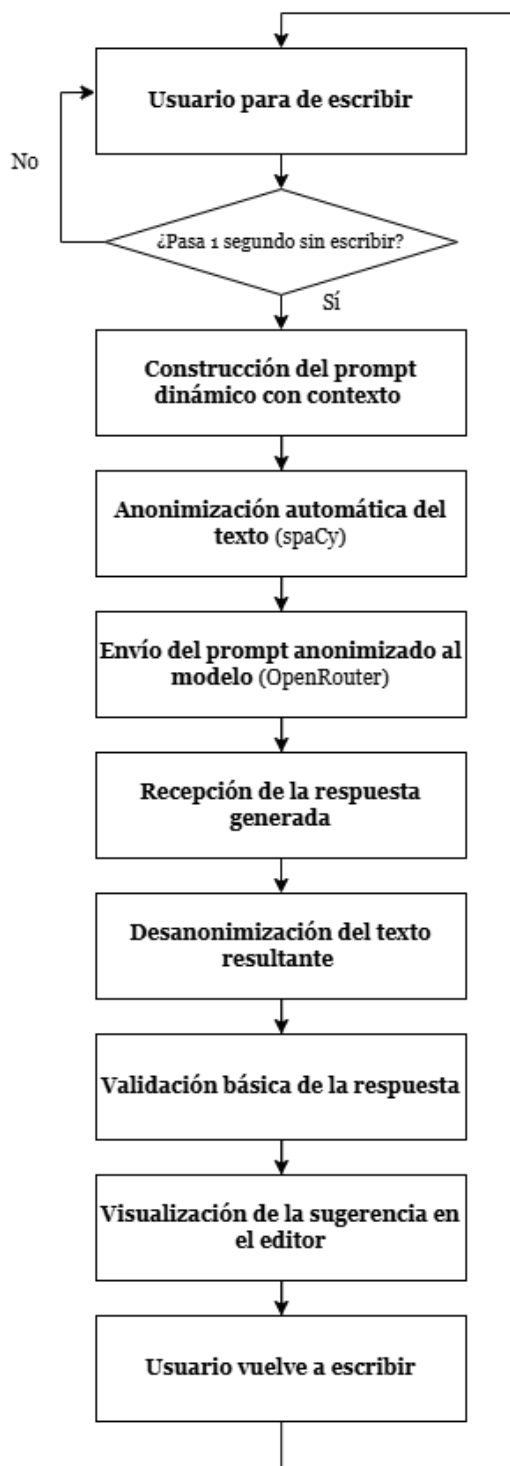


Figura 6-Flujo completo del proceso de autocompletado adaptativo en el Copilot.

Una vez que el usuario ha pausado, se construye un *prompt* dinámico que combina instrucciones predefinidas (mantener estilo jurídico formal, no inventar datos etc) con el contexto escrito hasta el momento. Antes de enviar el *prompt* al modelo de lenguaje (Mistral 7B, vía OpenRouter) el texto pasa por un sistema de anonimización automática utilizando spaCy, que detecta y sustituye entidades sensibles y datos (como nombres, DNI o direcciones).

Tras recibir la sugerencia generada por el modelo, se lleva a cabo la reconstrucción del texto original, restableciendo los datos anonimizados, y se aplican validaciones básicas de formato. Solo entonces se presenta al usuario, integrando directamente con el editor como una sugerencia visualmente distinguible. El usuario no llega a ser consciente del proceso de protección de datos ya que en la plataforma nunca se hace visible.

A continuación se muestra en el

Código 1 el *endpoint* principal encargado de gestionar la generación de sugerencias contextuales en el Copilot de Contratos. Esta función se activa desde el *frontend* tras detectar una pausa de escritura superior a un segundo, e inicia una secuencia de procesamiento estructurado:

- Se obtiene el texto actual del contrato y se anonimiza para proteger los datos sensibles.
- Se detecta la sección contractual en la que se encuentra el usuario (por ejemplo, "EXPONEN" o "CLÁUSULAS").
- En caso de que se esté en la parte de cláusulas, se identifica automáticamente la última cláusula redactada para generar la numeración adecuada (Segunda, Tercera, etc.).
- En función de la sección detectada, se incorporan al *prompt* instrucciones jurídicas específicas sobre estilo, contenido y formato esperados.
- El *prompt* final se envía al modelo Mistral 7B (vía OpenRouter) y se recibe una respuesta generada.

- Antes de mostrar la sugerencia, se aplica un proceso de reversión de la anonimización para recuperar el texto completo con sus datos estructurados.

El proceso completo se representa en el siguiente fragmento de código:

```
@autocomplete_bp.route('/trackChanges', methods=['POST'])
@handle_exceptions
def track_changes():
    data = request.get_json()
    original_text = data.get('changes', '').strip()
    contract_name = data.get('contract_name', 'Contrato General')

    if not original_text:
        return jsonify({"error": "No se recibió texto válido para cambios"}), 400

    # Anonimizar el texto antes de enviarlo a la IA
    anonymized_text, replacements = anonymize_text(original_text)

    # Detectar la sección basada en texto original (mejor para detección humana)
    section_match = re.findall(r'^[A-ZÁÉÍÓÚÑ ]{4,}$', original_text,
re.MULTILINE)
    if section_match:
        section = section_match[-1].strip()
    else:
        section = "INTRODUCCIÓN"

    # Clasificar la sección
    section_upper = section.upper()
    logical_section = (
        "REUNIDOS" if "REUNIDOS" in section_upper else
        "EXPONEN" if "EXPONEN" in section_upper else
        "CLÁUSULAS" if "CLÁUSULAS" in section_upper else
        "FIRMA FINAL" if "FIRMAN" in section_upper or "FIRMA" in section_upper
    else
        "ANEXOS" if "ANEXO" in section_upper else
        "CONDICIONES PARTICULARES" if "CONDICIONES PARTICULARES" in section_upper
    else
        "SECCIÓN PERSONALIZADA"
    )

    # Detectar última cláusula para numeración
    clausulas =
re.findall(r'(Primera|Segunda|Tercera|Cuarta|Quinta|Sexta|Séptima|Octava|Novena|D
écima)', original_text, re.IGNORECASE)
    if clausulas:
        ult_clause = clausulas[-1].capitalize()
        mapa_clausulas = {
            "Primera": "Segunda",
            "Segunda": "Tercera",
            "Tercera": "Cuarta",
            "Cuarta": "Quinta",
```

```
"Quinta": "Sexta",
"Sexta": "Séptima",
"Séptima": "Octava",
"Octava": "Novena",
"Novena": "Décima",
"Décima": "Undécima"
}
siguiente_clausula = mapa_clausulas.get(ult_clause, "Nueva cláusula")
else:
    siguiente_clausula = "Primera"

# Instrucciones específicas por sección (como definimos antes)
instructions = {
    "REUNIDOS": (
        "- Presenta los datos de las partes (nombre, dirección, NIF) usando
placeholders [NOMBRE], [DIRECCIÓN], [DNI].\n"
        "- No repitas 'REUNIDOS'. Sé formal y breve."
    ),
    "EXPONEN": (
        "- Expón motivos jurídicos sólidos, en numeración romana (II., III.,
IV.).\n"
        "- Cada motivo debe explicar razones legales (capacidad, propiedad,
voluntad, etc.).\n"
        "- No repitas 'EXPONEN'. Sé claro y formal."
    ),
    "CLÁUSULAS": (
        f"- La siguiente cláusula debe comenzar con: {siguiente_clausula}.\n"
        "- Escribe el número ordinal seguido de un título breve.\n"
        "- Después, OBLIGATORIAMENTE redacta un párrafo jurídico explicativo
de mínimo 3 frases.\n"
        "- NO se acepta solo título. Si escribes solo el título, se
considerará incompleto.\n"
        "- Cada cláusula debe abordar un tema distinto (Duración, Precio,
Entrega, Garantías, Resolución de Conflictos, etc.).\n"
        "- Usa lenguaje jurídico formal, claro y técnico."
    ),
    "FIRMA FINAL": (
        "- Redacta el cierre del contrato: lugar, fecha y espacio para
firmas.\n"
        "- No introduces nuevas cláusulas aquí."
    ),
    "ANEXOS": (
        "- Redacta una LISTA ENUMERADA de entregables, productos o documentos
concretos.\n"
        "- No escribas párrafos largos.\n"
        "- Ejemplo:\n"
        "  - Producto A\n"
        "  - Producto B\n"
        "  - Documento C\n"
        "- Sé específico, breve y técnico."
    ),
    "CONDICIONES PARTICULARES": (
        "- Redacta condiciones especiales adaptadas a este contrato.\n"
    )
}
```

```
        "- Sé preciso, jurídico y técnico."
    ),
    "SECCIÓN PERSONALIZADA": (
        "- Analiza el contenido anterior.\n"
        "- Continúa de forma coherente, formal y adaptada al tipo de
contrato.\n"
        "- Usa lenguaje jurídico claro y técnico."
    ),
)

# Montar prompt adaptado
prompt = (
    f"Estás redactando un contrato titulado '{contract_name}'.\n\n"
    f"Sección actual detectada: {section}.\n\n"
    "Instrucciones específicas:\n"
    f"{instructions.get(logical_section, instructions['SECCIÓN
PERSONALIZADA'])}\n\n"
    "Normas adicionales:\n"
    "- No inventes datos personales.\n"
    "- Usa placeholders como [NOMBRE], [DIRECCIÓN], [DNI], [FECHA],
[IMPORTE].\n"
    "- Escribe en español formal, jurídico y técnico.\n"
    "- Si no hay nada relevante que continuar, responde 'No sugerir
nada'.\n\n"
    f"Texto actual del contrato:\n{anonymized_text}\n\n"
    "Continúa el contrato aquí:"
)

# Llamada a la IA
response = client.chat.completions.create(
    model="mistralai/mistral-7b-instruct",
    messages=[{"role": "user", "content": prompt}],
    max_tokens=100,
    temperature=0.2
)

anonymized_completion = response.choices[0].message.content.strip()

# Revertir reemplazos para el usuario
final_completion = revert_replacements(anonymized_completion, replacements)

return jsonify({"autocomplete": final_completion}), 200
```

Código 1-Endpoint encargado de la generación de sugerencias jurídicas..

La Figura 7 muestra un ejemplo de sugerencia generada por el Copilot tras la redacción de las primeras cláusulas de un contrato de prestación de servicios. En este caso, el sistema propone una cláusula de precio con campos personalizables como [IMPORTE], que permiten al usuario ajustar rápidamente los valores sin comprometer la estructura jurídica.

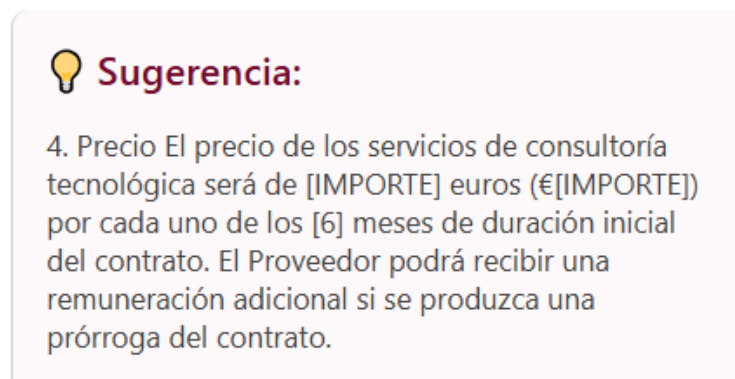


Figura 7-Sugerencia jurídica generada por el Copilot tras analizar el contexto.

Este tipo de sugerencia contextual es coherente con el contenido anterior, respeta el estilo legal esperado y demuestra la capacidad del sistema para generar texto especializado sin recurrir a plantillas predefinidas.

3.5.1.2 Verificación de datos sensibles

El módulo Copilot de Contratos integra una funcionalidad de verificación automática que permite al usuario validar los datos sensibles presentes en el contrato. Esta verificación semántica no solo busca errores de formato sino también inconsistencias internas y duplicidades, mejorando significativamente la calidad del documento antes de su cierre.

El proceso comienza cuando el usuario pulsa el botón “Verificar” que activa un endpoint específico en el *backend* encargado de analizar el texto completo del contrato. Utilizando expresiones regulares combinadas con procesamiento de lenguaje natural mediante spaCy, el sistema detecta y evalúa entidades y datos clave como DNI, números de teléfono, referencias catastrales y direcciones.

Las validaciones incluyen:

- Formato y validez estructural del DNI (8 cifras + letra correcta)
- Detección de duplicados en identificadores personales.
- Presencia o ausencia de teléfonos, direcciones o referencias catastrales.
- Coherencia entre secciones donde aparece la misma información.

El flujo interno de este sistema se resume en la Figura 8:

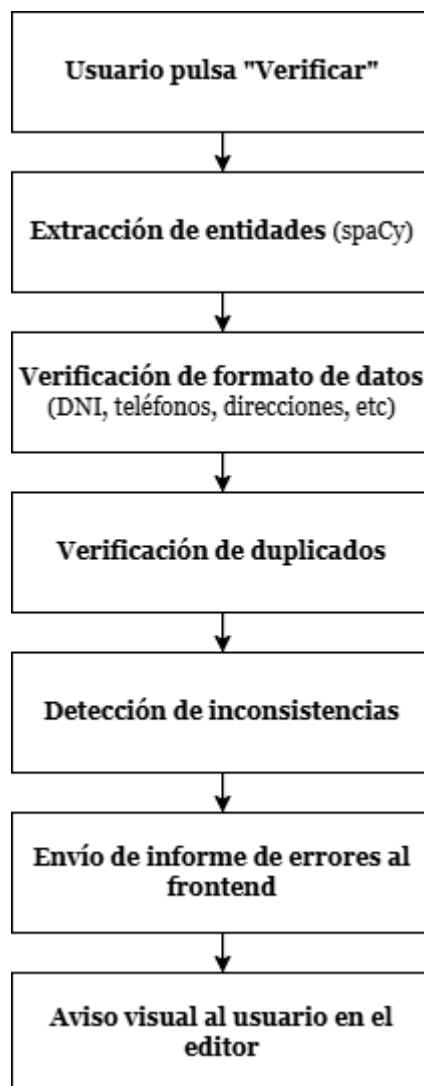


Figura 8-Verificación automática de datos sensibles

Para ejecutar el proceso completo de verificación de datos sensibles, se ha diseñado una función

central llamada `verify_contract_data (`

Código 1), que analiza el texto contractual en múltiples dimensiones: identificación de patrones, validación de formato, control de coherencia interna y emisión de advertencias estructuradas.

Entre estas validaciones, una de las más fundamentadas es la verificación de formato mediante expresiones regulares, que actúan como funciones booleanas capaces de determinar si un campo cumple con la estructura esperada. Esta lógica se puede representar de forma matemática como:

$$\mathbb{I}(x) = 1 \text{ si } x \text{ cumple el patrón esperado}$$

$$\mathbb{I}(x) = 0 \text{ en otro caso}$$

Ecuación 2- Función indicadora basada en expresión regular.

Esta abstracción permite formalizar el funcionamiento de los validadores implementados en Python, especialmente cuando se utilizan patrones para datos sensibles.

La función escanea de forma secuencial la información y los elementos más comunes y críticos en la documentación legal: DNIs, teléfonos, referencias catastrales y direcciones, y devuelve un informe clasificado de errores e inconsistencias. La validación de DNIs incluye tanto el formato como la letra, y permite identificar errores comunes como duplicidades o letras incorrectas. Los resultados se agrupan en listas de errores (*issues*) y advertencias (*warnings*) que son posteriormente devueltas al *frontend*.

El siguiente fragmento recoge el código que implementa este flujo completo de verificación:

```
import re
from collections import defaultdict

# Función para validar letra correcta de DNI
def validar_dni_letra(dni):
    letras = "TRWAGMYFPDXBNJZSQVHLCKE"
    try:
        numero = int(dni[:-1])
        letra_correcta = letras[numero % 23]
        return dni[-1].upper() == letra_correcta
    except:
        return False
```

```
# Función para limpiar texto antes de analizar
def limpiar_texto(text):
    text = re.sub(r'N\.\I\.\F\.\n\.\i\.\f\.\Número|número|nº|número de
identificación|identificación', '', text, flags=re.IGNORECASE)
    text = text.replace(".", " ").replace(",", " ").replace(":", " ")
    text = text.replace(";", " ")
    return text

def verify_contract_data(contract_text):
    issues = []
    warnings = []

    # Paso 1: Limpiar texto
    clean_text = limpiar_texto(contract_text)

    # Paso 2: Escanear todo el texto
    posibles_dnis = re.findall(r'\b\d{7,8}[A-Za-z]\b', clean_text)
    posibles_telefonos = re.findall(r'\b\d{8}\b|\b7\d{8}\b', clean_text)
    posibles_referencias = re.findall(r'\b\d{14}\b', clean_text)
    posibles_direcciones = re.findall(r'\b(calle|avenida|plaza|camino|paseo)\b',
clean_text, re.IGNORECASE)

    # Paso 3: Validar DNIs
    if not posibles_dnis:
        issues.append("✗ No se detectaron DNIs válidos en el contrato.")
    else:
        dnis_correctos = []
        dnis_incorrectos = []
        for dni in posibles_dnis:
            if not re.fullmatch(r'\d{8}[A-Za-z]', dni):
                issues.append(f"✗ El DNI '{dni}' tiene formato inválido (deben
ser exactamente 8 cifras + 1 letra).")
                dnis_incorrectos.append(dni)
            else:
                dnis_correctos.append(dni)

        # Validar letras de DNIs correctos
        for dni in dnis_correctos:
            if not validar_dni_letra(dni):
                issues.append(f"✗ El DNI '{dni}' tiene letra incorrecta.")

        # Detectar duplicidad de DNIs
        dnis_repetidos = defaultdict(int)
        for dni in dnis_correctos:
            dnis_repetidos[dni] += 1
        for dni, count in dnis_repetidos.items():
            if count > 1:
                issues.append(f"✗ El DNI '{dni}' aparece duplicado en el
contrato (potencial conflicto de identidad).")

    # Paso 4: Validar teléfonos
```

```
if not posibles_telefonos:
    warnings.append("⚠ No se detectaron teléfonos en el contrato.")
else:
    telefonos_repetidos = defaultdict(int)
    for tel in posibles_telefonos:
        telefonos_repetidos[tel] += 1
    for tel, count in telefonos_repetidos.items():
        if count > 1:
            warnings.append(f"⚠ El teléfono '{tel}' está asociado a
múltiples personas o se repite.")

# Paso 5: Validar referencias catastrales
if not posibles_referencias:
    warnings.append("⚠ No se detectó ninguna referencia catastral en el
contrato (puede ser necesaria).")
else:
    referencias_repetidas = defaultdict(int)
    for ref in posibles_referencias:
        referencias_repetidas[ref] += 1
    for ref, count in referencias_repetidas.items():
        if count > 1:
            issues.append(f"❌ La referencia catastral '{ref}' aparece
asignada a múltiples propiedades distintas.")

# Paso 6: Validar direcciones
if not posibles_direcciones:
    warnings.append("⚠ No se detectaron direcciones asociadas a personas o
inmuebles en el contrato.")

# Resultado final
if not issues and not warnings:
    return ["✅ No se detectaron problemas en los datos del contrato."]
else:
    return issues + warnings
```

Código 2-Función central de verificación semántica del contrato..

Finalmente, los resultados del análisis se devuelven al *frontend* en forma de lista de incidencias y advertencias, que se visualizan como una alerta en el entorno del editor. La Figura 9 muestra un ejemplo real de este informe, donde se detectan errores de formato y campos ausentes en el texto contractual.

 Problemas:

-  El DNI '51727321K' tiene letra incorrecta.
-  No se detectaron teléfonos en el contrato.
-  No se detectó ninguna referencia catastral en el contrato (puede ser necesaria).
-  No se detectaron direcciones asociadas a personas o inmuebles en el contrato.

Figura 9- Errores detectados en la verificación de datos sensibles.

3.5.1.3 Verificación jurídica preliminar

Más allá del análisis de datos sensibles, *Themis* incorpora un sistema de verificación jurídica preliminar diseñado para revisar el contenido del contrato desde una perspectiva legal estructural. Esta función no busca detectar errores gramaticales o estilísticos, sino identificar posibles lagunas legales o incoherencias jurídicas que comprometen la validez o la claridad del acuerdo.

El sistema se activa mediante una petición directa desde el *frontend*, y ejecuta una llamada a Mistral 7B con un *prompt* adaptado al tipo de contrato. Antes de enviar el contenido, se aplica un proceso de anonimización automática que garantiza la privacidad del texto analizado.

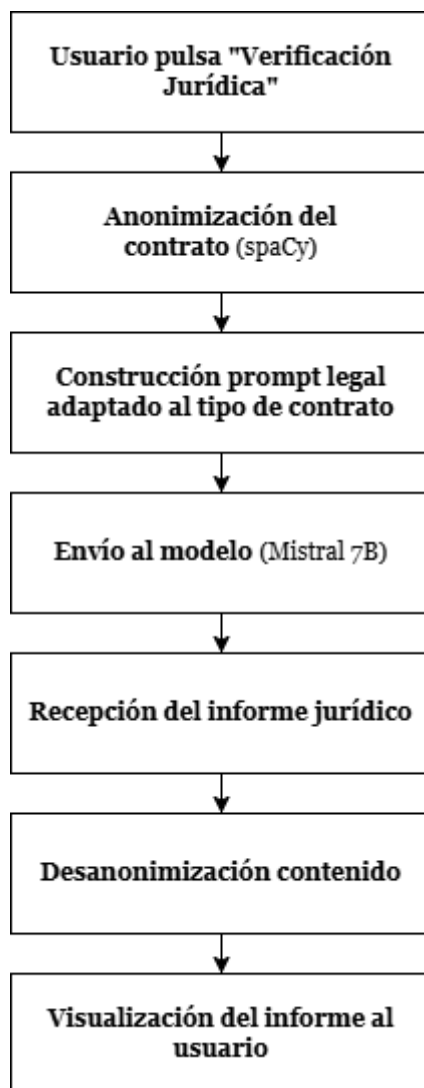


Figura 10-Verificación jurídica preliminar mediante modelo de lenguaje.

El modelo recibe instrucciones específicas para generar un informe estructurado dividido en tres secciones:

- Sentido Jurídico: Evalúa si el contrato tiene coherencia global y propósito legal comprensible.
- Faltan Cláusulas Esenciales: Enumera las secciones imprescindibles que no están presentes (por ejemplo, precio, duración, cancelación...).
- Incoherencias o Riesgos: Señala ambigüedades, errores formales o riesgos potenciales para las partes.

El flujo completo del proceso se muestra en la Figura 10. Este análisis, si bien preliminar, permite reforzar la calidad legal del contrato antes de su cierre o exportación, actuando como una segunda capa de revisión para el usuario. Un ejemplo de resultado se muestra en la Figura 11.

 **Informe Jurídico:**

1. Sentido Jurídico: El contrato presenta un sentido jurídico general, ya que se establece un acuerdo entre dos partes (EXPONEN y Proveedor) para que la primera contrate los servicios de consultoría tecnológica de la segunda.

2. Faltan Cláusulas Esenciales:

- Precio y forma de pago: No se especifica el precio a pagar por los servicios de consultoría tecnológica ni la forma de pago.
- Términos de pago: No se especifica cuándo y cómo se pagarán las cuotas o pagos parciales.

Figura 11-Informe jurídico generado por Themis (contrato incompleto).

A nivel de *backend*, el sistema se basa en un *endpoint* específico (`/legalCheck`) que anonimiza el contenido, genera el *prompt*, lanza la verificación jurídica y reconstruye el texto final con los datos reales si fuera necesario. El siguiente fragmento muestra la solicitud enviada al modelo de lenguaje:

```
prompt = (  
    f"Eres un abogado especializado en contratos del tipo '{contract_type}'.\n"  
    f"Analiza de forma profesional y organizada el siguiente contrato.\n\n"  
    f"CONTRATO:\n{anonymized_content}\n\n"  
    f"Responde obligatoriamente siguiendo esta estructura:\n"  
    f"1. **Sentido Jurídico:** ...\n"  
    f"2. **Faltan Cláusulas Esenciales:** ...\n"  
    f"3. **Incoherencias Legales o Riesgos:** ...\n"  
)
```

Código 3-Prompt estructurado utilizado para el análisis jurídico del contrato.

Este diseño orientado garantiza que la respuesta generada sea clara, precisa, legalmente coherente y segmentada, facilitando su interpretación directa por parte del profesional.

3.5.1.4 Renderizado y exportación del contrato

Para optimizar el flujo de trabajo, *Themis* dispone de un módulo de renderizado en tiempo real y exportación automática de contratos en PDF, facilitando la revisión y entrega final del documento.

Este proceso se divide en dos fases diferenciadas: primero, el renderizado dinámico del contrato en HTML a partir del texto plano redactado; y, posteriormente, la exportación estructurada a un documento PDF profesional listo para firmar o archivar.

A nivel técnico, el flujo general de trabajo se representa en la Figura 12.

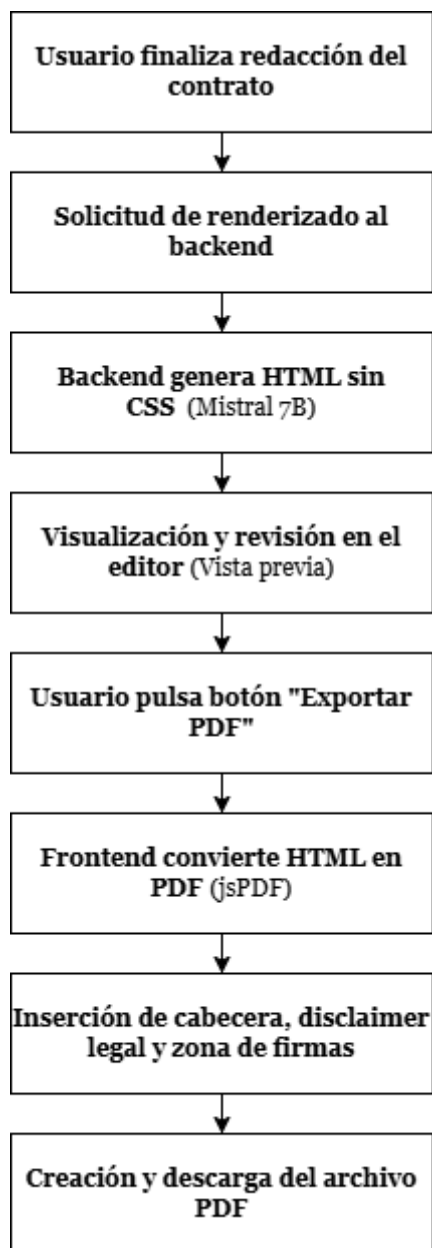


Figura 12-Flujo de proceso completo de renderizado y exportación de contrato

3.5.1.4.1 Renderizado dinámico en HTML

El proceso de renderizado comienza cuando el usuario decide visualizar su contrato antes de exportarlo. El contenido redactado es enviado al *backend*, donde se utiliza Mistral 7B para

formatearlo en un HTML limpio y estructurado propio de un documento legal. Este proceso se muestra en la Figura 13.

Copilot de Contratos

Redacción jurídica asistida por IA en tiempo real.

REUNIDOS

De una parte, D./Dña. María Castilla, mayor de edad, con domicilio en C/ Mayor 1 y NIF 51727321K.

Y de otra parte, D./Dña. Francisco Pérez, mayor de edad, con domicilio en C/Alberto Aguilera 2 y NIF 51727320K.

Ambas partes se reconocen capacidad legal suficiente para obligarse en los términos del presente contrato, y a tal efecto,

EXPONEN

I. Que el Cliente está interesado en contratar los servicios de consultoría tecnológica ofrecidos por el Proveedor.

II. Que el Proveedor manifiesta tener la capacidad y medios necesarios para llevar a cabo la prestación solicitada.

CLÁUSULAS

Figura 13-Resultado de renderizado de un contrato generado por *Themis*

El *prompt* enviado al modelo indica expresamente que el contenido debe estructurarse utilizando etiquetas `<strong>` para títulos de cláusulas y `<p>` para párrafos, evitando la introducción de CSS o alteraciones estética no deseadas. Se busca un documento riguroso.

El siguiente fragmento muestra el *endpoint* encargado de este renderizado:

```
@render_bp.route("/renderContract", methods=["POST"])
@handle_exceptions
def render_contract():
    data = request.get_json()
    contract_text = data.get("content", "").strip()

    if not contract_text:
        return jsonify({"error": "El contenido está vacío"}), 400

    prompt = (
        "Recibes un contrato en texto plano. "
        "Devuélvelo formateado en HTML limpio, usando <strong> para los títulos "
        "de cláusulas importantes "
```

```
"y <p> para cada párrafo. No inventes contenido adicional. No pongas
CSS.\n\n"
    f"Contrato:\n{contract_text}"
)

response = client.chat.completions.create(
    model="mistralai/mistral-7b-instruct",
    messages=[{"role": "user", "content": prompt}],
    max_tokens=4096,
    temperature=0.2
)

rendered_html = response.choices[0].message.content.strip()
return jsonify({"rendered_html": rendered_html}), 200
```

Código 4-Endpoint de renderizado de contratos en HTML a partir de texto plano.

3.5.1.5 Exportación final a PDF

Una vez el contrato es renderizado es revisado por el usuario. Cuando se verifica que todo está correcto, el contenido HTML es procesado en el *frontend* mediante la librería `jsPDF`. Se analizan los nodos del documento y se organizan en un archivo PDF personalizado que incluye:

- Título del contrato centrado en negrita.
- Introducción de un aviso legal dinámico según el tipo de contrato.
- Texto completo estructurado en párrafos.
- Inclusión automática de espacios reservados para firmas de las partes.

Este flujo garantiza que el documento mantenga una estructura profesional y sea inmediatamente utilizable en un entorno jurídico. Un ejemplo de documento es la Figura 14.

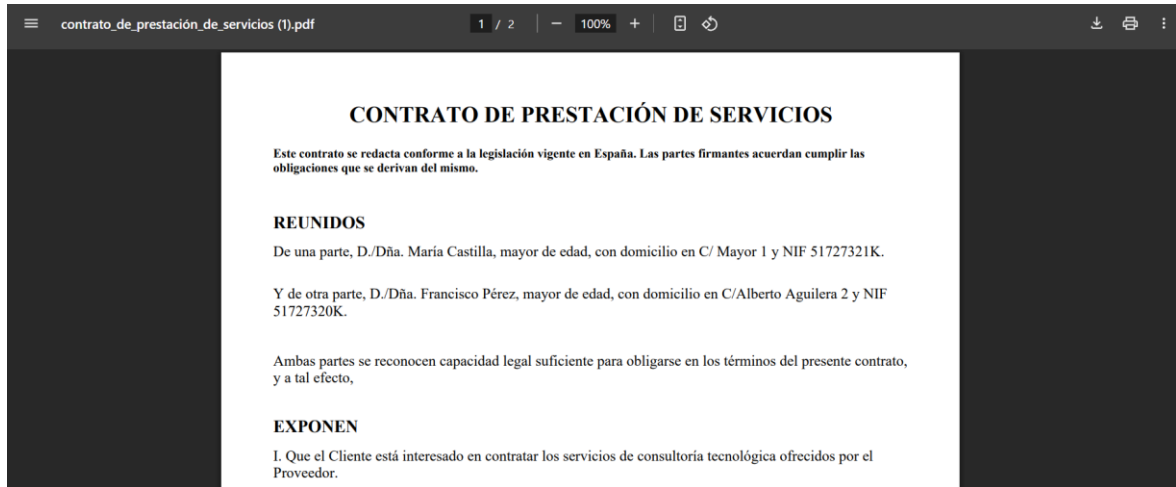


Figura 14-Ejemplo de exportación a PDF de un contrato generado por *Themis*

El proceso de generación de PDF está implementado en el cliente mediante la siguiente lógica:

```
const handleSaveRenderedContractPro = () => {
  if (!renderedContract.trim()) {
    alert("El contrato está vacío.");
    return;
  }

  const doc = new jsPDF();
  doc.setFont("times", "normal");

  // Título principal
  doc.setFontSize(18);
  doc.setFont("times", "bold");
  doc.text(contractName.toUpperCase(), 105, 20, { align: "center" });

  // Disclaimer legal
  doc.setFontSize(10);
  const disclaimer = getDisclaimerByContractName(contractName);
  const disclaimerLines = doc.splitTextToSize(disclaimer, 180);
  doc.text(disclaimerLines, 15, 30);

  // Crear contenedor DOM virtual
  const container = document.createElement("div");
  container.innerHTML = renderedContract;

  let currentY = 50;

  function processNode(node) {
    if (currentY > 270) {
      doc.addPage();
    }
  }
```

```
currentY = 20;
}

if (node.nodeType === 3) { // Text node
  const text = node.nodeValue.trim();
  if (text) {
    doc.setFontSize(12);
    doc.setFont("times", "normal");
    const lines = doc.splitTextToSize(text, 180);
    doc.text(lines, 15, currentY);
    currentY += lines.length * 7;
  }
} else if (node.tagName === "STRONG") {
  const text = node.innerText.trim();
  if (text) {
    doc.setFontSize(14);
    doc.setFont("times", "bold");
    const lines = doc.splitTextToSize(text, 180);
    doc.text(lines, 15, currentY);
    currentY += lines.length * 8;
  }
} else if (node.tagName === "P" || node.tagName === "DIV") {
  for (const child of node.childNodes) {
    processNode(child);
  }
  currentY += 5; // Espacio extra tras cada párrafo
} else {
  for (const child of node.childNodes) {
    processNode(child);
  }
}

for (const child of container.childNodes) {
  processNode(child);
}

// Firmas
if (currentY + 30 > 270) {
  doc.addPage();
  currentY = 30;
}

doc.setFont("times", "normal");
doc.setFontSize(12);
doc.line(30, currentY + 20, 80, currentY + 20);
doc.text("Firma del Comprador", 30, currentY + 25);

doc.line(130, currentY + 20, 180, currentY + 20);
doc.text("Firma del Vendedor", 130, currentY + 25);

const filename = contractName
  ? `${contractName.replace(/\s+/g, "_").toLowerCase()}.pdf`
```

```
: "contratocopilotthemis.pdf";  
  
doc.save(filename);  
};
```

Código 5-Función de *frontend* para generación y exportación de contratos.

3.5.1.6 Sistema de anonimización de datos sensibles

Como parte de las medidas de protección de datos sensibles implementadas en *Themis*, el sistema incorpora un mecanismo de anonimización automática de contenido antes de enviar cualquier fragmento contractual a modelos externos de procesamiento de lenguaje natural.

Este proceso se basa en técnicas de procesamiento de entidades nombradas (*Named Entity Recognition*, NER) mediante el modelo spaCy, complementadas por patrones regulares para datos estructurados (DNIs, teléfonos, etc). Al detectar una entidad sensible, el sistema la sustituye temporalmente por un marcador anónimo estructurado, garantizando que el contenido procesado externamente no incluya información crítica de forma directa.

El flujo de trabajo de anonimización puede resumirse en las siguientes etapas:

- Identificación de entidades sensibles (personas, localizaciones, organizaciones) mediante modelo NLP.
- Detección de patrones explícitos (DNI, teléfonos) mediante expresiones regulares.
- Reemplazo de las entidades detectadas por claves anónimas del tipo <PER0>, <DNI1>, <TELEFONO2>, preservando la estructura del contrato.

Al recibir la respuesta generada, se aplica un proceso inverso de reconstrucción segura (`revert_replacements`) que restituye los datos originales respetando la semántica del texto.

El Código 6 es un ejemplo de como se transforma un texto a la hora de ser enviado.

```
Texto original: El cliente María Castilla Montes con DNI 51727321K y domicilio en  
Mayor 2  
Texto anonimizado: ('El cliente <PER0> con DNI <DNI2> y domicilio en <LOC1> 2',  
{ '<PER0>': 'María Castilla Montes', '<LOC1>': 'Mayor', '<DNI2>': '51727321K'})
```

Código 6-Ejemplo de como se anonimizan los datos sensibles y como se guardan

A continuación se muestra el fragmento principal de código utilizado para el proceso de anonimización:

```
def anonymize_text(text):
    replacements = {}
    doc = nlp(text)

    # Reemplazar entidades de personas, localizaciones y organizaciones
    for ent in doc.ents:
        if ent.label_ in ["PER", "LOC", "ORG"]:
            key = f"<{ent.label_}{len(replacements)}>"
            replacements[key] = ent.text
            text = text.replace(ent.text, key)

    # Reemplazar DNIs
    text = replace_pattern(text, r'\b\d{8}[A-Z]\b', "DNI", replacements)

    # Reemplazar teléfonos
    text = replace_pattern(text, r'\b(6|7)\d{8}\b', "TELEFONO", replacements)

    return text, replacements
```

Código 7-Función principal de anonimización automática en contratos

3.5.1.7 Conclusión

El módulo Copilot de Contratos de *Themis* implementa un sistema de asistencia a la redacción jurídica que permite integrar sugerencias legales adaptativas en tiempo real sin comprometer la flexibilidad del usuario. Su diseño modular incorpora funcionalidades de autocompletado contextual, verificación de datos sensibles, análisis jurídico preliminar y exportación profesional de documentos.

La estructura técnica del sistema, basada en técnicas de procesamiento de lenguaje natural, generación dinámica de *prompts*, control de privacidad mediante anonimización automática y procesos de validación estructural, permite optimizar tiempos de trabajo y reducir errores habituales en la redacción contractual.

A través de estos mecanismos, el Copilot facilita el trabajo jurídico diario en entornos profesionales, mejorando la calidad del documento final y asegurando un tratamiento seguro de la información gestionada, sin imponer modelos cerrados ni restringir la autonomía del redactor.

3.5.2 ASISTENTE LEGAL

El Asistente Legal de *Themis* está diseñado como un sistema conversacional que permite al usuario formular preguntas jurídicas en lenguaje natural y obtener respuestas fundamentadas y comprensibles. A diferencia de los buscadores jurídicos tradicionales o los formularios guiados, este módulo permite al profesional (o al usuario general) plantear una duda directamente, sin preocuparse por el formato o los términos técnicos, y recibir una respuesta estructurada, clara y vinculada a una norma real.

El sistema combina tres elementos clave:

- Procesamiento semántico de la consulta mediante *embeddings*
- Recuperación de artículos legales relevantes previamente indexados
- Generación de una respuesta explicativa utilizando un modelo generativo (Mistral 7B).

Esta lógica se ha implementado respetando el principio de privacidad: se anonimiza la consulta antes del envío a modelos externos, y se reconstruye posteriormente para mantener la coherencia del lenguaje natural.

A modo ilustrativo, la Figura 15 muestra un ejemplo real del uso del Asistente Legal. En este caso, el usuario plantea una pregunta informal sobre una situación de divorcio, y el sistema responde con una explicación clara, razonada jurídicamente, y acompañada del artículo normativo correspondiente.



Figura 15- Consulta respondida con normativa explicativa.

3.5.2.1 Flujo general del funcionamiento

El funcionamiento interno del Asistente Legal de *Themis* se basa en una arquitectura modular que combina recuperación de normativa jurídica, procesamiento semántico y generación de texto mediante modelos de lenguaje.

Cuando un usuario formula una consulta en lenguaje natural, el sistema ejecuta una serie de pasos secuenciales que garantizan que la respuesta no solo sea comprensible, sino también jurídicamente fundada. Este proceso comienza con el preprocesamiento y anonimización de la consulta, seguido de su conversión en vector semántico (*embedding*), que se compara con un índice previamente generado a partir de leyes ya vectorizadas. Finalmente, se construye un *prompt* con la normativa más relevante y se genera una respuesta adaptada al contexto a través del modelo de lenguaje.

El flujo general del sistema se representa en la Figura 16.

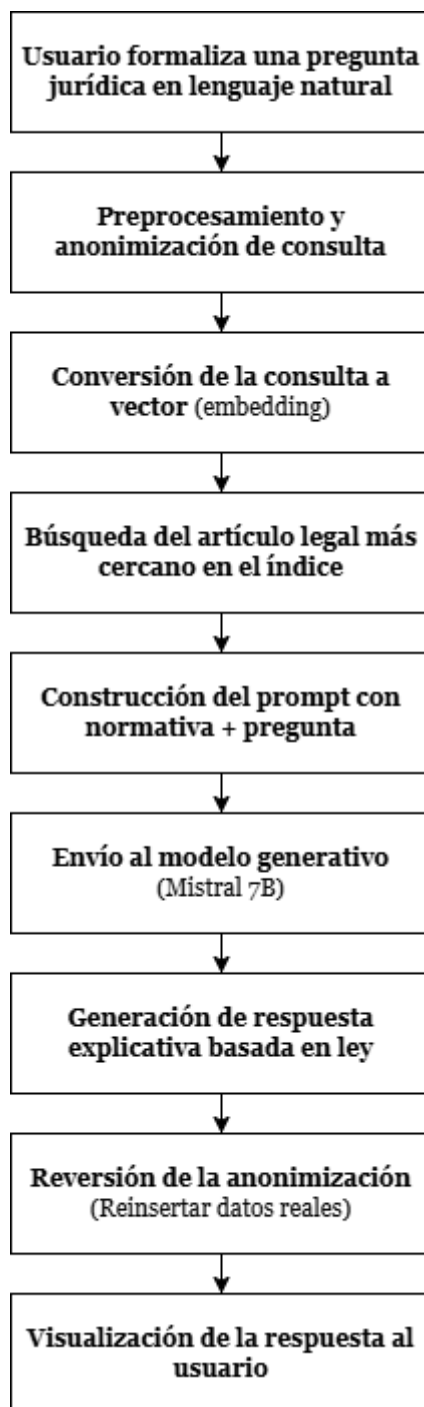


Figura 16-Flujo interno del procesamiento de consultas en el Asistente Legal

3.5.2.2 Construcción del corpus legal vectorizado

Con el objetivo de proporcionar respuestas jurídicamente fundamentadas, el Asistente Legal de *Themis* necesita acceder a normativa estructurada de forma semántica. Para ello, se ha desarrollado un proceso completo de construcción de un corpus legal vectorizado, que permite representar el contenido normativo, no solo como texto, sino también como vectores en espacio de *embeddings*. Esta representación permite calcular la similitud entre una consulta en lenguaje natural y los fragmentos normativos más relevantes, sin necesidad de coincidencias exactas en la redacción.

El sistema parte de la recuperación automatizada de la legislación publicada en el Boletín Oficial del Estado (BOE), realiza su segmentación por unidades semánticas (principalmente artículos), y genera *embeddings* de cada fragmento mediante modelos de lenguaje preentrenados. Estos vectores son posteriormente indexados mediante FAISS⁸, lo que permite realizar búsquedas rápidas por proximidad semántica. El proceso garantiza que, ante una consulta jurídica, el sistema pueda identificar el fragmento normativo más cercano por significado, incluso aunque no se mencione de forma explícita la ley o el artículo concreto.

El proceso comienza con la recopilación de fuentes normativas, a partir de una lista de URLs seleccionadas que contienen disposiciones del BOE. A continuación, se realiza un web scraping del contenido legal de cada URL, utilizando BeautifulSoup⁹ para extraer el texto contenido en el bloque HTML correspondiente al cuerpo normativo principal.

Una vez obtenido el texto, se segmenta en fragmentos semánticos, como artículos o apartados, mediante expresiones regulares adaptadas a las convenciones editoriales del BOE, lo que permite una división adecuada para la generación de *embeddings*. Estos fragmentos

⁸ FAISS (*Facebook AI Similarity Search*) es una biblioteca desarrollada por Meta (Facebook AI Research) para realizar búsquedas eficientes de similitud entre vectores en espacios de alta dimensión. En este proyecto, se utiliza para indexar los *embeddings* legales y recuperar rápidamente el fragmento normativo más cercano a una consulta del usuario.

⁹ BeautifulSoup es una librería de Python utilizada para analizar y extraer datos de archivos HTML y XML de manera eficiente. En *Themis*, se emplea para realizar web scraping del contenido legal de las disposiciones del BOE, permitiendo la extracción precisa del texto normativo contenido en los bloques HTML correspondientes.

se vectorizan semánticamente mediante el modelo `distiluse-base-multilingual-cased-v1` de SentenceTransformers¹⁰, transformando el texto en una representación numérica que mantiene su significado contextual. Finalmente, los vectores generados se indexan con FAISS (`IndexFlatL2` o `IndexIDMap`) y se almacenan junto con sus metadatos, como el título de la ley, artículo, texto original y URL, en un archivo JSON estructurado que facilita la recuperación semántica durante las consultas.

Desde el punto de vista matemático, la búsqueda se basa en la comparación entre *embeddings* de la consulta jurídica $\vec{q}$ y los vectores del corpus legal indexado $\vec{d}_i$. El índice FAISS permite recuperar los fragmentos más relevantes aplicando una métrica de similitud, como la distancia euclídea que se calcula mediante la siguiente expresión:

$$d(\vec{q}, \vec{d}_i) = \sqrt{\sum_{j=1}^n (q_j - d_{ij})^2}$$

Ecuación 3- Distancia euclídea aplicada sobre sus embeddings.

Esta fórmula permite evaluar la cercanía semántica entre dos representaciones vectoriales, sin depender de coincidencias literales.

Además, cuando se utiliza FAISS con índices basados en similitud del coseno, como *IndexFlatIP*, es necesario normalizar previamente los *embeddings*. Este proceso consiste en proyectar cada vector sobre la esfera unitaria, lo cual garantiza que la comparación entre vectores se base únicamente en su orientación y no en su magnitud. La normalización se realiza mediante la siguiente fórmula:

¹⁰ *SentenceTransformers* es una librería de Python basada en modelos de lenguaje de tipo BERT optimizados para generar representaciones vectoriales (*embeddings*) de frases o textos completos, manteniendo la semántica del contenido. En *Themis*, se utiliza para transformar fragmentos legales y consultas del usuario en vectores comparables por similitud de significado.

$$\hat{\vec{v}} = \frac{\vec{v}}{|\vec{v}|_2} = \frac{\vec{v}}{\sqrt{\sum_{i=1}^n v_i^2}}$$

Ecuación 4- Normalización L2 del vector antes de su indexación

Este paso es fundamental cuando se emplea la función `normalize_L2()` de FAISS, y mejora la precisión de recuperación semántica en consultas abiertas donde la redacción puede variar significativamente.

El proceso descrito se representa de forma secuencial en la Figura 17, donde se visualizan las distintas transformaciones desde la fuente original hasta el índice final de búsqueda.

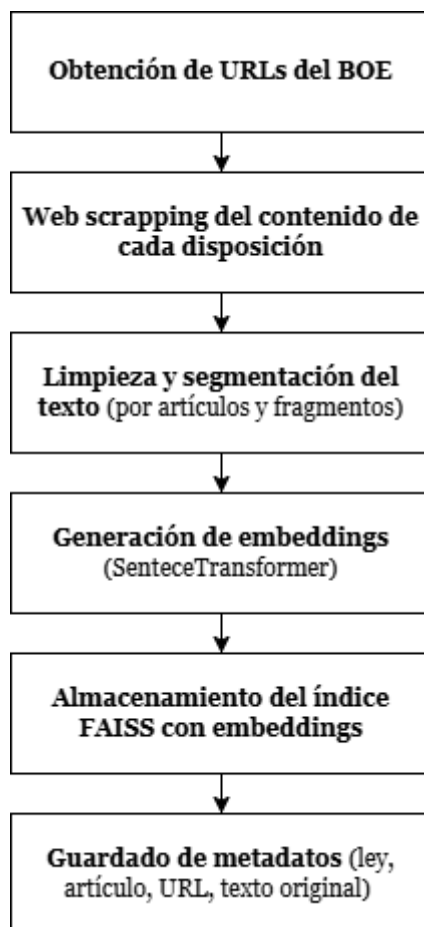


Figura 17- Obtención y vectorización de normativa jurídica.

A modo de ejemplo, a continuación se muestra un fragmento representativo del archivo JSON utilizado para almacenar los datos normativos.

```
{
```

```
"texto": "Artículo 1.\n1. No será castigada ninguna acción ni omisión que no esté prevista como delito por ley anterior a su perpetración.\n2. Las medidas de seguridad sólo podrán aplicarse cuando concurran los presupuestos establecidos previamente por la Ley.\nSe modifica el apartado 1 por el art. único.1 de la Ley Orgánica 1/2015, de 30 de marzo. Ref. BOE-A-2015-3439.\nÚltima actualización, publicada el 31/03/2015, en vigor a partir del 01/07/2015.\nTexto original, publicado el 24/11/1995, en vigor a partir del 24/05/1996.\nSubir\n[Bloque 5: #a2]",
"ley_id": "Código Penal",
"articulo": "1."
},
```

Código 8-Ejemplo de fragmento legal almacenado

3.5.2.3 Recuperación semántica y generación de respuesta

Una vez vectorizado el *corpus legal*, el sistema está preparado para identificar, ante una consulta del usuario, el fragmento normativo más cercano en significado. Esta operación se realiza mediante una búsqueda de similitud semántica utilizando el índice previamente generado con FAISS.

La consulta del usuario es preprocesada y anonimizada, y a continuación se transforma en un vector mediante el mismo modelo de *embeddings* utilizado para las leyes. Este vector de entrada se compara contra el índice FAISS y se recupera el fragmento legal con mayor proximidad semántica, medida mediante distancia coseno.

El fragmento recuperado se utiliza como base para construir un *prompt* dirigido que combina el texto normativo con la pregunta original del usuario. Este *prompt* se envía a Mistral 7B que genera una respuesta jurídica explicativa, clara y adaptada al lenguaje no técnico (o técnico si se elige el modo de respuesta profesional).

Para evitar respuestas ambiguas o desvinculadas del contenido legal, se emplea un *prompt* explícito que instruye al modelo a centrarse en explicar la duda del usuario en base a las leyes relacionadas con su pregunta obtenidas mediante los *embeddings*.

Una vez generado el *prompt*, se lanza una petición al modelo de lenguaje alojado en OpenRouter. La respuesta generada incluye una explicación formal, contextualizada con las leyes proporcionadas.

Antes de ser entregada al usuario, la respuesta es reconstruida y presentada junto a la lista de artículos que han servido como base normativa.

Se consideró la posibilidad de incluir enlaces a casos de jurisprudencia relevantes obtenidos desde una plataforma oficial del poder judicial; sin embargo, esta plataforma está diseñada para prevenir este tipo de acciones, lo que impidió su implementación a pesar de que el código ya estaba preparado. Además, se exploró la opción de utilizar una base de datos de jurisprudencia accesible para este propósito, pero todas las opciones disponibles eran de pago, lo que también dificultó su integración. Por esta razón, se optó por utilizar *embeddings* para procesar la consulta del usuario, transformándola en una representación vectorial. Al final del proceso, se añade una lista de palabras clave relevantes, lo que permite al usuario realizar búsquedas más precisas y optimizadas.

3.5.2.4 Visualización y entrega de resultados al usuario

La última fase del flujo del Asistente Legal es la entrega de la respuesta generada al usuario en un formato claro, legible y organizado. Para ello, el sistema muestra en la interfaz tanto la respuesta redactada por el modelo como el artículo legal que ha servido como base normativa, ofreciendo transparencia sobre el origen de la información.

La visualización incluye:

- Pregunta original del usuario (en lenguaje natural).
- Respuesta explicativa generada por el modelo.
- Referencia a las leyes utilizadas (nombre de la ley y artículo).

Esta información se presenta en una estructura tipo chat que permite al usuario realizar múltiples consultas encadenadas, aunque aún no se mantiene el contexto de unas a otras, es decir, no se pueden realizar hilos de conversación fluida.

Pedro y Laura se quieren divorciar y no se ponen de acuerdo sobre la división de su casa. Estaban casados en gananciales, no tienen hijos y ambos tienen buena posición económica

Figura 18-Pregunta realizada por el usuario.

En tu situación, si Pedro y Laura se quieren divorciar y no se ponen de acuerdo sobre la división de su casa, la ley puede ayudar a resolver el problema.

Teniendo en cuenta el Artículo 102 del Código Civil, la demanda de divorcio producirá los efectos siguientes: Los cónyuges podrán vivir separados y cesa la presunción de convivencia, pero no se menciona nada específico sobre la división de bienes.

Sin embargo, en el Artículo 975 del Código Civil se menciona que si uno de los cónyuges (en este caso, Pedro o Laura) enajena (vende) bienes inmuebles sujetos a reserva después de contraer segundo matrimonio, la enajenación subsistirá únicamente si a su muerte no quedan hijos. Como Pedro y Laura no tienen hijos, la enajenación de la casa podría ser una opción para ambos, pero es importante tener en cuenta que la división de bienes debe ser justa y equitativa.

Por otro lado, en el Artículo 47 del Código Civil se menciona que los parientes en línea recta por consanguinidad o adopción, los colaterales por consanguinidad hasta el tercer grado y los condenados no pueden contraer matrimonio entre sí. Como Pedro y Laura no están emparentados de ninguna manera, no hay problemas legales en cuanto a la posibilidad de divorciarse.

En resumen, si Pedro y Laura no pueden llegar a un acuerdo sobre la división de su casa, podrían recurrir a la justicia para que la divida de manera justa y equitativa. Sin embargo, es importante tener en cuenta que la ley no establece ningún procedimiento específico para la división de bienes en divorcios, por lo que es importante contar con la ayuda de un abogado para asegurarse de que la división sea justa y equitativa.

Figura 19- Parte de la respuesta con explicación del problema.

que la división sea justa y equitativa.

Leyes citadas

Código Civil. Artículo 102.

Admitida la demanda de nulidad, separación o divorcio, se producen, por ministerio de la Ley, los efectos siguientes:

1.º Los cónyuges p...

Código Civil. Artículo 975.

La enajenación que de los bienes inmuebles sujetos a reserva hubiere hecho el viudo o la viuda después de contraer segundo matrimonio su...

Código Civil. Artículo 47.

Tampoco pueden contraer matrimonio entre sí:

1. Los parientes en línea recta por consanguinidad o adopción.

2. Los colaterales por consan...

Figura 20-Mención de las leyes utilizadas.

Palabras clave

Divorcio, matrimonio ganancial, propiedad inmueble, desacuerdo, división, bienes, posición



Figura 21-Parte de la respuesta generada que se basa en la mención de palabras clave.

La interfaz de usuario del Asistente Legal está diseñada para ofrecer una experiencia de consulta completa, donde se visualiza no solo la respuesta generada, sino también los elementos que justifican su contenido. En la Figura 18 se muestra cómo el usuario plantea su consulta jurídica en lenguaje natural, sin necesidad de utilizar terminología legal específica. A continuación, la Figura 19 refleja parte de la respuesta generada por el sistema, estructurada en un lenguaje claro y

accesible, explicando la situación legal de forma contextualizada. Esta explicación se complementa con la citación directa de los artículos legales que respaldan la respuesta, como puede verse en la Figura 20, lo que garantiza la trazabilidad y precisión del contenido ofrecido. Finalmente, en la Figura 21 se pueden ver las palabras clave extraídas automáticamente del análisis semántico de la consulta, que ayudan a reforzar la relación entre el planteamiento del usuario y la base legal utilizada para responder.

Esta presentación permite al usuario entender no solo la respuesta sino también por qué se da esa respuesta, mejorando la trazabilidad, la confianza y la utilidad del sistema como herramienta de apoyo jurídico.

3.5.2.5 Jurisprudencia y limitaciones

Una de las funcionalidades inicialmente previstas para el Asistente Legal era la integración de un módulo específico para consultas basadas en jurisprudencia. La idea consistía en permitir al usuario realizar preguntas jurídicas que no solo se resolvieran mediante normativa vigente, sino también mediante resoluciones judiciales que pudieran aportar interpretación, doctrina o aplicación práctica de la ley.

Desde un punto de vista técnico, el enfoque seguido para este módulo era similar al desarrollado para las normas legales: recopilación de sentencias publicadas en fuentes oficiales (especialmente el CENDOJ¹¹), segmentación por apartados relevantes (resumen, fundamentos de Derecho, fallo), vectorización mediante *embeddings* y posterior indexación para consulta semántica. La respuesta generada estaría apoyada en los fragmentos jurisprudenciales más relevantes, ofreciendo una explicación adaptada, con enlaces y referencias.

Sin embargo, durante el desarrollo del proyecto se identificaron importantes limitaciones técnicas y legales relacionadas con la recopilación de sentencias:

¹¹ Fuente: Centro de Documentación Judicial (CENDOJ), Consejo General del Poder Judicial. Disponible en: https://www.poderjudicial.es/cgpj/es/Poder_Judicial

- La web del CENDOJ no permite scraping automatizado, estando protegida contra bots, y no ofrece una API pública directa para acceder a su base de datos jurisprudencial.
- Existen restricciones legales sobre la reutilización masiva de resoluciones judiciales, especialmente en lo referente a su anonimización, uso comercial o republicación sin licencia.

Por este motivo, si bien la arquitectura del sistema ha sido diseñada para permitir la integración futura de jurisprudencia directa en el asistente, actualmente no se realiza ningún procesamiento local ni recuperación automatizada de resoluciones judiciales.

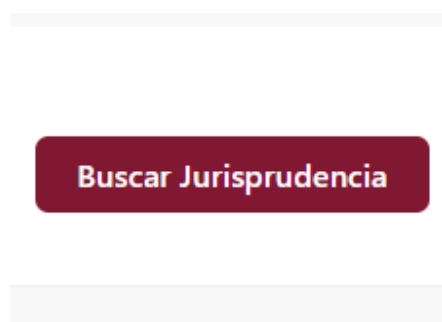


Figura 22-Botón “Buscar Jurisprudencia”.

No obstante, el asistente cuenta con un botón visible de “Buscar Jurisprudencia” en la interfaz (ver Figura 22), que redirige directamente al buscador oficial del Consejo General del Poder Judicial (<https://www.poderjudicial.es/search/indexAN.jsp>). Esto permite al usuario realizar consultas manuales en la base de datos jurisprudencial en caso de que desee ampliar la información más allá de lo ofrecido por la normativa vigente.

Esta solución, aunque externa, garantiza acceso a fuentes oficiales mientras se mantiene la compatibilidad con una futura integración automatizada si se habilitan vías legales o técnicas adecuadas.

3.5.2.6 Generación y descarga del informe jurídico

Una vez completada y generada la consulta, el Asistente Legal permite generar un informe en formato PDF con el contenido de la respuesta. Esta funcionalidad ofrece al usuario una forma práctica de guardar o compartir la información jurídica generada durante la interacción.

La estructura de este PDF es similar a la implementada en el módulo Copilot, por lo que no se detallan de nuevo las funciones técnicas de exportación. No obstante, en este caso concreto, el nombre del archivo se genera automáticamente a partir de palabras clave extraídas de la consulta, lo que permite que el nombre del documento tenga sentido semántico sin intervención del usuario (por ejemplo: `pedro_laura_quieren_divorciar.pdf`).

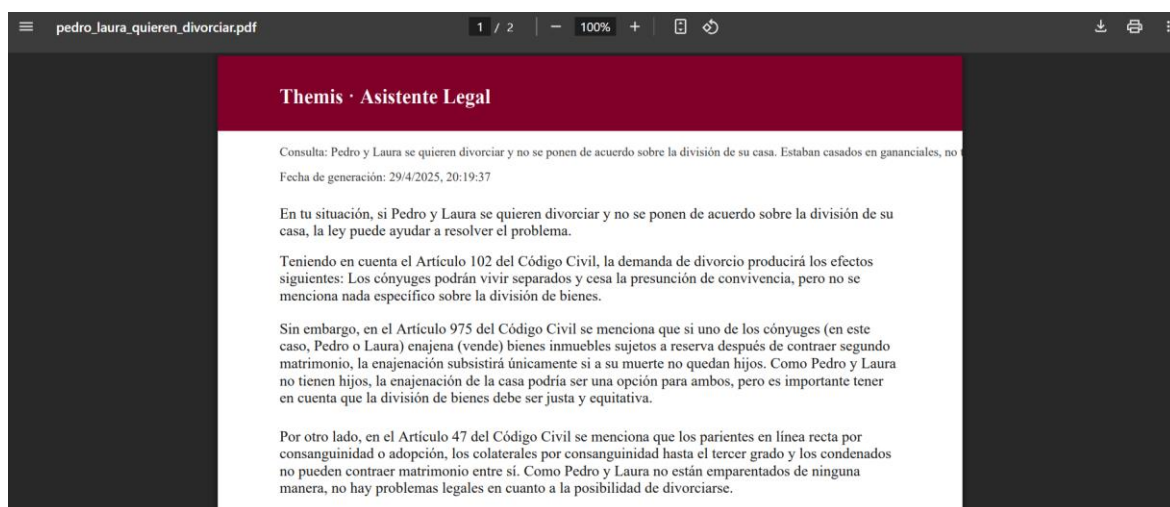


Figura 23-Fragmento del informe generado.

En la Figura 23 se muestra un ejemplo real del informe generado por el Asistente Legal, donde puede observarse cómo se integra la consulta del usuario, la fecha y la hora de generación y la respuesta jurídica redactada de forma estructurada. Este formato permite al usuario guardar el contenido como registro de consulta o utilizarlo como base documental. Por su parte, en la Figura 24 se presenta el pie del documento, que incluye de forma automática un aviso legal indicando que el informe ha sido generado por un sistema de

inteligencia artificial y que no sustituye la asesoría jurídica profesional. Esta indicación es importante desde el punto de vista de cumplimiento y responsabilidad, y se añade de forma estandarizada en todos los informes generados.

Este informe ha sido generado automáticamente por el asistente de IA de Themis. No sustituye asesoría legal profesional.

Figura 24-Aviso legal incluido en el pie del documento PDF generado.

3.5.2.7 Conclusión

El módulo Asistente Legal de *Themis* demuestra cómo es posible aplicar técnicas de procesamiento de lenguaje natural, recuperación semántica y generación de texto para ofrecer respuestas jurídicas claras y fundamentadas. A diferencia de otros sistemas genéricos, este asistente está diseñado específicamente para operar sobre legislación española real, vectorizada previamente, y devolver explicaciones adaptadas al lenguaje natural del usuario.

La arquitectura modular implementada (desde la construcción del *corpus legal* hasta la presentación visual y generación del informe) permite escalar y adaptar el sistema en futuras versiones, incluyendo la posible incorporación de jurisprudencia y otras fuentes doctrinales. En su estado actual, el Asistente Legal ya proporciona una herramienta funcional y práctica de apoyo para la consulta jurídica informal, conservando siempre principios de trazabilidad, transparencia y respeto a la privacidad de los datos procesados.

Capítulo 4. TECNOLOGÍAS UTILIZADAS

Este proyecto ha sido desarrollado con una arquitectura modular basada en cliente-servidor, donde cada parte del sistema utiliza tecnologías específicas seleccionadas por su robustez, rendimiento, compatibilidad con el lenguaje natural y facilidad de integración. En este capítulo se detalla el conjunto de lenguajes, *frameworks*, librerías y servicios más importantes empleados, así como las razones técnicas que justifican su uso en el contexto de *Themis*.

4.1 LENGUAJES DE PROGRAMACIÓN Y ENTORNO BASE

Lenguajes de programación principales y entorno base empleados para el desarrollo del *backend*, *frontend* y entorno de ejecución de *Themis*. Se detallan en función y la motivación técnica para su elección

Tecnología	Uso en el proyecto	Justificación
Python	Desarrollo completo del <i>backend</i> , <i>scraping</i> , <i>embeddings</i> , validación legal y procesamiento NLP.	Lenguaje versátil, con gran ecosistema de usuarios y soporte para librerías de IA y NLP.
JavaScript	Desarrollo del <i>frontend</i> en React.	Permite una experiencia dinámica e interactiva en el navegador.
Node.js	Entorno de ejecución necesario para compilar y ejecutar el <i>frontend</i> React localmente.	Imprescindible para gestión de dependencias (<code>npm</code>) y compilación en desarrollo web.

HTML + CSS	Generación de vistas renderizadas y maquetación de documentos para exportación.	Estándares necesarios para estructurar y presentar el contenido legal renderizado.
------------	---	--

Tabla 1-Lenguajes y entornos utilizados.

4.2 BACKEND Y ESTRUCTURA DEL SERVIDOR

Frameworks y librerías del servidor utilizados para construir la lógica de negocio, la API REST y la interacción con los modelos de lenguaje. Se justifica su elección en función de su ligereza, versatilidad y compatibilidad con Python.

Tecnología	Rol específico	Justificación
Flask	Framework de <i>backend</i> (API REST).	Ligero, fácil de desplegar y suficiente para las necesidades del sistema.
OpenRouter API	Acceso a modelos de lenguaje como Mistral 7B.	Ofrece una interfaz sencilla y personalizable para trabajar con LLMs sin necesidad de desplegarlos localmente.
spaCy	Anonimización de datos y extracción de entidades.	Librería eficiente y entrenada en español para NLP legal.
re / json / os / logging (built-in Python)	Validación, manipulación de texto, control de flujo y trazabilidad.	Evita dependencias externas innecesarias y mejora rendimiento.

Tabla 2-Tecnología empleadas en el *backend*.

4.3 FRONTEND

Herramientas y librerías utilizadas en el desarrollo de la interfaz gráfica del usuario, orientadas a una experiencia reactiva, accesible y con capacidad de exportación documental.

Tecnología	Rol específico	Justificación
React	Interfaz del usuario de todos los módulos (Copilot y Asistente Legal).	Framework moderno y escalable con gran compatibilidad para SPAs. ¹²
jsPDF	Exportación de contratos y respuestas legales a PDF.	Permite controlar con precisión el formato y estructura del documento generado directamente desde el navegador.
DOMParser / Blob / FileSaver.js	Conversión de HTML a PDF y gestión de descargas.	Soporte directo en navegadores, necesario para generar y descargar archivos localmente.

Tabla 3-Tecnologías empleadas en el *frontend*.

4.4 PROCESAMIENTO DEL LENGUAJE NATURAL

Librerías especializadas en generación de texto, extracción de entidades y vectorización semántica. Estas tecnologías permiten estructurar, interpretar y generar contenido jurídico de forma automatizada.

¹² SPAs (Single Page Applications) son aplicaciones web o sitios web diseñados para funcionar como una única página HTML, donde el contenido se actualiza dinámicamente sin necesidad de recargar la página completa. Esto mejora la experiencia del usuario al ofrecer una navegación más fluida y rápida. En el contexto de *Themis*, se emplean para crear interfaces de usuario interactivas que permiten consultas legales sin interrupciones, proporcionando una respuesta inmediata a las acciones del usuario.

Tecnología	Rol específico	Justificación
SentenceTransformers	Generación de <i>embeddings</i> de leyes y consultas.	Modelo multilingüe (distiluse-base-multilingual-cased-v1) que ofrece alta precisión semántica con bajo coste computacional.
FAISS	Búsqueda de leyes relevantes mediante similitud vectorial.	Biblioteca optimizada para recuperar vectores similares en espacios de alta dimensión. Ideal para NLP jurídico.
BeautifulSoup4	<i>Web scraping</i> del BOE para construir el corpus legal.	Permite extraer el texto de forma controlada y robusta desde páginas HTML sin cargar JS.
Transformers (Hugging Face)	Base de los modelos utilizados en procesamiento previo.	Biblioteca estándar para uso de modelos tipo BERT ¹³ , aunque solo empleada indirectamente a través de SentenceTransformers.

Tabla 4-Tecnologías de procesamiento de lenguaje natural.

¹³ Modelos tipo BERT (Bidirectional Encoder Representations from Transformers) son modelos de lenguaje que entienden el contexto de las palabras observando tanto las anteriores como las posteriores. Esta capacidad bidireccional los hace efectivos para tareas de procesamiento de lenguaje natural como la generación de *embeddings* y la respuesta a preguntas. En *Themis*, se utilizan para transformar fragmentos legales y consultas del usuario en representaciones vectoriales, facilitando comparaciones semánticas.

4.5 VALIDACIÓN, SEGURIDAD Y CONTROL DE DATOS

Tecnología	Rol específico	Justificación
spaCy NER	Detección y anonimización de entidades sensibles como nombres, organizaciones o localizaciones.	Permite anonimizar automáticamente antes de enviar texto a modelos externos, respetando la privacidad.
Expresiones regulares (re)	Validación estructural de datos como DNIs, teléfonos, referencias catastrales, etc.	Técnica ligera y de alto rendimiento para comprobar patrones específicos sin necesidad de modelos entrenados.

Tabla 5-Herramientas empleadas para la detección y control de datos sensibles

4.6 INFRAESTRUCTURA Y HERRAMIENTAS DE DESARROLLO

Utilidades de desarrollo y entorno utilizadas para la gestión de proyecto, instalación de dependencias, control de versiones y ejecución del *frontend*.

Tecnología	Rol específico	Justificación
Git	Control de versiones y colaboración.	Estándar en desarrollo software, permite trazabilidad completa del código.
npm / Node	Instalación y ejecución del <i>frontend</i> .	Necesario para gestionar las dependencias en React.
Visual Studio Code	Entorno de desarrollo (IDE).	Entorno ligero y extensible con soporte completo para Python y JavaScript.

Tabla 6-Herramientas de soporte y entorno de desarrollo.

4.7 INSTALACIÓN Y DEPENDENCIAS CLAVE

Para ejecutar correctamente el sistema completo se han instalado y configurado los siguientes elementos en el entorno local:

- Python 3.10.6
- Node.js (v18.x LTS)
- `pip + requirements.txt` con:
 - `flask, openai, spacy, sentence-transformers, faiss-cpu, beautifulsoup4, requests, python-dotenv`
- `npm install` con:
 - `react, jsPDF, react-router-dom`

4.8 BASES DE DATOS Y PERSISTENCIA DE INFORMACIÓN

Themis emplea un enfoque mixto en relación con a la gestión y almacenamiento de datos. Aunque ciertos módulos como el asistente legal se apoyan en estructuras no relacionales para trabajar con fragmentos vectorizados y consultas semánticas requieren una estructura relacional que permita integridad referencial, trazabilidad de acciones y relaciones entre tablas.

Para ello, se ha utilizado:

Tecnología	Rol específico	Justificación
SQLAlchemy	Mapeo objeto-relacional (ORM) entre modelos de datos Python y base de datos SQL.	Permite trabajar con bases de datos relacionales de forma abstracta, manteniendo portabilidad y claridad en el código. Facilita la creación de modelos relacionales sin escribir SQL puro.
SQLite	Motor de base de datos local durante el desarrollo y pruebas.	Ligera, sin necesidad de servidor, ideal para entornos de desarrollo. Permite almacenar datos estructurados de forma persistente.

Tabla 7-Tecnologías empleadas para la gestión de bases de datos.

4.9 SELECCIÓN DEL MODELO DE LENGUAJE: MISTRAL 7B VÍA OPENROUTER

Themis se apoya en modelos de lenguaje de gran tamaño (LLMs) para la generación de sugerencias jurídicas, respuestas conversacionales y redacción estructurada de contratos. Tras valorar varias opciones, se optó por utilizar *Mistral 7B* a través de la API pública de *OpenRouter*.

Esta decisión responde a un equilibrio entre rendimiento, flexibilidad y accesibilidad económica durante el proceso de desarrollo e iteración del sistema. *Mistral 7B* es un modelo *open source* basado en arquitectura *transformer*, entrenado sobre corpus multilingües y optimizado para tareas de generación instructiva. En las pruebas realizadas, ofreció resultados satisfactorios en términos de coherencia gramatical, sentido jurídico y comprensión contextual para los tipos de tareas exigidos por *Themis*.

Aunque modelos comerciales como GPT-4 (*OpenAI*) presentan un rendimiento superior en tareas complejas de interpretación jurídica o razonamiento multietapa, su uso a través de la API oficial implica un coste por *token* elevado, que no resultaba viable para un proyecto en fase de desarrollo intensivo. Durante la implementación, era necesario realizar pruebas frecuentes con grandes volúmenes de textos y generar múltiples iteraciones de ajuste, lo cual hubiera necesitado una gran necesidad de recursos.

Además, *OpenRouter* ofrece una API con una arquitectura muy similar a la de *OpenAI*, lo que facilita su integración y permite una futura migración transparente a otros modelos si fuese necesario.

Capítulo 5. METODOLOGÍA Y PLANIFICACIÓN

El desarrollo de *Themis* se ha llevado a cabo siguiendo una metodología *Agile*, iterativa y basada en validación continua, adaptada a un contexto académico unipersonal. Aunque los principios fundamentales de *Agile* han guiado el proceso, las decisiones técnicas, funcionales y de priorización han sido centralizadas. Durante el proyecto, se ha priorizado la construcción de valor funcional desde etapas tempranas, sin esperar al desarrollo completo de cada módulo. Este enfoque ha combinado ciclos breves de implementación con validación práctica por parte de profesionales del Derecho. La colaboración directa con abogados ha sido crucial, permitiendo identificar necesidades reales, ajustar el lenguaje y la estructura del sistema, y asegurar que las funcionalidades fueran útiles en un contexto legal auténtico.

5.1 PRINCIPIOS APLICADOS DEL ENFOQUE AGILE

Aunque no se aplicó un marco cerrado como Scrum¹⁴ o XP¹⁵, sí se adoptaron varios principios clave del desarrollo ágil:

¹⁴ Scrum es un marco de trabajo ágil que organiza el desarrollo de proyectos en ciclos cortos llamados "*sprints*". Fomenta la colaboración en equipo, la adaptabilidad y la entrega continua de valor, con roles clave como el Product Owner y el Scrum Master.

¹⁵ XP (*Extreme Programming*) es una metodología ágil que se enfoca en mejorar la calidad del software y la capacidad de respuesta al cambio. Utiliza prácticas como el desarrollo iterativo, la programación en pareja, las pruebas continuas y la integración frecuente para asegurar un software de alta calidad y adaptarse rápidamente a los cambios en los requisitos.

Principio Agile aplicado	Implementación práctica en <i>Themis</i>
Iteraciones cortas y funcionales	El desarrollo se dividió en ciclos de uno o dos meses, donde se completaban módulos concretos. Dentro de los <i>sprints</i> había <i>subtaks</i> que se iban completando poco a poco y de forma iterativa.
Desarrollo incremental	Cada módulo fue creciendo en profundidad y complejidad a lo largo de varias iteraciones, en lugar de desarrollarse de forma monolítica.
Feedback continuo	Se mantuvo contacto frecuente con abogados para validar textos generados, estructura de contratos, lógica legal del asistente, etc.
Prioridad al valor funcional	Se priorizaron funcionalidades útiles y visibles (Copilot, análisis legal, exportación), frente a integraciones futuras menos críticas (como el gestor de tareas completo o estética de la plataforma).
Refactorización continua	El código fue revisado y mejorado tras cada iteración, especialmente en el manejo de errores, anonimización y diseño del <i>frontend</i> .

Tabla 8-Principios ágiles adaptados al desarrollo del sistema *Themis*.

5.2 COMUNICACIÓN DIRECTA CON EL USUARIO FINAL

Una de las claves del enfoque iterativo ha sido la colaboración directa y constante con profesionales del sector jurídico, que han actuado como validadores del producto desde su concepción. El conocimiento de primera mano de las dinámicas reales en despachos de

abogados permitió ajustar tanto el lenguaje usado en el sistema como las funcionalidades a priorizar.

Ejemplos de cómo este *feedback* ha impactado directamente en el desarrollo:

- El sistema de redacción libre con sugerencias surgió de la necesidad expresada de evitar formularios rígidos.
- La anonimización automática fue una petición directa para permitir el uso del sistema sin vulnerar datos sensibles durante pruebas.
- La verificación jurídica preliminar fue sugerida para detectar omisiones habituales en contratos reales y minimizar riesgos.

Esta comunicación no se estructuró en reuniones formales regulares, pero sí se mantuvo viva a través de iteraciones semanales, prototipos funcionales enviados, y breves validaciones de resultados por parte de usuarios expertos.

5.3 DIAGRAMA ITERATIVO CICLOS DE TRABAJO (SPRINTS)

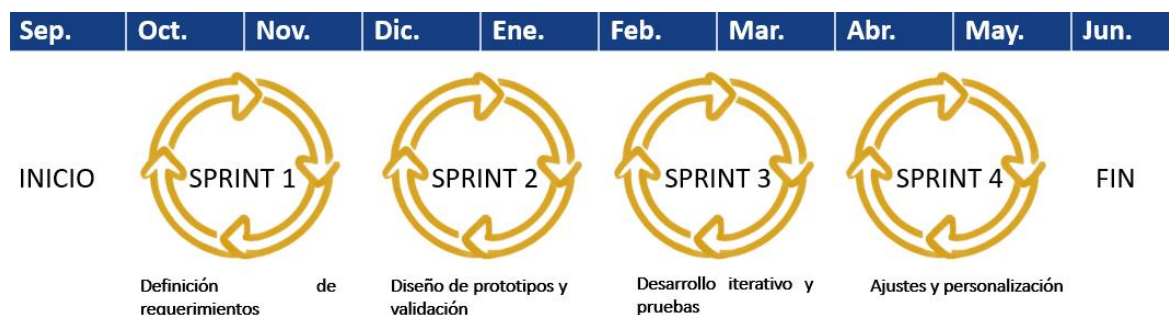


Figura 25-Ciclos iterativos de desarrollo del sistema *Themis*.

El proyecto *Themis* se ha organizado en cuatro *sprints* iterativos en los que se priorizó entregar valor funcional de forma progresiva. Aunque no se aplicó *Scrum* de forma formal (sin equipo completo ni reuniones diarias), se respetaron sus fundamentos: desarrollo incremental, validación frecuente con *stakeholders* (en este caso, abogados) y entrega de versiones mejoradas tras cada ciclo. En la Figura 25 se puede observar a alto nivel el

desarrollo de los *sprints* a lo largo del curso académico en el cual se ha llevado a cabo este proyecto. A continuación, se describe cada sprint con sus tareas clave y temporalización.

5.3.1 SPRINT 1 – DEFINICIÓN DE REQUISITOS Y BACKLOG

Septiembre – Octubre

En esta primera fase se realizó la definición estratégica del sistema. Se identificaron las necesidades clave del usuario objetivo (profesionales del Derecho), así como los principales módulos a desarrollar. Se estableció el alcance técnico del proyecto, y se analizaron las tecnologías más apropiadas según criterios de rendimiento y compatibilidad con NLP. Se generó el primer backlog con las historias de usuario asociadas a los módulos principales. Se creó el primer listado de tareas (backlog¹⁶) con las historias de usuario correspondientes a los módulos principales.

Tareas principales realizadas:

- Análisis del contexto LegalTech y referencias de mercado.
- Identificación de necesidades a través de feedback jurídico real.
- Selección y justificación de tecnologías base (*React, Flask, spaCy, FAISS...*).
- Estructura general del repositorio, planificación de tareas y backlog funcional.

¹⁶ *Backlog* es una lista priorizada de tareas y requisitos pendientes de desarrollar en un proyecto. En metodologías ágiles, el *Product Backlog* incluye todas las funcionalidades del producto, mientras que el *Sprint Backlog* contiene las tareas seleccionadas para el sprint actual.

5.3.2 SPRINT 2 – DISEÑO DE PROTOTIPOS INICIALES Y VALIDACIÓN

Noviembre – Enero

Esta fase se centró en el diseño y construcción de los primeros prototipos funcionales. Se priorizó la interfaz del Copilot de Contratos por ser el módulo más visual y comprensible, además del más complejo de programar. Se establecieron mecanismos de comunicación entre *frontend* y *backend* y se implementaron los primeros flujos de sugerencias. Se realizaron validaciones tempranas con profesionales del Derecho.

Tareas principales realizadas:

- Desarrollo del editor de texto para contratos con integración *React-Flask*.
- Implementación del sistema de autocompletado adaptativo con espera de 1 segundo.
- Integración del pipeline de anonimización (*spaCy* + expresión regular).
- Renderizado HTML y exportación básica a PDF.
- Feedback real sobre redacción legal y estructura de cláusulas.
- Primer test con contratos reales (caso “contrato de prestación de servicios”).

5.3.3 SPRINT 3 – DESARROLLO ITERATIVO Y PRUEBAS SEMÁNTICAS

Febrero – Marzo

En este *sprint* se abordó la construcción del Asistente Legal con recuperación semántica de normativa. Fue el punto de mayor carga técnica. Se generó el corpus legal vectorizado (*scraping* BOE + *embeddings*), se integró *FAISS* como índice de búsqueda, y se construyeron los prompts para explicar el contenido normativo. Se testearon consultas jurídicas abiertas y se validó la solidez legal de las respuestas.

Tareas principales realizadas:

- Implementación del *scraper* del BOE y fragmentación de leyes por artículos.
- Generación y almacenamiento de *embeddings* legales mediante *SentenceTransformers*.
- Indexación semántica con *FAISS* y recuperación eficiente de artículos.
- Generación de respuestas mediante *Mistral 7B* y pruebas de coherencia legal.
- Análisis jurídico preliminar de contratos mediante LLM.
- Validación directa de consultas con abogados (“Pedro y Laura se quieren divorciar” como caso testigo).

5.3.4 SPRINT 4 – AJUSTES, PERSONALIZACIÓN Y DOCUMENTACIÓN

Abril – Mayo

En este último ciclo se enfocó el esfuerzo en la estabilidad y presentación del sistema. Se depuraron errores, se añadieron mejoras visuales y de experiencia de usuario, y se ajustaron pequeños detalles técnicos (nombres de archivos generados, consistencia visual entre módulos, *disclaimer* legal). Se finalizó la integración de todos los módulos y se redactó la documentación técnica y memoria completa del proyecto.

Tareas principales realizadas:

- Depuración de errores de formato, sugerencias duplicadas y reconstrucción de datos.
- Incorporación del botón “Buscar Jurisprudencia” con redirección externa.
- Ajustes en la generación automática del nombre de archivos PDF con base en palabras clave.
- Mejora del flujo visual de edición–verificación–exportación.
- Redacción de la memoria, documentación técnica y diagramas funcionales.
- Preparación de entregables, capturas, vídeos y pruebas finales.

A lo largo del proyecto, se ha seguido una metodología iterativa, flexible y centrada en la entrega de valor funcional desde las primeras fases. La adaptación de principios ágiles ha permitido desarrollar un sistema progresivamente más robusto, ajustando prioridades según el impacto real de cada funcionalidad central (como el Copilot y el Asistente Legal), y validando decisiones técnicas en colaboración continua con profesionales del Derecho.

Esta combinación de planificación estructurada y evolución basada en el feedback ha sido clave para que *Themis* no solo funcione técnicamente, sino que también responda a necesidades concretas del sector jurídico. Los *sprints* han permitido aislar funcionalidades, reducir el riesgo técnico y asegurar la integración fluida de los distintos componentes del sistema.

Esta metodología, más allá de facilitar el desarrollo del proyecto, ha sentado las bases para su escalabilidad y mantenimiento futuro.

5.4 ESTIMACIÓN DE COTES: DESARROLLO DE UN PROTOTIPO FUNCIONAL OPEN SOURCE

El presente proyecto ha consistido en el desarrollo de un prototipo funcional avanzado (lo que en la industria se conoce como un *mockup funcional*¹⁷), que reproduce con alta fidelidad el comportamiento esperado de una solución Legaltech real. Aunque no se trata aún de un prototipo desplegado de forma comercial, su grado de completitud permite interactuar con el sistema, probar sus funcionalidades clave y evaluar su valor técnico de forma tangible.

Desde el inicio se ha apostado por un enfoque reproducible, usando de esta forma tecnologías de código abierto (*open source*). Además, ha permitido reducir al mínimo los costes económicos sin sacrificar funcionalidad ni realismo.

¹⁷ Se entiende por mockup funcional una simulación visual que reproduce la interfaz y las funcionalidades clave del sistema, con fines demostrativos y sin requerir desarrollo completo.

Este apartado presenta una estimación del coste equivalente que habría tenido este desarrollo si se hubiera llevado a cabo en un entorno profesional, así como una proyección de costes adicionales necesarios para convertir esta versión académica en una solución lista para lanzar al mercado.

5.4.1 ALCANCE DEL PROTOTIPO

El sistema desarrollado incluye:

- Un modelo operativo de redacción asistida de contratos, con generación de cláusulas, autocompletado inteligente y verificación semántica.
- Un sistema de anonimización y reconstrucción de datos conforma al RGPD¹⁸.
- Interfaz funcional en React, exportación a PDF y *backend* en Flask.
- Arquitectura modular escalable, preparada para su expansión posterior.

Todo ello permite validar las funcionalidades esenciales, la viabilidad técnica del enfoque y la experiencia del usuario, aunque aún no se ha implementado una infraestructura de producción ni mecanismos de escalabilidad, ni cumplimiento normativo avanzado (firma digital, logs trazables, etc.)

5.4.2 COSTE ESTIMADO DEL DESARROLLO DEL PROTOTIPO

Aunque el desarrollo ha sido llevado a cabo por única persona en el marco del Trabajo de Fin de Grado, es posible estimar su valor económico aproximado tomando como referencia el tiempo dedicado y una tarifa media común en el sector tecnológico:

¹⁸ Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo, de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos (RGPD).

Actividad	Horas estimadas
Análisis del sector LegalTech y del usuario	20h
Diseño técnico y arquitectura del sistema	35h
<i>Backend</i> (Flask, NLP, lógica IA)	70h
<i>Frontend</i> (React, PDF, interfaz visual)	50h
Módulo Copilot y generación contextual	40h
Verificación semántica y anonimización	20 h
Reuniones, feedback y revisión iterativa	15h
Pruebas funcionales y mejora de UX	15 h
Redacción y documentación técnica	45 h
Total estimado	310h

Tabla 9- Distribución realista de horas dedicadas durante el desarrollo del prototipo.

Estimando un valor de mercado de 25€/h para un perfil junior técnico, el desarrollo equivaldría a un coste de

$$310h \times 25€/h = 7750€$$

Ecuación 5-Estimación de costes

5.4.3 COSTES ADICIONALES PARA SU CONVERSIÓN EN PRODUCTO FINAL

Como ya hemos mencionado, este prototipo demuestra la viabilidad técnica y el valor funcional de la solución, pero no está aún preparado para un uso comercial real. Su despliegue como producto real implicaría costes adicionales entre los que se incluirían:

- Infraestructura escalable en la nube (servidores, bases de datos, APIs)
- Seguridad, autenticación robusta y cumplimiento normativo.
- Mejora de un diseño UX/UI que sea profesional y accesible.
- Testeo intensivo en entornos reales y control de calidad legal.
- Soporte técnico, actualizaciones y mantenimiento.
- Marketing, documentación y formación de usuarios finales.

El presupuesto necesario para este tipo de desarrollo completo podría situarse en torno a los 50.000€ y 150.000€, dependiendo del nivel de automatización, tipo de cliente objetivo (despacho, SaaS, administración pública...), y la cobertura funcional deseada.

Capítulo 6. RESULTADOS FUNCIONALES Y

VISUALIZACIÓN DEL SISTEMA

6.1 INTRODUCCIÓN

El presente capítulo recoge los resultados visuales más relevantes obtenidos tras el desarrollo de *Themis*. A través de estas capturas reales y ejemplos prácticos se busca evidenciar que el sistema no es únicamente funcional, sino que ofrece una experiencia de usuario intuitiva y una solución tecnológica operativa que responde a las necesidades del entorno jurídico.

A lo largo del mismo, se mostrarán casos de uso concretos que ilustran la redacción asistida de contratos con sugerencias contextuales, la exportación profesional de documentos en formato PDF, y la interacción con el asistente legal conversacional mediante consultas en lenguaje natural. Esta demostración práctica permite validar de forma tangible la viabilidad y eficacia del ecosistema LegalTech diseñado e implementado.

Como parte del proceso de evaluación y validación, el código fuente completo, junto con vídeos de demostración y documentación, está disponible en un repositorio público en GitHub:

<https://github.com/tfgmcasmon/copilotcontratos.git>

Este repositorio facilita la replicación y ampliación del sistema por parte de la comunidad LegalTech.

6.2 CASO DE USO COMPLETO: REDACCIÓN ASISTIDA DE CONTRATOS

El siguiente contrato ha sido redactado en el editor del Copilot de Contratos de *Themis*, incluyendo errores intencionados para la demostración del sistema. A continuación, se presentan capturas reales del editor y sus funcionalidades automáticas de detección, sugerencias y análisis jurídico, junto con la explicación detallada de cada error y su motivo.

CONTRATO DE ARRENDAMIENTO DE VIVIENDA

Entre Don/Doña Ana Sánchez Ruiz, mayor de edad, con domicilio en Calle Mayor, nº 15, y NIF 23456789A, en adelante “ARRENDADORA”,

y Don/Doña Pedro Gómez López, mayor de edad, con domicilio en Calle Luna, nº 8, y NIF 98765432B, en adelante “ARRENDATARIO”,

se acuerda lo siguiente:

1. Objeto del contrato

La ARRENDADORA da en arrendamiento al ARRENDATARIO la vivienda situada en Calle Sol, nº 20, con una superficie de 80 metros cuadrados.

2. Duración

El contrato tendrá una duración de 1 año, iniciándose el 01/05/2025 y finalizando el 30/04/2026.

3. Renta

El ARRENDATARIO pagará una renta mensual de 600 euros, que deberá abonar los primeros cinco días de cada mes.

4. Fianza

El ARRENDATARIO entrega en este acto la cantidad de 1.200 euros en concepto de fianza.

5. Obligaciones del arrendatario

- *Mantener en buen estado la vivienda.*
- *No realizar obras sin autorización escrita de la ARRENDADORA.*

6. Resolución del contrato

En caso de incumplimiento grave por parte del ARRENDATARIO, la ARRENDADORA podrá resolver el contrato sin necesidad de preaviso.

7. Jurisdicción

Para cualquier controversia derivada del presente contrato, ambas partes se someten a la jurisdicción de los tribunales de Madrid.

Firmas:

Figura 26-Contrato con errores intencionados para demostración del Copilot.

6.2.1 REDACCIÓN CONTEXTUAL ASISTIDA

El editor del Copilot (ver Figura 27) permite una redacción fluida y contextualizada gracias a su sistema de autocompletado en tiempo real. Conforme el usuario escribe, el sistema analiza el contenido y ofrece sugerencias adaptadas al tipo de cláusula y contexto legal. Estas sugerencias aparecen en un panel lateral, como se observa en Figura 28, y pueden ser incorporadas al texto pulsando la tecla Tab, lo que agiliza significativamente el proceso de redacción y reduce errores.

Este enfoque facilita que el profesional jurídico mantenga el control total del documento, mientras se beneficia de recomendaciones automatizadas que mejoran la precisión, completitud y coherencia del contrato. Además, la integración de esta funcionalidad en un

único entorno permite optimizar los tiempos y mejorar la calidad documental sin depender de plantillas rígidas ni formularios estáticos.

Entre Don/Doña Ana Sánchez Ruiz, mayor de edad, con domicilio en Calle Mayor, n° 15, y NIF 23456789A, en adelante "ARRENDADORA",
y Don/Doña Pedro Gómez López, mayor de edad, con domicilio en Calle Luna, n° 8, y NIF 98765432B, en adelante "ARRENDATARIO",

se acuerda lo siguiente:

1. Objeto del contrato
La ARRENDADORA da en arrendamiento al ARRENDATARIO la vivienda situada en Calle Sol, n° 20, con una superficie de 80 metros cuadrados.

Figura 27-Editor de contratos con autocompletado en tiempo real.

 Sugerencia:

2. Duración del contrato Este contrato de arrendamiento de vivienda tendrá una duración de [DURACION] a partir de la fecha de esta escritura, [FECHA].

Figura 28-Panel de sugerencias contractuales en tiempo real.

6.2.2 VERIFICACIÓN AUTOMÁTICA DE DATOS

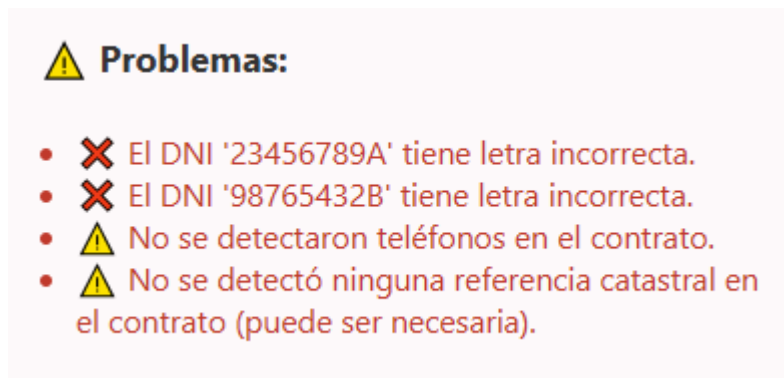


Figura 29- Panel de errores detectados durante la redacción.

Como se muestra en la Figura 29, el sistema de verificación automática de datos resalta en tiempo real las incoherencias y errores detectados en el contrato, facilitando la revisión inmediata por parte del jurista. Entre los principales problemas identificados se encuentran:

- **DNI con letra incorrecta:** Ejemplos detectados como 23456789A y 98765432B, donde la letra no coincide con el número proporcionado, violando el formato oficial.
- **Falta de teléfonos:** Ausencia de números telefónicos en el contrato, elemento importante para notificaciones y contacto.
- **Referencia catastral no detectada:** Omisión de referencias catastrales que pueden ser necesarias para identificar inmuebles correctamente.
- **Nombres duplicados o inconsistentes:** Detección de nombres que aparecen repetidos o con variaciones que podrían generar confusión o errores legales.
- **Campos sin rellenar o marcadores:** Presencia de etiquetas como [DURACIÓN], [FECHA] o [NÚMERO] que indican información pendiente de completar.
- **Errores de formato general:** Inconsistencias en el uso de mayúsculas, signos de puntuación o espacios que afectan la formalidad y presentación del documento.

Este mecanismo de detección automática permite minimizar riesgos legales derivados de errores formales y mejora significativamente la calidad y coherencia del contrato antes de su validación definitiva.

6.2.3 INFORME JURÍDICO AUTOMATIZADO

El sistema genera automáticamente un informe jurídico (Figura 30) preliminar que analiza el contenido del contrato redactado, identificando aspectos clave, omisiones y posibles riesgos legales. Este informe ofrece al profesional del Derecho una visión rápida y estructurada del estado del documento, facilitando su revisión y mejora.

Entre los puntos destacados que el informe puede contener se incluyen:

- **Sentido Jurídico:** Evaluación general sobre la coherencia y validez del contrato, confirmando que refleja un acuerdo legal válido entre las partes.
- **Cláusulas esenciales faltantes:** Identificación de cláusulas críticas no incluidas, como:
 - Cláusula de Terminación: falta una disposición que establezca condiciones para la finalización anticipada del contrato.
 - Cláusula de Responsabilidad: ausencia de regulación sobre la responsabilidad de las partes frente a incumplimientos.
 - Cláusula de Confidencialidad: falta de protección explícita para la información sensible.
- **Inconsistencias normativas:** Señalamiento de posibles contradicciones o incoherencias entre distintas secciones del contrato.
- **Recomendaciones de mejora:** Sugerencias para añadir cláusulas o modificar textos con el fin de reforzar la seguridad jurídica y la claridad.

Informe Jurídico:

1. **Sentido Jurídico:** El contrato tiene sentido jurídico general, ya que se establece un acuerdo entre dos partes (el prestador y la organización) para la prestación de servicios de consultoría tecnológica durante un período de 12 meses, con un precio y forma de pago especificados, así como obligaciones y compromisos de ambas partes.
2. **Faltan Cláusulas Esenciales:**
 - **Cláusula de Terminación:** Falta una cláusula que especifique las condiciones bajo las cuales el contrato podría ser terminado por cualquiera de las partes antes del plazo establecido.
 - **Cláusula de Responsabilidad:** Falta una cláusula que establezca la responsabilidad de cada parte en caso de incumplimiento del contrato.
 - **Cláusula de Modificación:** Falta una cláusula que establezca las condiciones bajo las cuales el contrato podría ser modificado.
 - **Cláusula de Disputas:** Falta una cláusula que establezca los procedimientos a seguir en caso de disputas entre las partes.

Figura 30- Informe jurídico automatizado con recomendaciones de mejora.

6.2.4 ANONIMIZACIÓN Y PROCESAMIENTO EXTERNO

Para proteger la privacidad de los datos sensibles incluidos en los contratos, *Themis* implementa un proceso de anonimización (Figura 31) automática antes de enviar cualquier texto a los modelos de lenguaje externos (como Mistral 7B vía OpenRouter).

Este proceso consiste en identificar y sustituir en el texto original todas las entidades sensibles (como nombres, documentos de identidad, teléfonos, direcciones, etc.) por *tokens*

neutros y únicos (por ejemplo, NOMBRE_1, DNI_1). Esta sustitución garantiza que ninguna información personal identificable sea expuesta durante el procesamiento por la IA.

Una vez que el modelo devuelve el texto generado o analizado, *Themis* realiza la operación inversa, reemplazando los *tokens* por los datos originales para asegurar que el resultado final mantenga la coherencia y precisión con la información real.

Nombre original	<i>Token</i> generado
Juan Pérez López	NOMBRE_1
DNI 12345678Z	DNI_1
Calle Falsa 123, Madrid	DIRECCION_1
Teléfono 600123456	TELEFONO_1

Tabla 10-Ejemplo de diccionario de anonimización utilizado.

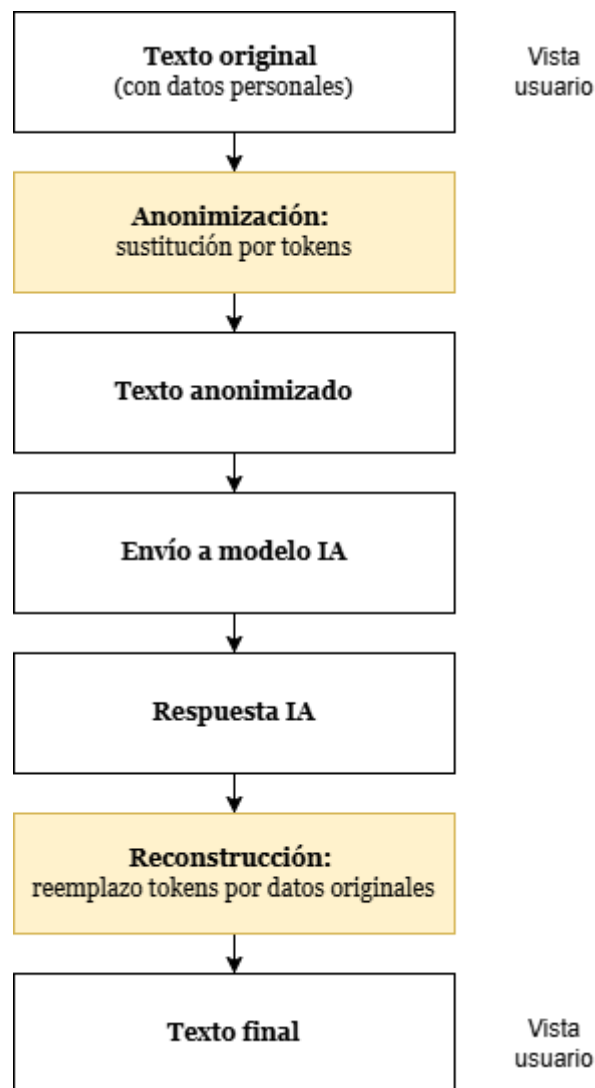
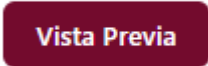


Figura 31-Esquema del proceso de anonimización y reconstrucción de datos sensibles.

6.2.5 RESULTADO GENERADO Y EXPORTACIÓN A PDF

La interfaz del Copilot de Contratos incluye un botón claramente identificado como “Vista previa” (Figura 32) “Exportar PDF” (Figura 33), que permite al usuario generar la versión final del contrato con un solo clic. Antes de exportar, el sistema realiza un renderizado (Figura 34) interno del documento, asegurando que el formato sea adecuado para su impresión y archivo.



Vista Previa

Figura 32-Botón vista previa para renderizar contrato.



Guardar Contrato PDF

Figura 33-Botón guardar contrato PDF para exportar el contrato.

Al pulsar el botón, se crea el archivo PDF con una estructura legal formal, que incluye encabezados, numeración, espacios para firmas (Figura 35) y demás elementos propios de un contrato profesional. El nombre del archivo se genera automáticamente, combinando datos relevantes del contrato.

Esta funcionalidad facilita la transición desde la redacción asistida hasta la entrega formal del documento, garantizando calidad visual y organizativa sin necesidad de pasos manuales adicionales.

CONTRATO DE PRESTACIÓN DE SERVICIOS

Entre Don/Doña Juan Pérez López, mayor de edad, con domicilio en Calle Falsa 123, Madrid, y DNI 1234567A, en adelante "EL CLIENTE",
y Don/Doña María García Martínez, mayor de edad, con domicilio en Avenida Siempre Viva 456, Barcelona, y DNI 98765432Z, en adelante
"EL PRESTADOR",

1. Objeto del contrato

EL PRESTADOR se compromete a prestar servicios de consultoría tecnológica para EL CLIENTE conforme a las especificaciones acordadas verbalmente.

2. Duración

El contrato tendrá una duración de 12 meses a partir del día 1 de Enero de 2024, hasta el 31 de diciembre de 2024.

3. Precio y forma de pago

EL CLIENTE abonará al PRESTADOR la cantidad de 10000 euros pagaderos en 12 mensualidades de 833,33 € cada una, por transferencia bancaria o cheque.

4. Obligaciones del PRESTADOR

Realizar los servicios de manera diligente y profesional

Figura 34-Contrato renderizado con estructura legal clara.

8. Jurisdicción

Las partes acuerdan someterse a la jurisdicción de los tribunales de Barcelona.

Firma del Comprador

Firma del Vendedor

Figura 35-Fragmento final del contrato exportado con cláusula de jurisdicción.

6.3 CASO DE USO: CONSULTA JURÍDICA EN LENGUAJE NATURAL

6.3.1 PREGUNTA DEL USUARIO

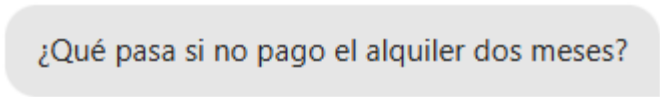
El usuario puede plantear consultas jurídicas en lenguaje natural sin necesidad de conocimientos técnicos o jurídicos avanzados. En la interfaz, dispone de la opción para elegir

entre un modo explicativo o un modo experto para recibir respuestas adaptadas a su nivel de conocimiento (ver Figura 36).



Figura 36-Selector de modo de respuesta en el asistente legal.

Por ejemplo, en la siguiente captura (Figura 37) se observa una consulta sobre el impago del alquiler:



¿Qué pasa si no pago el alquiler dos meses?

Figura 37-Consulta formulada en lenguaje natural sobre el impago del alquiler.

6.3.2 RESPUESTA DEL SISTEMA

Themis ofrece dos modos de respuesta para adaptarse al perfil y necesidades del usuario:

- **Modo explicativo** (Figura 38): La respuesta se presenta de forma detallada y accesible, con explicación clara de las consecuencias legales, referencias a artículos relevantes y un resumen comprensible para usuarios sin formación jurídica avanzada.

Si no pagas el alquiler dos meses, el arrendador podría tomar las siguientes medidas:

1. **Deuda en el subarrendatario:** Si el alquiler no se ha pagado y el arrendatario ha subarrendado la propiedad, el artículo 1552 del Código Civil establece que el subarrendatario queda obligado, en relación con el arrendador, por el importe del precio convenido en el subarriendo que se halle debiendo al tiempo del requerimiento, considerando no haberse cumplido con el pago. Esto significa que si el arrendatario no paga el alquiler, el subarrendatario podría ser demandado por el arrendador para que pague el alquiler de los dos meses no pagados.
2. **Demanda de restitución:** Si el arrendatario no paga el alquiler, el arrendador podría presentar una demanda de restitución para recuperar la propiedad arrendada. Esto significa que el arrendador podría pedir que se le devuelva la propiedad y que se le devuelvan los pagos de alquiler que se hayan hecho hasta ese momento.
3. **Reparación urgente:** Si durante el arrendamiento es necesario hacer alguna reparación urgente en la cosa arrendada que no pueda diferirse hasta la conclusión del arriendo, el arrendatario tiene la obligación de hacerla, según el artículo 1558 del Código Civil. Si el arrendatario no hace la reparación urgente, el arrendador podría hacerla por cuenta propia y cobrarle al arrendatario el costo de la misma.

En resumen, si no pagas el alquiler dos meses, el arrendador puede demandar al arrendatario o al subarrendatario para que pague el alquiler de los dos meses no pagados, podría pedir que se le devuelva la propiedad y que se le devuelvan los pagos de alquiler que se hayan hecho hasta ese momento, o podría hacer la reparación urgente por cuenta propia y cobrarle al arrendatario el costo de la misma.

Figura 38-Respuesta explicativa del asistente legal (impago de alquiler).

- **Modo experto (Figura 39):** La respuesta es más técnica, concentrándose en los aspectos legales concretos y las citas normativas específicas, ideal para profesionales que requieren información precisa y detallada sin explicaciones adicionales.

En virtud de la Ley Civil, si no se ha establecido un plazo específico para el arrendamiento y se ha fijado un alquiler mensual (Artículo 1581), el arrendamiento se entenderá por un período de meses.

Si el arrendatario no paga el alquiler durante dos meses, se incurrirá en un incumplimiento del contrato de arrendamiento. El arrendador podrá exigir el pago del importe del alquiler debido, considerando el importe convenido en el contrato (Artículo 1552).

Además, en virtud del mismo artículo, el arrendador podrá requerir al subarrendatario el pago del importe del alquiler que se halle debiendo al tiempo del requerimiento, si el subarrendatario está obligado a pagar el alquiler al arrendatario.

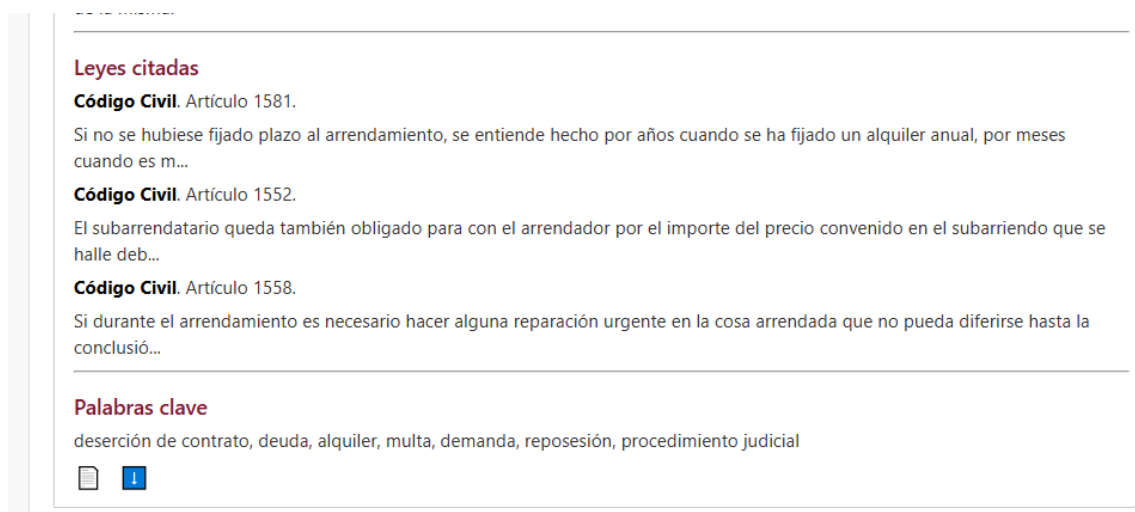
En cuanto a la obligación del arrendatario de realizar reparaciones urgentes en la cosa arrendada (Artículo 1558), si durante el arrendamiento es necesario hacer alguna reparación urgente que no pueda diferirse hasta la conclusión del arriendo, el arrendatario tiene la obligación de hacerlas. Sin embargo, la falta de pago del alquiler no se relaciona directamente con esta obligación de reparación.

En resumen, si el arrendatario no paga el alquiler durante dos meses, el arrendador podrá exigir el pago del importe del alquiler debido y requerir al subarrendatario, si lo hubiera, el pago del importe del alquiler que se halle debiendo al tiempo del requerimiento. Además, el arrendatario debe hacer reparaciones urgentes en la cosa arrendada si son necesarias, pero la falta de pago del alquiler no se relaciona directamente con esta obligación.

Figura 39-Respuesta experta del asistente legal (impago de alquiler).

6.3.1 LEYES CITADAS

El sistema proporciona también un listado con los textos legales consultados para fundamentar la respuesta (Figura 40), permitiendo al usuario verificar y consultar directamente las fuentes normativas.



Leyes citadas

Código Civil. Artículo 1581.
Si no se hubiese fijado plazo al arrendamiento, se entiende hecho por años cuando se ha fijado un alquiler anual, por meses cuando es m...

Código Civil. Artículo 1552.
El subarrendatario queda también obligado para con el arrendador por el importe del precio convenido en el subarriendo que se halle deb...

Código Civil. Artículo 1558.
Si durante el arrendamiento es necesario hacer alguna reparación urgente en la cosa arrendada que no pueda diferirse hasta la conclusió...

Palabras clave

deserción de contrato, deuda, alquiler, multa, demanda, reposición, procedimiento judicial

Figura 40-Listado de leyes y artículos legales citados por el asistente jurídico.

6.4 ANÁLISIS DE FLUJO DE INTERACCIÓN

El flujo de interacción en *Themis* (Figura 41) comienza con la solicitud del usuario, que se realiza a través del *frontend* desarrollado en React. Esta solicitud es enviada al *backend* en Flask, donde se aplica un procesamiento inicial con spaCy, encargado de anonimizar los datos sensibles sustituyéndolos por *tokens* neutros para proteger la privacidad. A continuación, el texto anonimizado se envía al modelo de lenguaje avanzado accesible mediante OpenRouter (como Mistral 7B), que genera la respuesta contextualizada. Finalmente, el *backend* reconstruye los datos originales en la respuesta y la envía al usuario, completando el ciclo de interacción garantizando seguridad y coherencia.

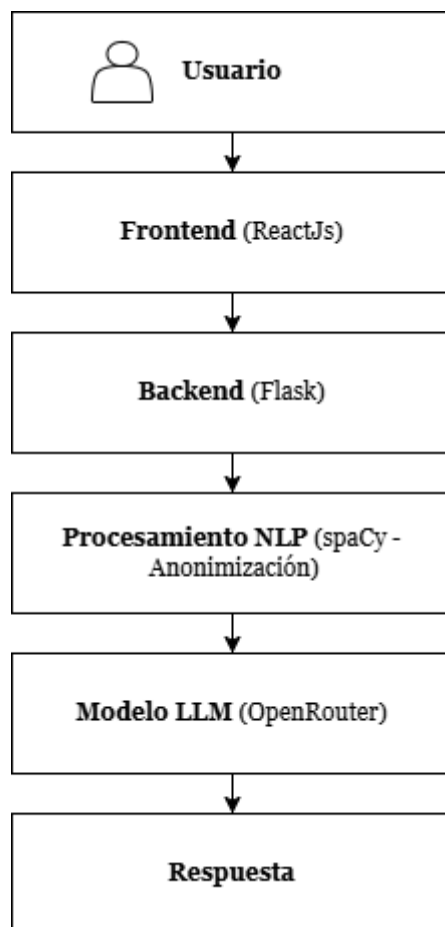


Figura 41-Esquema del flujo de interacción en *Themis*.

6.5 DEMOSTRACIÓN GRABADA

Para facilitar la comprensión práctica del funcionamiento de *Themis*, se ha preparado una demostración grabada que recorre los principales módulos del sistema. El vídeo incluye la redacción asistida de contratos, la detección automática de errores, la exportación a PDF y la consulta jurídica en lenguaje natural mediante el asistente conversacional. Para ver los videos se recomienda ver el Github.

Se recomienda ver el vídeo para apreciar la fluidez de la interfaz y la capacidad real del sistema en un entorno simulado.

Capítulo 7. CONCLUSIONES Y ANÁLISIS DE RESULTADOS

El desarrollo de *Themis* ha supuesto la construcción de un sistema LegalTech funcional, modular y centrado en la automatización de tareas jurídicas, con especial atención a la redacción de contratos y la resolución de consultas legales mediante lenguaje natural. El enfoque seguido ha combinado viabilidad técnica, sensibilidad jurídica y enfoque práctico, lo que ha permitido construir un producto coherente, intuitivo y adaptado a escenarios reales.

A través de una arquitectura clara y un desarrollo iterativo, se han abordado con éxito algunos de los retos más relevantes en la digitalización del trabajo legal: la estructuración del lenguaje jurídico, la protección de datos sensibles, la generación de contenido formal y la integración de tecnologías NLP sin comprometer el control ni la calidad documental. El sistema ha sido diseñado con visión de continuidad, garantizando su modularidad y posibilidad de evolución a largo plazo.

7.1 *QUÉ APORTA THEMIS AL MERCADO ACTUAL*

Themis se sitúa en un punto intermedio entre los grandes productos LegalTech corporativos y los asistentes genéricos basados en IA no especializados. Frente a las plataformas tradicionales, ofrece:

- Un enfoque más flexible y adaptativo en la redacción jurídica, eliminando la rigidez de los formularios tipo.
- Un sistema real de asistencia con generación semántica contextual y sugerencias formales personalizadas.

- Un módulo conversacional jurídico que responde en lenguaje natural con base normativa, sin necesidad de que el usuario conozca el marco legal de forma explícita.
- Verificación automática de datos sensibles, que facilita la detección temprana de errores comunes (DNI incorrecto, duplicidades, omisiones).
- Una arquitectura abierta y asequible, diseñada para facilitar pruebas, ampliaciones y adaptaciones a distintos casos de uso sin necesidad de infraestructuras costosas.

En un mercado donde muchas soluciones requieren licencias cerradas, plantillas fijas o infraestructuras complejas, *Themis* representa una propuesta funcional, accesible y técnicamente sólida para entornos profesionales que buscan digitalizar sus procesos sin renunciar al control del contenido.

Esta propuesta diferenciadora se puede observar en detalle en la Tabla 11, donde se comparan las funcionalidades clave de *Themis* frente a otras herramientas LegalTech ampliamente implantadas. El análisis refleja que, sin pretender competir directamente en escala, *Themis* resuelve de forma unificada y extensible varios aspectos que hoy se cubren de forma parcial, fragmentada o con limitaciones de acceso.

Criterio	LexisNexis Westlaw	Contract Express / DocuSign	Asistentes Genéricos (ChatGPT, Gemini)	<i>Themis</i>
Redacción jurídica contextual	Plantillas cerradas	Limitada a plantillas prediseñadas	Capacidad genérica, sin contexto legal	Libre y sugerencias adaptadas
Anonimización automática de datos	No disponible	No disponible	No incluida	spaCy con reconstrucción inteligente
Verificación de datos sensibles	No disponible	No disponible	No disponible	Validación automática (DNI, teléfono, etc.)
Recuperación semántica de normativa	Búsqueda por palabra clave	No disponible	Depende del modelo	<i>Embeddings</i> legales + búsqueda vectorial
Asistente conversacional jurídico	No interactivo	No disponible	No especializado	Basado en ley real, estructurado
Exportación profesional de contratos	Sí	Sí	No, (requiere integración externa)	PDF con formato legal y nombres semánticos
Adaptabilidad / código abierto	Plataforma cerrada	Servicios comerciales cerrados	Dependencia de proveedores	Modular, ampliable y personalizable
Coste de uso para prototipado	Licencias comerciales	Coste por usuario o integración	Gratuito (pero limitado)	Gratuito y sostenible con OpenRouter

Tabla 11- Comparativa funcional entre *Themis* y soluciones LegalTech existentes.

7.2 TRABAJO FUTURO

Aunque *Themis* se presenta como una plataforma operativa, estructurada, escalable y funcional en todos sus módulos principales, existen algunos aspectos específicos identificados durante el desarrollo que pueden ser implementados o mejorados en una fase de evolución real del producto. Estos puntos no corresponden a funcionalidades por implementar desde cero, sino realmente a extensiones naturales del sistema actual, derivadas tanto de restricciones de recursos como del enfoque iterativo adoptado.

Las líneas de trabajo futuro más relevantes son:

- **Integración directa de jurisprudencia.** Actualmente el sistema redirige al buscador del Consejo General del Poder Judicial para realizar búsquedas manuales. Una ampliación clave sería automatizar la recuperación y análisis semántico de resoluciones judiciales, permitiendo que el Asistente Legal no solo fundamente sus respuestas en normativa vigente, sino también en interpretación jurisprudencial. Para ello, sería necesario contar con una API oficial o sistemas autorizados de *scraping* controlado, así como adoptar el *pipeline* de *embeddings* para sentencias judiciales (que necesitaría tener una estructura diferente a los artículos legales ya guardados).
- **Sistema de autenticación y trabajo multiusuario.** En una versión profesionalizada, se prevé incorporar la gestión de usuarios autenticados y perfiles diferenciados (por ejemplo, abogados senior, paralegales, administradores...) Esto permitiría guardar el histórico de contratos, sesiones del asistente legal y tareas asociadas a cada profesional.
- **Optimización del modelo generativo.** El uso actual de Mistral 7B vía OpenRouter ha permitido mantener el equilibrio entre calidad de resultados y viabilidad económica, especialmente durante las fases de prueba intensiva. Sin embargo, se ha detectado que ciertos escenarios podrían requerir un modelo con mayor capacidad de razonamiento y adaptación al contexto (como GPT-4). En un entorno con acceso a *tokens* de producción o licencias empresariales, se plantea realizar A/B testing con modelos avanzados para casos jurídicos no estructurados.

- **Validación jurídica automática avanzada.** El sistema actual ya permite detectar errores de formato, duplicados o datos inconsistentes. Una evolución razonable sería incorporar validadores semánticos, capaces de identificar contradicciones normativas, cláusulas mal redactadas, omisiones frecuentes o desalineaciones entre las secciones del contrato.
- **Despliegue de web seguro y público.** Aunque actualmente se ejecuta en entorno local, el sistema está preparado para ser desplegado como una aplicación accesible vía navegador. Esto incluiría:
 - *Backend* en entorno virtualizado (Docker)
 - Integración con HTTPs, control de sesiones y protección.
- **Gestor inteligente de tareas jurídicas.** Aunque esta funcionalidad no se ha incluido en la versión final de *Themis*, durante el desarrollo se ha implementado parcialmente la lógica del gestor en el *backend*. Este componente permitiría organizar tareas asociadas a contratos y flujos legales de forma estructurada, con seguimiento de estado, alertas y trazabilidad. Se considera una extensión natural del sistema, especialmente útil en versiones destinadas a entornos profesionales o colaborativos.

Estas líneas de trabajo futuro están pensadas desde la continuidad, no como adiciones especulativas y permitirían evolucionar *Themis* desde una herramienta académicamente funcional hacia un producto profesional orientado a pequeños despachos, departamentos jurídicos o, incluso, centrándonos en el Asistente Legan y el Copilot de Contratos, en entornos de empresas, académicos y/o formativos.

7.1 LIMITACIONES Y CONSIDERACIONES TÉCNICAS CLAVE

Durante el desarrollo y documentación de *Themis*, se han identificado varias cuestiones que, aunque no comprometen la funcionalidad general del sistema, merecen una explicación detallada para evitar malentendidos y garantizar transparencia respecto a sus capacidades y limitaciones.

7.1.1 PRIVACIDAD POR DISEÑO Y USO DE MODELOS EXTERNOS

Themis aplica un enfoque de privacidad por diseño mediante la anonimización automática previa al envío de cualquier dato sensible a servicios externos, como el modelo Mistral 7B accesible vía OpenRouter. Este proceso, implementado con spaCy, sustituye las entidades personales por *tokens* neutros, minimizando el riesgo de exposición. Sin embargo, dado que el procesamiento se realiza en servidores externos, no se garantiza un control total sobre el tratamiento de los datos durante este paso. Por ello, se recomienda que la plataforma se utilice en entornos con políticas de privacidad adecuadas y que se evalúe la incorporación futura de modelos locales para mitigar este riesgo.

7.1.2 ANONIMIZACIÓN PREVIA Y RECONSTRUCCIÓN EXACTA

El sistema realiza una reconstrucción precisa de los datos originales mediante la sustitución de *tokens* tras recibir la respuesta del modelo. Aunque en casos excepcionales con formatos muy atípicos pueda haber pequeñas variaciones en la restitución, *Themis* ofrece un nivel de precisión que facilita enormemente la revisión final. Por este motivo, la herramienta está concebida como un complemento que potencia y agiliza el trabajo del profesional jurídico, reforzando su control y decisión en la validación definitiva de los documentos.

7.1.3 CAPACIDAD DEL ASISTENTE LEGAL PARA MANTENER CONTEXTO

Si bien el asistente legal utiliza información contextual predefinida para adaptar sus respuestas, actualmente no mantiene un historial dinámico de conversación entre interacciones. Esto implica que cada consulta se procesa de manera independiente, lo que limita la continuidad conversacional. Se planea evaluar futuras mejoras que permitan incorporar memoria de sesión para mejorar la coherencia en diálogos extendidos.

7.1.4 LIBERTAD VS. ESTRUCTURA EN LA REDACCIÓN

Aunque *Themis* ofrece un entorno de redacción flexible y sin formularios rígidos, la generación de texto se guía mediante *prompts* y etiquetas que estructuran y condicionan la

respuesta del modelo para mantener la coherencia jurídica. Esta combinación busca equilibrar la libertad creativa con el control necesario para garantizar calidad y precisión, pero puede parecer en ocasiones contradictoria. Se recomienda comunicar claramente este balance a los usuarios para ajustar sus expectativas.

7.1.5 AFIRMACIONES CATEGÓRICAS Y VERIFICACIÓN LEGAL PRELIMINAR

El sistema integra mecanismos automáticos para la detección de errores formales y omisiones comunes sin intervención humana, acelerando la revisión inicial. Sin embargo, no reemplaza la evaluación jurídica profesional, que sigue siendo necesaria para validar el documento en su conjunto. Asimismo, aunque se diseñó para minimizar ambigüedades, la generación libre de texto depende en gran medida del diseño del prompt y la calidad del modelo, por lo que se admiten limitaciones inherentes.

7.1.6 USO PARA USUARIOS NO EXPERTOS

Themis está orientado a facilitar el trabajo jurídico incluso a usuarios con conocimientos limitados, ofreciendo explicaciones y referencias normativas. No obstante, algunas respuestas incluyen citas legales que requieren interpretación experta, por lo que el sistema debe usarse como apoyo y no como sustituto del asesoramiento legal profesional.

Capítulo 8. BIBLIOGRAFÍA

- [1] Katz, D. M., Bommarito, M. J., & Blackman, J. (2017). A general approach for predicting the behavior of the Supreme Court of the United States. *PLoS ONE*, 12(4), e0174698. <https://doi.org/10.1371/journal.pone.0174698>
- [2] Chandrasekaran, S. Gurusamy, S.; Kannan, R. "Advancing Legal Research Using Natural Language Processing." *International Journal of Information Management*, vol. 56, 2021. <https://www.sciencedirect.com/science/article/abs/pii/S0268401220314304?via%3Dihub>
- [3] Susskind, R. *Tomorrow's Lawyers: An Introduction to Your Future*. Oxford University Press, 2nd ed., 2019.
- [4] Zhong, R., Hu, X., & Zhao, W. (2020). Legal document generation via machine learning. *IEEE Transactions on Artificial Intelligence*, 2(3). <https://doi.org/10.1109/TAI.2020.3023157>
- [5] Pilkington, N. (2021). Workflow automation in LegalTech: Emerging trends and opportunities. *Legal Innovation Journal*, 10(2).
- [6] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*. <https://aclanthology.org/N19-1423>
- [7] Loeffler, B. (2011, October). Cloud computing: What is infrastructure as a service. *Microsoft Technet Magazine*. <https://technet.microsoft.com/en-us/magazine/hh509051.aspx>
- [8] spaCy. (2024). *spaCy: Industrial-strength Natural Language Processing in Python*. <https://spacy.io>
- [9] Hugging Face. (2024). *Transformers: State-of-the-art Machine Learning for NLP*. <https://huggingface.co/transformers>
- [10] SentenceTransformers. (2024). *Documentation and usage*. <https://www.sbert.net>
- [11] FAISS – Facebook AI Similarity Search. (2024). <https://github.com/facebookresearch/faiss>
- [12] OpenRouter. (2024). *API Access to Open Source LLMs*. <https://openrouter.ai/docs>
- [13] jsPDF. (2024). *Generate PDF documents from the browser*. <https://github.com/parallax/jsPDF>
- [14] ReactJS. (2024). *React – A JavaScript library for building user interfaces*. <https://reactjs.org/>
- [15] Flask. (2024). *Flask documentation*. <https://flask.palletsprojects.com>

- [16] Git. (2024). *Git: Distributed version control system*. <https://git-scm.com>
- [17] SQLAlchemy. (2024). *The Database Toolkit for Python*. <https://www.sqlalchemy.org>
- [18] Contract Express – Thomson Reuters. (2024). <https://legal.thomsonreuters.com>
- [19] DocuSign. (2024). *DocuSign: Contract lifecycle management*. <https://www.docusign.com>
- [20] Westlaw – Thomson Reuters. (2024). <https://legal.thomsonreuters.com/en/westlaw>
- [21] LexisNexis. (2024). *Lexis+: The future of legal research*. <https://www.lexisnexis.com>
- [22] MyCase. (2024). *Law practice management software*. <https://www.mycase.com>
- [23] Clio. (2024). *Legal practice management platform*. <https://www.clio.com>
- [24] El Confidencial. (2023, abril 4). Los bufetes demandan soluciones que entiendan el proceso legal, no solo productividad. El Confidencial. <https://www.elconfidencial.com>
- [25] The Guardian. (2024, February 12). Westlaw and LexisNexis remain powerful, but complex: LegalTech must democratize. *The Guardian Legal Column*. <https://www.theguardian.com>
- [26] Instituto de Ingeniería del Conocimiento. (2024, enero). MEL: Una alternativa a los modelos generativos en el sector legal. *Procesamiento del Lenguaje Natural, IIC-UAM*. <https://www.iic.uam.es/procesamiento-del-lenguaje-natural/mel-alternativa-a-modelos-generativos-sector-legal/>
- [27] Navarro, I. (2023, octubre 9). ¿Quién es Harvey, IAG para abogados? *Expansión*. <https://www.expansion.com/blogs/navarro/2023/10/09/quien-es-harvey-iag-para-abogados.html>

ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS

El proyecto “*Desarrollo de un toolkit para la automatización y optimización de tareas legales basado en IA*”, a través de la plataforma *Themis*, se alinea con varios de los Objetivos de Desarrollo Sostenible (ODS) definidos por la Organización de las Naciones Unidas, aportando valor no solo tecnológico, sino también social e institucional dentro del ámbito jurídico. En particular, *Themis* contribuye de forma específica a los siguientes:

ODS 8: Trabajo decente y crecimiento económico

El sistema promueve la mejora de la eficiencia operativa en el trabajo jurídico, reduciendo la carga de tareas repetitivas mediante automatización (como la redacción de contratos o la validación documental). Esta optimización permite a los profesionales centrarse en tareas de mayor valor añadido, fomentando condiciones de trabajo más sostenibles y productivas, así como nuevas oportunidades de especialización tecnológica dentro del sector legal.

ODS 9: Industria, innovación e infraestructura

El desarrollo de *Themis* introduce soluciones basadas en Inteligencia Artificial y Procesamiento de Lenguaje Natural aplicadas a la gestión legal. Al incorporar funcionalidades como generación de texto contextualizado, verificación automatizada de datos o recuperación semántica de normativa, el proyecto contribuye a la digitalización del ecosistema jurídico y a la construcción de una infraestructura técnica sólida para la gestión del conocimiento jurídico.

ODS 16: Paz, justicia e instituciones sólidas

La plataforma busca mejorar el acceso a la justicia y la transparencia institucional mediante herramientas que agilizan procesos, reducen errores y permiten una mayor trazabilidad

documental. La asistencia jurídica conversacional basada en leyes reales, la validación automatizada de contratos y la accesibilidad técnica favorecen un entorno legal más equitativo, moderno y comprensible, tanto para profesionales como para ciudadanos.

En conjunto, *Themis* no solo representa un avance desde el punto de vista tecnológico, sino que contribuye activamente al fortalecimiento institucional y al desarrollo sostenible del sector jurídico, en línea con los principios promovidos por la Agenda 2030.

ANEXO II-FRAGMENTO LEGAL

ALMACENADO

```
[
{
  "texto": "Artículo 1. Objeto y finalidad.\n1. Esta ley establece las bases que deben regir la evaluación ambiental de los planes, programas y proyectos que puedan tener efectos significativos sobre el medio ambiente, garantizando en todo el territorio del Estado un elevado nivel de protección ambiental, con el fin de promover un desarrollo sostenible, mediante:\n(a) La integración de los aspectos medioambientales en la elaboración y en la adopción, aprobación o autorización de los planes, programas y proyectos;\n(b) el análisis y la selección de las alternativas que resulten ambientalmente viables;\n(c) el establecimiento de las medidas que permitan prevenir, corregir y, en su caso, compensar los efectos adversos sobre el medio ambiente;\n(d) el establecimiento de las medidas de vigilancia, seguimiento y sanción necesarias para cumplir con las finalidades de esta ley.\n2. Asimismo, esta ley establece los principios que informarán el procedimiento de evaluación ambiental de los planes, programas y proyectos que puedan tener efectos significativos sobre el medio ambiente, así como el régimen de cooperación entre la Administración General del Estado y las comunidades autónomas a través de la Conferencia Sectorial de Medio Ambiente.\nSubir\n[Bloque 4: #a2]",
  "ley_id": "Ley de Evaluación Ambiental",
  "articulo": "1."
},
{
  "texto": "Artículo 2. Principios de la evaluación ambiental.\nLos procedimientos de evaluación ambiental se sujetarán a los siguientes principios:\n(a) Protección y mejora del medio ambiente.\n(b) Precaución y acción cautelar.\n(c) Acción preventiva, corrección y compensación de los impactos sobre el medio ambiente.\n(d) Quien contamina paga.\n(e) Racionalización, simplificación y concertación de los procedimientos de evaluación ambiental.\n(f) Cooperación y coordinación entre la Administración General del Estado y las Comunidades Autónomas.\n(g) Proporcionalidad entre los efectos sobre el medio ambiente de los planes, programas y proyectos, y el tipo de procedimiento de evaluación al que en su caso deban someterse.\n(h) Colaboración activa de los distintos órganos administrativos que intervienen en el procedimiento de evaluación, facilitando la información necesaria que se les requiera.\n(i) Participación pública.\n(j) Desarrollo sostenible.\n(k) Integración de los aspectos ambientales en la toma de decisiones.\n(l) Actuación de acuerdo al mejor conocimiento científico posible.\nSe modifican las letras b) y c) por el art. único.1 de la Ley 9/2018, de 5 de diciembre. Ref. BOE-A-2018-16674\nÚltima actualización, publicada el 06/12/2018, en vigor a partir del 07/12/2018.\nTexto original, publicado el 11/12/2013, en vigor a partir del 12/12/2013.\nSubir\n[Bloque 5: #a3]",
  "ley_id": "Ley de Evaluación Ambiental",
  "articulo": "2."
}
```

```
},  
{  
  "texto": "Artículo 3. Actuación y relaciones entre Administraciones  
Públicas.\n1. Las Administraciones Públicas ajustarán sus actuaciones en materia  
de evaluación ambiental a los principios de lealtad institucional, coordinación,  
información mutua, cooperación, colaboración y coherencia.\nEn particular, las  
Administraciones que puedan estar interesadas en el plan, programa o proyecto  
debido a sus responsabilidades medioambientales específicas o a sus competencias  
autonómicas o locales serán consultadas sobre la información proporcionada por el  
promotor y sobre la solicitud de adopción, aprobación o autorización del plan,  
programa o proyecto.\nDe conformidad con los principios de racionalización y  
agilidad de los procedimientos administrativos y de cooperación, colaboración y  
coordinación, las Administraciones Públicas consultadas emitirán los informes que  
correspondan con la máxima diligencia posible y, en todo caso, dentro de los  
plazos establecidos al efecto.\nCuando corresponda a la Administración General  
del Estado formular la declaración ambiental estratégica o la declaración de  
impacto ambiental, o bien emitir el informe ambiental estratégico o el informe de  
impacto ambiental regulados en esta ley, se consultará preceptivamente al órgano  
que ostente las competencias en materia de medio ambiente de la comunidad  
autónoma afectada por el plan, programa o proyecto.\n2. Las Administraciones  
Públicas garantizarán que el órgano ambiental y el órgano sustantivo ejerzan las  
funciones derivadas de la presente ley de manera objetiva, y aplicarán en su  
organización una adecuada separación de las funciones que puedan dar lugar a un  
conflicto de intereses cuando el órgano ambiental sea simultáneamente el órgano  
sustantivo o el promotor del plan, programa o proyecto.\n3. Cuando el órgano  
sustantivo sea simultáneamente el promotor del plan, programa o proyecto, el  
órgano sustantivo realizará las actuaciones atribuidas al promotor en esta  
ley.\n4. El órgano sustantivo informará al órgano ambiental de cualquier  
incidencia que se produzca durante la tramitación del procedimiento  
administrativo sustantivo de adopción, aprobación o autorización de un plan,  
programa o proyecto que tenga relevancia a los efectos de la tramitación de la  
evaluación ambiental, singularmente aquellas que supongan el archivo o la  
caducidad del procedimiento sustantivo.\nSe modifica por el art. único.2 de la  
Ley 9/2018, de 5 de diciembre. Ref. BOE-A-2018-16674\nÚltima actualización,  
publicada el 06/12/2018, en vigor a partir del 07/12/2018.\nTexto original,  
publicado el 11/12/2013, en vigor a partir del 12/12/2013.\nSubir\n[Bloque 6:  
#a4]",  
  "ley_id": "Ley de Evaluación Ambiental",  
  "articulo": "3."  
},
```

Código 9-Fragmento de ley de evaluación ambiental fragmentada en formato json.