

Universidad de La Habana
Facultad de Matemática y Computación



Construcción Incremental de Bases de Conocimiento Semi-estructuradas con Modelos de Lenguaje

Autor: Carlos Mauricio Reyes Escudero

Tutor: Dr.C Alejandro Piad Morffis

Trabajo de Diploma presentado en opción al título de
Licenciado en Ciencia de la Computación

Junio de 2025

Índice general

1. Introducción	1
1.1. Gestión del Conocimiento Personal (PKM)	2
1.1.1. Toma de notas (Note-taking)	2
1.1.2. Bases de Conocimiento Personal (PKB)	3
1.1.3. Sistemas Digitales de Toma de Notas	4
1.2. Grandes Modelos de Lenguaje (LLM)	5
1.2.1. Técnicas de Prompting	5
1.2.2. Agentes Basados en LLM	7
1.3. Integración de Herramientas Basadas en LLM en Aplicaciones de Toma de Notas	8
1.4. Objetivos de la Tesis	10
Bibliografía	12

Capítulo 1

Introducción

La sobrecarga de información en la actualidad plantea un desafío significativo para la productividad y el aprendizaje efectivo. En este contexto, la gestión del conocimiento personal emerge como una disciplina fundamental, no solo para académicos e investigadores, sino para cualquier individuo que busque optimizar su capacidad de aprendizaje y generación de ideas. Como señala Ahrens, *How to Take Smart Notes: One Simple Technique to Boost Writing, Learning and Thinking – for Students, Academics and Nonfiction Book Writers*, una gestión eficaz de las notas y el conocimiento adquirido no solo facilita la escritura y el estudio, sino que transforma la manera en que interactuamos con la información, convirtiéndola en un activo dinámico y generador de nuevas perspectivas. El desarrollo de un *segundo cerebro* (Forte, *Building a Second Brain*), un sistema externo confiable para almacenar y conectar ideas, libera recursos cognitivos, permitiendo un enfoque más profundo en el pensamiento crítico y la creatividad. El potencial de los Grandes Modelos de Lenguaje (LLM) para revolucionar este campo es inmenso. Estas tecnologías ofrecen la posibilidad de automatizar y enriquecer la integración del conocimiento, asistiendo en la destilación de información, la identificación de conexiones y la generación de contenido relevante dentro de las bases de conocimiento personales. La automatización de estos procesos no solo promete un aumento en la eficiencia, sino también una democratización del acceso a metodologías avanzadas de gestión del conocimiento.

1.1. Gestión del Conocimiento Personal (PKM)

La Gestión del Conocimiento Personal (PKM, por sus siglas en inglés) se define como el proceso mediante el cual un individuo recopila, clasifica, almacena, busca, recupera y comparte conocimiento en sus actividades diarias (Grundspenkis, «Agent Based Approach for Organization and Personal Knowledge Modelling»). Esta disciplina es particularmente relevante para los denominados *trabajadores del conocimiento* (Knowledge Workers), profesionales para quienes el conocimiento es su activo más valioso y que dedican una parte significativa de su tiempo a gestionar grandes cantidades de información. La PKM busca empoderar a estos individuos para que sean más efectivos en sus entornos personales, organizacionales y sociales.

1.1.1. Toma de notas (Note-taking)

La toma de notas, lejos de ser un mero acto de transcripción, es una estrategia fundamental para mejorar el aprendizaje y la retención de información (Jansen et al., «An Integrative Review of the Cognitive Costs and Benefits of Note-Taking»). Al tomar notas, el individuo no solo registra la esencia de la información, sino que también se involucra en un proceso activo de filtrado, organización y reestructuración del conocimiento.

Existen diversas metodologías para la toma de notas, que pueden clasificarse ampliamente en lineales y no lineales. Las metodologías lineales implican registrar la información en el orden en que se recibe. Un ejemplo común es la creación de esquemas (Outlining), donde las notas y pensamientos se organizan de manera estructurada y lógica. Otra técnica lineal es el método de frases (Sentence method), que consiste en anotar cada tema como una oración corta y simple, ideal para lecciones de ritmo rápido.

Las metodologías no lineales, por otro lado, utilizan la organización espacial y los diagramas para ensamblar la información. Entre estas se encuentra el Charting, útil para temas que pueden desglosarse en categorías. El mapeo (Mapping), como los mapas mentales, organiza las ideas en una estructura de árbol a partir de un punto central. Las Notas Cornell, desarrolladas por Walter Pauk (Pauk et al., *How to Study in College*), dividen la página en secciones para notas, pistas y un resumen. El método SQ3R (Survey, Question, Read, Recite, Review) es más una técnica de lectura comprensiva que implica generar preguntas y luego tomar notas

para responderlas. Las notas guiadas (Guided notes) proporcionan un esquema predefinido con puntos clave faltantes que el estudiante completa. Más allá de estas técnicas tradicionales, han surgido sistemas más sofisticados. El método Zettelkasten, popularizado por Niklas Luhmann y descrito en detalle por Ahrens, *How to Take Smart Notes: One Simple Technique to Boost Writing, Learning and Thinking – for Students, Academics and Nonfiction Book Writers*, se basa en la creación de notas atómicas interconectadas, formando una red de conocimiento que fomenta la generación de ideas y la escritura prolífica. Este sistema, aunque con raíces históricas, ha ganado nueva relevancia en la era digital. Por otro lado, el método PARA (Projects, Areas, Resources, Archives), propuesto por Forte, *Building a Second Brain*, ofrece un marco para organizar la información digital en función de su accionabilidad, facilitando la gestión de proyectos y responsabilidades a largo plazo.

1.1.2. Bases de Conocimiento Personal (PKB)

Una Base de Conocimiento Personal (PKB, por sus siglas en inglés) es una herramienta electrónica utilizada por un individuo para expresar, capturar y recuperar conocimiento personal. Se diferencia de una base de datos tradicional al contener material subjetivo y específico del propietario. El concepto de extender la memoria y las capacidades cognitivas del individuo mediante herramientas externas no es nuevo. Ya en 1945, Vannevar Bush imaginó el *Memex* (memory extension), un dispositivo electromecánico en el que un individuo almacenaría todos sus libros, registros y comunicaciones, mecanizado de tal manera que pudiera ser consultado con gran velocidad y flexibilidad (Bush, *As We May Think*). Bush previó un futuro donde la sobrecarga de información requeriría nuevas formas de acceder y conectar el conocimiento acumulado.

La forma en que una PKB organiza el conocimiento se define por su modelo de datos. Estudios como los de Davies y colegas (Davies et al., «*Building the Memex Sixty Years Later: Trends and Directions in Personal Knowledge Bases*»; Davies, «*Still Building the Memex*») han analizado estos modelos en función de su marco estructural (cómo se interrelacionan los elementos), los elementos de conocimiento (las unidades básicas de información) y su esquema (el nivel de semántica formal). Un aspecto crucial destacado es la transclusión, la capacidad de ver el mismo elemento de conocimiento en múltiples contextos sin duplicación.

Grafos de Conocimiento Personal (PKG). Dentro de las PKB, los Grafos de Conocimiento Personal (PKG, por sus siglas en inglés) han ganado prominencia. Un PKG representa el conocimiento como una red de nodos interconectados, donde cada nodo es una pieza de información y las aristas representan las relaciones entre ellas, a menudo con visualizaciones gráficas que facilitan la exploración y el descubrimiento de conexiones (Pyne et al., «[Meta-Work](#)»).

1.1.3. Sistemas Digitales de Toma de Notas

La visión del Memex de Bush encuentra un eco contemporáneo en la plétora de sistemas digitales de toma de notas. Estos sistemas varían ampliamente en sus características y enfoques. Algunos, como Roam Research y Logseq, enfatizan los enlaces bidireccionales y la visualización gráfica, alineándose estrechamente con la idea de un PKG. Otros, como Notion, ofrecen una gran flexibilidad para crear bases de datos y vistas personalizadas, mientras que Evernote y Google Keep son populares por su simplicidad y accesibilidad multiplataforma. Microsoft OneNote se integra bien con el ecosistema de Microsoft. Alternativas de código abierto como Zettlr y Joplin también ofrecen robustas funcionalidades, a menudo con un enfoque en la privacidad y el control local de los datos. Herramientas más tradicionales como VimWiki (para usuarios de Vim) y OrgMode (para Emacs) ofrecen sistemas de toma de notas altamente personalizables y potentes para usuarios con conocimientos técnicos. Markor y SimpleNote se centran en la simplicidad y la edición Markdown.

Entre estas herramientas, Obsidian.md ha ganado una considerable popularidad para la escritura académica y la gestión del conocimiento personal. Siguiendo el modelo de datos de las PKB, Obsidian almacena las notas como archivos locales de texto plano en formato Markdown, lo que garantiza la portabilidad y la longevidad de los datos. Su marco estructural se basa en un grafo de conocimiento, donde cada nota es un nodo y los enlaces bidireccionales permiten crear una red interconectada de información. Esto facilita la transclusión, ya que una misma idea o nota puede ser referenciada y contextualizada desde múltiples puntos del grafo sin necesidad de duplicación. Los elementos de conocimiento son flexibles, desde conceptos simples hasta notas extensas. Obsidian también soporta un esquema enriquecido mediante el uso de metadatos (frontmatter), etiquetas y la posibilidad de extender su funcionalidad mediante plugins, como Dataview, que permite realizar consultas complejas sobre las notas. Su vista de grafo visualiza

las conexiones, ayudando a identificar relaciones y patrones emergentes.

1.2. Grandes Modelos de Lenguaje (LLM)

Los Grandes Modelos de Lenguaje (LLM) representan un avance transformador en el campo de la inteligencia artificial, con la capacidad de comprender, generar y manipular el lenguaje natural a niveles sin precedentes. Modelos como GPT-4 de OpenAI (OpenAI et al., [GPT-4 Technical Report](#)) y Gemini de Google (Team et al., [Gemini](#)) han demostrado habilidades notables en una amplia gama de tareas lingüísticas. En paralelo, la comunidad de código abierto ha respondido con modelos competitivos como DeepSeek (DeepSeek-AI et al., [DeepSeek-V3 Technical Report](#)), Qwen (Bai et al., [Qwen Technical Report](#)) y LLaMa (Grattafiori et al., [The Llama 3 Herd of Models](#)), impulsando la innovación y la accesibilidad en este campo.

1.2.1. Técnicas de Prompting

Para interactuar eficazmente con estos modelos, se ha desarrollado un sinfín de técnicas de *prompting* (formulación de instrucciones). Inicialmente, se descubrió que proporcionar unos pocos ejemplos (few-shot learning) mejoraba significativamente los resultados, apelando a la capacidad de generalización de los LLM sin necesidad de reentrenamiento (Brown et al., [«Language Models Are Few-Shot Learners»](#)). Investigaciones posteriores, como la de Min et al., [«Rethinking the Role of Demonstrations»](#), han explorado qué partes de estos ejemplos son cruciales, sugiriendo que la estructura del texto, el formato o la forma de proceder pueden ser más importantes que la etiqueta correcta para que el modelo aprenda la tarea. Con el aumento de la ventana de contexto de los modelos más recientes, el aprendizaje con muchos ejemplos (many-shot in-context learning) se ha vuelto una estrategia viable y potente (Agarwal et al., [Many-Shot In-Context Learning](#)).

Se han explorado diversas estrategias para guiar el comportamiento del modelo. El *role-prompting* (Kong et al., [Better Zero-Shot Reasoning with Role-Play Prompting](#)), que asigna un rol específico al LLM (e.g., actuar como un experto en un tema), y el *style prompting* (Lu et al., [Bounding the Capabilities of Large Language Models in Open Text Generation with Prompt Constraints](#)), que especifica el estilo o tono deseado, permiten adaptar la respuesta a necesidades concretas. La capacidad de generalización de los LLM también se ha aprovechado para inducirlos a generar pasos intermedios de razonamiento, mejorando la calidad de las respuestas en

tareas complejas, un enfoque conocido como *cadena de pensamiento* (Chain-of-Thought) (Nye et al., *Show Your Work*; Wei et al., *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*). Este paradigma se ha extendido incluso a escenarios sin ejemplos explícitos (zero-shot CoT), donde se instruye al modelo a *pensar paso a paso* (Kojima et al., *Large Language Models Are Zero-Shot Reasoners*; L. Wang et al., *Plan-and-Solve Prompting*).

Otras técnicas se centran en la complejidad y la descomposición. El *complexity-based prompting* aprovecha la observación de que respuestas más largas o complejas a menudo correlacionan con una mayor probabilidad de acierto (Fu et al., *Complexity-Based Prompting for Multi-Step Reasoning*), mientras que la descomposición de problemas (*least-to-most prompting*) se basa en que el modelo resuelva sub-tareas más simples para luego integrar sus respuestas en la solución de un problema mayor (Zhou et al., *Least-to-Most Prompting Enables Complex Reasoning in Large Language Models*). La consistencia también se ha utilizado como un criterio de selección, generando múltiples respuestas y eligiendo la más frecuente (*self-consistency*) (X. Wang et al., *Self-Consistency Improves Chain of Thought Reasoning in Language Models*).

Para problemas que requieren exploración o planificación, se han propuesto estructuras como el *Tree of Thoughts* y el *Graph of Thoughts*, que permiten al LLM explorar diferentes caminos de razonamiento tras descomponer el problema inicial (Yao, Yu et al., *Tree of Thoughts*; Besta et al., «*Graph of Thoughts*»). El *Buffer of Thought* (Yang et al., *Buffer of Thoughts*) acumula cadenas de razonamiento previas y las recupera selectivamente para nuevas tareas, de forma similar al razonamiento analógico, que busca aprovechar experiencias pasadas (Yasunaga et al., *Large Language Models as Analogical Reasoners*). La técnica de *step-back prompting* (Zheng et al., *Take a Step Back*) instruye al modelo a resolver primero una versión más general y abstracta del problema, lo que puede simplificar la tarea y evitar desviaciones por detalles excesivos. También se ha investigado la inducción de un análisis previo del modo de proceder antes de generar la cadena de razonamiento (*meta-reasoning*) (P. Gao et al., *Meta Reasoning for Large Language Models*). Técnicas como *Self-Ask* incitan al LLM a generar y responder preguntas de seguimiento para clarificar la tarea original antes de abordarla (Press et al., *Measuring and Narrowing the Compositionality Gap in Language Models*), mientras que *Self-Refine* establece un marco iterativo donde el modelo critica y mejora sus propias respuestas (Madaan et al., *Self-Refine*). La reformulación del problema o la instrucción inicial también

ha demostrado ser efectiva (Mishra et al., «[Reframing Instructional Prompts to GPTk's Language](#)»; Deng et al., [Rephrase and Respond](#)). Finalmente, se exploran enfoques que aprovechan representaciones gráficas textuales (*Mind's Eye*) (Wu et al., «[Mind's Eye of LLMs: Visualization-of-Thought Elicits Spatial Reasoning in Large Language Models](#)») o procesos cognitivos como el *Sketch of Thought* (Aytes et al., [Sketch-of-Thought](#)) y el *Metacognitive Prompting* (Y. Wang et al., [Metacognitive Prompting Improves Understanding in Large Language Models](#)). El *Cumulative Reasoning* genera y evalúa pasos potenciales de forma iterativa hasta alcanzar una solución satisfactoria (Zhang et al., [Cumulative Reasoning with Large Language Models](#)).

1.2.2. Agentes Basados en LLM

El verdadero potencial de los Grandes Modelos de Lenguaje podría residir en su capacidad para actuar de forma autónoma y utilizar herramientas externas para superar sus limitaciones inherentes, como en cálculos matemáticos, razonamiento complejo o la verificación de hechos. A medida que los LLM han mejorado, investigadores y empresas han explorado cómo permitirles interactuar con sistemas externos. El sistema MRKL (Modular Reasoning, Knowledge, and Language) (Karpas et al., [MRKL Systems](#)) es una de las formulaciones más simples de un agente, utilizando un LLM como *router* para acceder a múltiples herramientas (e.g., obtener el clima o la fecha actual) y combinar la información para generar una respuesta final. Modelos como PAL (Program-aided Language Model) (L. Gao et al., [PAL](#)) traducen problemas directamente a código ejecutable, mientras que ToRA (Tool-Integrated Reasoning Agent) (Gou et al., [ToRA](#)) intercala pasos de código y razonamiento. El paradigma ReAct (Reasoning and Acting) (Yao, Zhao et al., [ReAct](#)) permite a los agentes generar un pensamiento, tomar una acción y recibir una observación, manteniendo un historial de estos pasos para informar decisiones futuras. Reflexion (Shinn et al., [Reflexion](#)) extiende ReAct incorporando retroalimentación lingüística para refinar el comportamiento del agente. La Generación Aumentada por Recuperación (RAG, por sus siglas en inglés) (Lewis et al., [Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks](#)), donde se recupera información de una fuente externa para insertarla en el prompt, se considera un sistema de agente cuando la propia recuperación se trata como una herramienta externa.

1.3. Integración de Herramientas Basadas en LLM en Aplicaciones de Toma de Notas

Los avances en los Grandes Modelos de Lenguaje han catalizado el desarrollo de numerosas herramientas que buscan automatizar y enriquecer la toma de notas, la sumarización, la creación de enlaces y la búsqueda dentro de los sistemas de PKM. Estas herramientas varían desde aplicaciones web independientes hasta plugins específicos para plataformas de gestión del conocimiento consolidadas.

Un ejemplo destacado de aplicación web es **InfraNodus**, que visualiza texto como redes de conocimiento, permitiendo identificar términos influyentes y lagunas conceptuales mediante el análisis de grafos generados a partir de texto y metadatos como *#hashtags* y *@mentions*. Aunque no es estrictamente una herramienta de toma de notas, su capacidad para generar nodos (conceptos) y conexiones (co-ocurrencias) a partir de texto ilustra el potencial de los modelos gráficos para la gestión del conocimiento.

En el ecosistema de **Logseq**, han surgido múltiples plugins. **Logseq Copilot** (por chhabrakadabra) y **Logseq Copilot Plugin** (por chloe-15) ofrecen interfaces de chat con IA, indexando las notas del usuario para proporcionar respuestas contextualizadas y, en algunos casos, permitiendo la recuperación de contenido de páginas enlazadas. El plugin **ollama-logseq** y **logseq-rag** se centran en la integración con LLMs locales a través de Ollama, facilitando la generación de resúmenes, tarjetas de estudio y la descomposición de tareas, utilizando el contexto de la página o bloque actual. **Logseq Composer** va un paso más allá al permitir la conexión de notas con cualquier LLM mediante Retrieval-Augmented Generation (RAG) a través de LiteLLM, enfocándose en la recuperación de notas relevantes para enriquecer las interacciones con la IA.

El entorno de **Obsidian.md** también ha visto una explosión de herramientas impulsadas por LLM. **Cannoli** permite construir y ejecutar scripts de LLM sin código directamente en el editor Canvas de Obsidian, posibilitando la creación de chatbots con lógica personalizada y la interacción con el contenido de la bóveda. **ExMemo Tools** y su predecesor, **ExMemo Assistant**, se enfocan en la gestión inteligente de documentos, ofreciendo archivado automático, inserción de contenido y generación de metadatos como etiquetas y descripciones. Para la mejora de la escritura y la organización, **Zettelkasten LLM Tools** proporciona búsqueda semántica y un copiloto de IA para reescribir y mejorar la estructura de las notas.

La generación de contenido específico también es una capacidad común, con plugins como **Obsidian Flashcards LLM** para crear tarjetas de estudio, **Canvas LLM Extender** para añadir nodos de texto generados por IA al Canvas, y **Daily Summary Plugin for Obsidian** para crear informes diarios a partir de las notas del día.

La integración con LLMs locales es una tendencia creciente, como se observa en **Local LLM Helper - Obsidian Plugin** y **Obsidian AI plugin**, que conectan con servidores como Ollama o LM Studio para procesar texto, generar backlinks o chatear con las notas, manteniendo la privacidad de los datos. La creación de material de estudio se ve potenciada por herramientas como **LLM Testing Plugin** y **Quiz Generator**, que generan preguntas y cuestionarios a partir de las notas utilizando una variedad de proveedores de LLM. Para una interacción más profunda y contextualizada, **LLM Workspace plugin for Obsidian** permite la creación de conjuntos de fuentes curadas manualmente para fundamentar las conversaciones con la IA mediante RAG. **LLM Summary** automatiza la creación de resúmenes de archivos PDF y la extracción de conceptos clave, generando nuevas notas y enlazándolas. La organización se facilita con **Obsidian LLM Tagger Plugin**, que utiliza LLMs locales para etiquetar notas automáticamente.

Otras herramientas como **Interact with LLMs in Obsidian** y **BMO Chatbot for Obsidian** proporcionan interfaces de chat versátiles, conectándose tanto a servicios en la nube (OpenAI, Anthropic, Google) como a LLMs locales (GPT4All, Ollama), permitiendo referenciar la nota actual en las conversaciones. **InsightA Obsidian Plugin** se especializa en transformar artículos largos en notas atómicas interconectadas y generar Mapas de Contenido (MOC). **ChatGPT MD** ofrece una integración profunda con ChatGPT, OpenRouter.ai y Ollama, permitiendo conversaciones interactivas que pueden tomar contexto de notas enlazadas. Herramientas más especializadas como **LaTeX Generator Plugin for Obsidian** convierten lenguaje natural a ecuaciones LaTeX usando Ollama, mientras que **Simple Prompt** facilita la generación de contenido y la reescritura de selecciones o documentos completos. **Obsidian Explain Selection with AI Plugin** permite obtener elaboraciones sobre texto seleccionado utilizando diversos LLMs. **Obsidian Cloud Atlas Plugin** introduce flujos de trabajo en Canvas o Markdown para procesar notas y obtener respuestas contextuales, con capacidades adicionales como reconocimiento de entidades para crear wikilinks. Finalmente, **Caret Obsidian Plugin** también ofrece un Canvas de IA para chat no lineal y la capacidad de referenciar

archivos de la bóveda.

Plataformas como Notion también han integrado capacidades de IA de forma nativa (Notion AI), permitiendo resumir notas, responder preguntas sobre el espacio de trabajo y generar contenido. Aunque Notion AI no crea enlaces automáticamente, herramientas de terceros pueden conectar con Notion para indexar sus páginas como una base de conocimiento.

Esta tendencia general hacia la integración de LLMs en herramientas de PKM subraya un esfuerzo por automatizar tareas, mejorar la interconexión de ideas y potenciar la utilidad de los sistemas de gestión del conocimiento personal. La diversidad de enfoques, desde el chat contextual hasta la generación estructurada de notas y metadatos, refleja la búsqueda de soluciones que se adapten a las necesidades específicas de los usuarios en la construcción y mantenimiento de sus bases de conocimiento.

1.4. Objetivos de la Tesis

El presente trabajo de tesis tiene como objetivo principal avanzar en la automatización de la construcción incremental y progresiva de bases de conocimiento personal. Este esfuerzo se alinea y busca continuar la línea de investigación explorada en trabajos como el de Fraga et al., «[On the Automatic Generation of Knowledge Connections](#)», que abordó la generación automática de conocimiento. Sin embargo, esta tesis se enfoca específicamente en el aprovechamiento de las capacidades de los Grandes Modelos de Lenguaje para generar y enriquecer información directamente en el contexto de una base de conocimiento personal existente o en desarrollo.

Particularmente, se investigará la aplicación de LLMs para la creación y expansión de notas semiestructuradas, utilizando lenguajes de marcado como Markdown, que son comunes en sistemas de PKM modernos. Estas notas, al estar interconectadas, forman un grafo de conocimiento dinámico. Dada la flexibilidad inherente a este formato de notas, se busca que la solución propuesta sea compatible con diversas metodologías y sistemas de toma de notas previamente discutidos, desde el Zettelkasten hasta enfoques más fluidos.

Hasta donde alcanza el conocimiento del autor, no existe actualmente una herramienta que combine de manera integral la potencia de los LLM con el objetivo específico de asistir en la construcción progresiva y contextualizada de una base

de conocimiento personal basada en notas interconectadas y semiestructuradas. Para abordar esta brecha, se ha desarrollado un prototipo de programa capaz de procesar archivos en diversas modalidades (e.g., texto, documentos PDF, páginas web) y, mediante la interacción con un LLM, extraer, sintetizar e integrar el conocimiento relevante en una base de notas existente (o crear una nueva desde cero), generando nuevas notas, resúmenes, conexiones y metadatos que enriquezcan el grafo de conocimiento del usuario de manera coherente y contextualmente apropiada. Este trabajo explorará la viabilidad, eficacia y los desafíos de dicha automatización.

Bibliografía

- Ahrens, Sönke. *How to Take Smart Notes: One Simple Technique to Boost Writing, Learning and Thinking – for Students, Academics and Nonfiction Book Writers*. 20 de feb. de 2017 (ver páginas 1, 3).
- Forte, Tiago. *Building a Second Brain*. Profile, 14 de jun. de 2022 (ver páginas 1, 3).
- Grundspenkis, Janis. «Agent Based Approach for Organization and Personal Knowledge Modelling: Knowledge Management Perspective». En: *Journal of Intelligent Manufacturing* 18.4 (30 de jul. de 2007), págs. 451-457. ISSN: 0956-5515, 1572-8145. DOI: [10.1007/s10845-007-0052-6](https://doi.org/10.1007/s10845-007-0052-6) (ver página 2).
- Jansen, Renée S., Daniel Lakens y Wijnand A. IJsselsteijn. «An Integrative Review of the Cognitive Costs and Benefits of Note-Taking». En: *Educational Research Review* 22 (nov. de 2017), págs. 223-233. ISSN: 1747938X. DOI: [10.1016/j.edurev.2017.10.001](https://doi.org/10.1016/j.edurev.2017.10.001) (ver página 2).
- Pauk, Walter y Ross J. Q. Owens. *How to Study in College*. 10th Ed. Boston, MA: Cengage Learning, 2010. 406 págs. ISBN: 978-1-4390-8446-5 978-0-495-90302-4 (ver página 2).
- Bush, Vannevar. *As We May Think*. 15 de ene. de 1945 (ver página 3).
- Davies, Stephen, Javier Velez-Morales y Roger King. «Building the Memex Sixty Years Later: Trends and Directions in Personal Knowledge Bases». En: () (ver página 3).
- Davies, Stephen. «Still Building the Memex». En: *Communications of the ACM* 54.2 (feb. de 2011), págs. 80-88. ISSN: 0001-0782, 1557-7317. DOI: [10.1145/1897816.1897840](https://doi.org/10.1145/1897816.1897840) (ver página 3).
- Pyne, Yvette y Stuart Stewart. «Meta-Work: How We Research Is as Important as What We Research». En: *The British Journal of General Practice* 72.716 (25 de feb. de 2022), págs. 130-131. ISSN: 0960-1643. DOI: [10.3399/bjgp22X718757](https://doi.org/10.3399/bjgp22X718757). pmid: [35210247](https://pubmed.ncbi.nlm.nih.gov/35210247/) (ver página 4).
- OpenAI et al. *GPT-4 Technical Report*. 4 de mar. de 2024. DOI: [10.48550/arXiv.2303.08774](https://doi.org/10.48550/arXiv.2303.08774). arXiv: [2303.08774 \[cs\]](https://arxiv.org/abs/2303.08774). URL: <http://arxiv.org/abs/2303.08774> (visitado 07-05-2025). Prepublicado (ver página 5).

- Team, Gemini et al. *Gemini: A Family of Highly Capable Multimodal Models*. 17 de jun. de 2024. DOI: [10.48550/arXiv.2312.11805](https://doi.org/10.48550/arXiv.2312.11805). arXiv: [2312.11805 \[cs\]](https://arxiv.org/abs/2312.11805). URL: <http://arxiv.org/abs/2312.11805> (visitado 07-05-2025). Prepublicado (ver página 5).
- DeepSeek-AI et al. *DeepSeek-V3 Technical Report*. Ver. 1. 27 de dic. de 2024. DOI: [10.48550/arXiv.2412.19437](https://doi.org/10.48550/arXiv.2412.19437). arXiv: [2412.19437 \[cs\]](https://arxiv.org/abs/2412.19437). URL: <http://arxiv.org/abs/2412.19437> (visitado 28-02-2025). Prepublicado (ver página 5).
- Bai, Jinze et al. *Qwen Technical Report*. 28 de sep. de 2023. DOI: [10.48550/arXiv.2309.16609](https://doi.org/10.48550/arXiv.2309.16609). arXiv: [2309.16609 \[cs\]](https://arxiv.org/abs/2309.16609). URL: <http://arxiv.org/abs/2309.16609> (visitado 07-05-2025). Prepublicado (ver página 5).
- Grattafiori, Aaron et al. *The Llama 3 Herd of Models*. 23 de nov. de 2024. DOI: [10.48550/arXiv.2407.21783](https://doi.org/10.48550/arXiv.2407.21783). arXiv: [2407.21783 \[cs\]](https://arxiv.org/abs/2407.21783). URL: <http://arxiv.org/abs/2407.21783> (visitado 07-05-2025). Prepublicado (ver página 5).
- Brown, Tom B. et al. «Language Models Are Few-Shot Learners». En: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 6 de dic. de 2020, págs. 1877-1901. ISBN: 978-1-7138-2954-6 (ver página 5).
- Min, Sewon et al. «Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?» En: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. EMNLP 2022. Ed. por Yoav Goldberg, Zornitsa Kozareva y Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, dic. de 2022, págs. 11048-11064. DOI: [10.18653/v1/2022.emnlp-main.759](https://doi.org/10.18653/v1/2022.emnlp-main.759) (ver página 5).
- Agarwal, Rishabh et al. *Many-Shot In-Context Learning*. 17 de oct. de 2024. DOI: [10.48550/arXiv.2404.11018](https://doi.org/10.48550/arXiv.2404.11018). arXiv: [2404.11018 \[cs\]](https://arxiv.org/abs/2404.11018). URL: <http://arxiv.org/abs/2404.11018> (visitado 28-02-2025). Prepublicado (ver página 5).
- Kong, Aobo et al. *Better Zero-Shot Reasoning with Role-Play Prompting*. 14 de mar. de 2024. DOI: [10.48550/arXiv.2308.07702](https://doi.org/10.48550/arXiv.2308.07702). arXiv: [2308.07702 \[cs\]](https://arxiv.org/abs/2308.07702). URL: <http://arxiv.org/abs/2308.07702> (visitado 06-05-2025). Prepublicado (ver página 5).
- Lu, Albert et al. *Bounding the Capabilities of Large Language Models in Open Text Generation with Prompt Constraints*. 17 de feb. de 2023. DOI: [10.48550/arXiv.2302.09185](https://doi.org/10.48550/arXiv.2302.09185). arXiv: [2302.09185 \[cs\]](https://arxiv.org/abs/2302.09185). URL: <http://arxiv.org/abs/2302.09185> (visitado 06-05-2025). Prepublicado (ver página 5).
- Nye, Maxwell et al. *Show Your Work: Scratchpads for Intermediate Computation with Language Models*. 30 de nov. de 2021. DOI: [10.48550/arXiv.2112.00114](https://doi.org/10.48550/arXiv.2112.00114). arXiv: [2112.00114 \[cs\]](https://arxiv.org/abs/2112.00114). URL: <http://arxiv.org/abs/2112.00114> (visitado 04-04-2025). Prepublicado (ver página 6).

- Wei, Jason et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 10 de ene. de 2023. DOI: [10.48550/arXiv.2201.11903](https://doi.org/10.48550/arXiv.2201.11903). arXiv: [2201.11903 \[cs\]](https://arxiv.org/abs/2201.11903). URL: <http://arxiv.org/abs/2201.11903> (visitado 01-05-2025). Prepublicado (ver página 6).
- Kojima, Takeshi et al. *Large Language Models Are Zero-Shot Reasoners*. 29 de ene. de 2023. DOI: [10.48550/arXiv.2205.11916](https://doi.org/10.48550/arXiv.2205.11916). arXiv: [2205.11916 \[cs\]](https://arxiv.org/abs/2205.11916). URL: <http://arxiv.org/abs/2205.11916> (visitado 02-05-2025). Prepublicado (ver página 6).
- Wang, Lei et al. *Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models*. 26 de mayo de 2023. DOI: [10.48550/arXiv.2305.04091](https://doi.org/10.48550/arXiv.2305.04091). arXiv: [2305.04091 \[cs\]](https://arxiv.org/abs/2305.04091). URL: <http://arxiv.org/abs/2305.04091> (visitado 19-04-2025). Prepublicado (ver página 6).
- Fu, Yao et al. *Complexity-Based Prompting for Multi-Step Reasoning*. 30 de ene. de 2023. DOI: [10.48550/arXiv.2210.00720](https://doi.org/10.48550/arXiv.2210.00720). arXiv: [2210.00720 \[cs\]](https://arxiv.org/abs/2210.00720). URL: <http://arxiv.org/abs/2210.00720> (visitado 01-05-2025). Prepublicado (ver página 6).
- Zhou, Denny et al. *Least-to-Most Prompting Enables Complex Reasoning in Large Language Models*. 16 de abr. de 2023. DOI: [10.48550/arXiv.2205.10625](https://doi.org/10.48550/arXiv.2205.10625). arXiv: [2205.10625 \[cs\]](https://arxiv.org/abs/2205.10625). URL: <http://arxiv.org/abs/2205.10625> (visitado 28-02-2025). Prepublicado (ver página 6).
- Wang, Xuezhi et al. *Self-Consistency Improves Chain of Thought Reasoning in Language Models*. 7 de mar. de 2023. DOI: [10.48550/arXiv.2203.11171](https://doi.org/10.48550/arXiv.2203.11171). arXiv: [2203.11171 \[cs\]](https://arxiv.org/abs/2203.11171). URL: <http://arxiv.org/abs/2203.11171> (visitado 01-05-2025). Prepublicado (ver página 6).
- Yao, Shunyu, Dian Yu et al. *Tree of Thoughts: Deliberate Problem Solving with Large Language Models*. 3 de dic. de 2023. DOI: [10.48550/arXiv.2305.10601](https://doi.org/10.48550/arXiv.2305.10601). arXiv: [2305.10601 \[cs\]](https://arxiv.org/abs/2305.10601). URL: <http://arxiv.org/abs/2305.10601> (visitado 28-02-2025). Prepublicado (ver página 6).
- Besta, Maciej et al. «Graph of Thoughts: Solving Elaborate Problems with Large Language Models». En: *Proceedings of the AAAI Conference on Artificial Intelligence* 38.16 (24 de mar. de 2024), págs. 17682-17690. ISSN: 2374-3468, 2159-5399. DOI: [10.1609/aaai.v38i16.29720](https://doi.org/10.1609/aaai.v38i16.29720). arXiv: [2308.09687 \[cs\]](https://arxiv.org/abs/2308.09687) (ver página 6).
- Yang, Ling et al. *Buffer of Thoughts: Thought-Augmented Reasoning with Large Language Models*. 14 de oct. de 2024. DOI: [10.48550/arXiv.2406.04271](https://doi.org/10.48550/arXiv.2406.04271). arXiv: [2406.04271 \[cs\]](https://arxiv.org/abs/2406.04271). URL: <http://arxiv.org/abs/2406.04271> (visitado 28-02-2025). Prepublicado (ver página 6).
- Yasunaga, Michihiro et al. *Large Language Models as Analogical Reasoners*. 9 de mar. de 2024. DOI: [10.48550/arXiv.2310.01714](https://doi.org/10.48550/arXiv.2310.01714). arXiv: [2310.01714 \[cs\]](https://arxiv.org/abs/2310.01714). URL: <http://arxiv.org/abs/2310.01714> (visitado 28-02-2025). Prepublicado (ver página 6).

- Zheng, Huaixiu Steven et al. *Take a Step Back: Evoking Reasoning via Abstraction in Large Language Models*. 12 de mar. de 2024. DOI: [10.48550/arXiv.2310.06117](https://doi.org/10.48550/arXiv.2310.06117). arXiv: [2310.06117 \[cs\]](https://arxiv.org/abs/2310.06117). URL: <http://arxiv.org/abs/2310.06117> (visitado 02-05-2025). Prepublicado (ver página 6).
- Gao, Peizhong et al. *Meta Reasoning for Large Language Models*. 17 de jun. de 2024. DOI: [10.48550/arXiv.2406.11698](https://doi.org/10.48550/arXiv.2406.11698). arXiv: [2406.11698 \[cs\]](https://arxiv.org/abs/2406.11698). URL: <http://arxiv.org/abs/2406.11698> (visitado 28-02-2025). Prepublicado (ver página 6).
- Press, Ofir et al. *Measuring and Narrowing the Compositionality Gap in Language Models*. 17 de oct. de 2023. DOI: [10.48550/arXiv.2210.03350](https://doi.org/10.48550/arXiv.2210.03350). arXiv: [2210.03350 \[cs\]](https://arxiv.org/abs/2210.03350). URL: <http://arxiv.org/abs/2210.03350> (visitado 06-05-2025). Prepublicado (ver página 6).
- Madaan, Aman et al. *Self-Refine: Iterative Refinement with Self-Feedback*. 25 de mayo de 2023. DOI: [10.48550/arXiv.2303.17651](https://doi.org/10.48550/arXiv.2303.17651). arXiv: [2303.17651 \[cs\]](https://arxiv.org/abs/2303.17651). URL: <http://arxiv.org/abs/2303.17651> (visitado 01-05-2025). Prepublicado (ver página 6).
- Mishra, Swaroop et al. «Reframing Instructional Prompts to GPTk's Language». En: *Findings of the Association for Computational Linguistics: ACL 2022*. Findings 2022. Ed. por Smaranda Muresan, Preslav Nakov y Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, mayo de 2022, págs. 589-612. DOI: [10.18653/v1/2022.findings-acl.50](https://doi.org/10.18653/v1/2022.findings-acl.50) (ver página 7).
- Deng, Yihe et al. *Rephrase and Respond: Let Large Language Models Ask Better Questions for Themselves*. 18 de abr. de 2024. DOI: [10.48550/arXiv.2311.04205](https://doi.org/10.48550/arXiv.2311.04205). arXiv: [2311.04205 \[cs\]](https://arxiv.org/abs/2311.04205). URL: <http://arxiv.org/abs/2311.04205> (visitado 02-05-2025). Prepublicado (ver página 7).
- Wu, Wenshan et al. «Mind's Eye of LLMs: Visualization-of-Thought Elicits Spatial Reasoning in Large Language Models». En: () (ver página 7).
- Aytes, Simon A., Jinheon Baek y Sung Ju Hwang. *Sketch-of-Thought: Efficient LLM Reasoning with Adaptive Cognitive-Inspired Sketching*. 7 de mar. de 2025. DOI: [10.48550/arXiv.2503.05179](https://doi.org/10.48550/arXiv.2503.05179). arXiv: [2503.05179 \[cs\]](https://arxiv.org/abs/2503.05179). URL: <http://arxiv.org/abs/2503.05179> (visitado 22-03-2025). Prepublicado (ver página 7).
- Wang, Yuqing y Yun Zhao. *Metacognitive Prompting Improves Understanding in Large Language Models*. 20 de mar. de 2024. DOI: [10.48550/arXiv.2308.05342](https://doi.org/10.48550/arXiv.2308.05342). arXiv: [2308.05342 \[cs\]](https://arxiv.org/abs/2308.05342). URL: <http://arxiv.org/abs/2308.05342> (visitado 06-05-2025). Prepublicado (ver página 7).
- Zhang, Yifan et al. *Cumulative Reasoning with Large Language Models*. 12 de mar. de 2025. DOI: [10.48550/arXiv.2308.04371](https://doi.org/10.48550/arXiv.2308.04371). arXiv: [2308.04371 \[cs\]](https://arxiv.org/abs/2308.04371). URL: <http://arxiv.org/abs/2308.04371> (visitado 06-05-2025). Prepublicado (ver página 7).

- Karpas, Ehud et al. *MRKL Systems: A Modular, Neuro-Symbolic Architecture That Combines Large Language Models, External Knowledge Sources and Discrete Reasoning*. 1 de mayo de 2022. DOI: [10.48550/arXiv.2205.00445](https://doi.org/10.48550/arXiv.2205.00445). arXiv: [2205.00445 \[cs\]](https://arxiv.org/abs/2205.00445). URL: <http://arxiv.org/abs/2205.00445> (visitado 01-05-2025). Prepublicado (ver página 7).
- Gao, Luyu et al. *PAL: Program-aided Language Models*. 27 de ene. de 2023. DOI: [10.48550/arXiv.2211.10435](https://doi.org/10.48550/arXiv.2211.10435). arXiv: [2211.10435 \[cs\]](https://arxiv.org/abs/2211.10435). URL: <http://arxiv.org/abs/2211.10435> (visitado 07-05-2025). Prepublicado (ver página 7).
- Gou, Zhibin et al. *ToRA: A Tool-Integrated Reasoning Agent for Mathematical Problem Solving*. 21 de feb. de 2024. DOI: [10.48550/arXiv.2309.17452](https://doi.org/10.48550/arXiv.2309.17452). arXiv: [2309.17452 \[cs\]](https://arxiv.org/abs/2309.17452). URL: <http://arxiv.org/abs/2309.17452> (visitado 07-05-2025). Prepublicado (ver página 7).
- Yao, Shunyu, Jeffrey Zhao et al. *ReAct: Synergizing Reasoning and Acting in Language Models*. 10 de mar. de 2023. DOI: [10.48550/arXiv.2210.03629](https://doi.org/10.48550/arXiv.2210.03629). arXiv: [2210.03629 \[cs\]](https://arxiv.org/abs/2210.03629). URL: <http://arxiv.org/abs/2210.03629> (visitado 25-04-2025). Prepublicado (ver página 7).
- Shinn, Noah et al. *Reflexion: Language Agents with Verbal Reinforcement Learning*. arXiv.org. 20 de mar. de 2023. URL: <https://arxiv.org/abs/2303.11366v4> (visitado 07-05-2025) (ver página 7).
- Lewis, Patrick et al. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. 12 de abr. de 2021. DOI: [10.48550/arXiv.2005.11401](https://doi.org/10.48550/arXiv.2005.11401). arXiv: [2005.11401 \[cs\]](https://arxiv.org/abs/2005.11401). URL: <http://arxiv.org/abs/2005.11401> (visitado 19-04-2025). Prepublicado (ver página 7).
- Fraga, Felipe et al. «On the Automatic Generation of Knowledge Connections:» en: *Proceedings of the 25th International Conference on Enterprise Information Systems*. 25th International Conference on Enterprise Information Systems. Prague, Czech Republic: SCITEPRESS - Science and Technology Publications, 2023, págs. 43-54. ISBN: 978-989-758-648-4. DOI: [10.5220/0011781100003467](https://doi.org/10.5220/0011781100003467) (ver página 10).