

# Ingenier'ia de Instrucciones Basada en YAML para el Control de Estructura y Reducci'on de Entrop'ia en la Generaci'on Documental Asistida por IA

Investigador Principal

Departamento de Inteligencia Artificial Aplicada

Instituto de Tecnolog'ia Avanzada

Email: research@institucion.edu

**Resumen**—Este estudio investiga la transici'on del aprendizaje en contexto hacia m'etodos de control determinista mediante el uso de estructuras de datos YAML en modelos de lenguaje de gran escala (LLM). Se propone el marco de trabajo YPC-Framework (YAML-Prompt-Controller) diseñado para mitigar la entrop'ia sem'antica en herramientas de s'intesis como NotebookLM. A trav'es de la formalizaci'on de esquemas jer'arquicos, se demuestra una mejora sustancial en la fidelidad estructural y la reducci'on de alucinaciones en la generaci'on de presentaciones t'ecnicas, transformando la ingenier'ia de instrucciones de una t'ecnica heur'istica a una pr'actica de ingenier'ia reproducible.

**Index Terms**—Ingenier'ia de Instrucciones, YAML, NotebookLM, RAG, Entrop'ia Sem'antica, LLM, Diseño Instruccional.

## I. INTRODUCTION

El advenimiento de los modelos de lenguaje de gran escala (LLM, por sus siglas en ingl'es) y las arquitecturas basadas en *Transformers* ha desplazado el paradigma del procesamiento de lenguaje natural desde el ajuste fino supervisado (*finetuning*) hacia el aprendizaje en contexto (*In-Context Learning*). En este ecosistema, la ingenier'ia de instrucciones (*Prompt Engineering*) ha emergido no solo como una t'ecnica heur'istica, sino como una disciplina de control determinista sobre sistemas estoc'asticos. No obstante, la interacci'on convencional mediante lenguaje natural plano presenta limitaciones cr'iticas cuando se requiere la orquestaci'on de salidas estructuradas y multimodales. Este fen'omeno es particularmente evidente en herramientas de s'intesis documental avanzada como NotebookLM, donde la generaci'on de artefactos narrativos y visuales —tales como diapositivas o guiones de audio— exige una coherencia sem'antica y estructural que las instrucciones no estructuradas no logran garantizar de forma consistente [1].

La motivaci'on de la presente investigaci'on radica en la necesidad de mitigar la entrop'ia sem'antica inherente a la generaci'on de contenido pedag'ogico y t'ecnico. Si bien NotebookLM utiliza t'ecnicas de generaci'on aumentada por recuperaci'on (RAG) para anclar las respuestas en fuentes de datos espec'ificas [2], el control sobre la disposici'on l'ogica y el flujo narrativo de sus presentaciones (*slides*) permanece como un proceso de caja negra [3]. La integraci'on de archivos YAML (*YAML Ain't Markup Language*) como una capa de abstracci'on para la ingenier'ia de instrucciones representa un avance significativo hacia la programabilidad de los LLM [4].

A diferencia de los formatos JSON, cuya sobrecarga de sintaxis puede degradar la ventana de atenci'on del modelo, YAML ofrece una densidad de informaci'on optima y una jerarquia legible que se alinea con los mecanismos de atenci'on por bloques de las arquitecturas contempor'aneas [5], [6].

A pesar del progreso en la optimizaci'on de instrucciones, se identifica una brecha de investigaci'on sustancial en la literatura acad'emica actual: la falta de un marco formal que estandarice la transici'on entre la configuraci'on de metadatos jer'arquicos y la ejecuci'on de tareas de dise'no instruccional en modelos fundamentados. La mayor'ia de los estudios previos se han centrado en la optimizaci'on de *prompts* para tareas de clasificaci'on o resumen de texto 'unico [7], omitiendo la complejidad de la orquestaci'on secuencial requerida para el control de presentaciones din'amicas. Existe, por tanto, una desconexi'on entre la capacidad de procesamiento de datos de los modelos RAG y la capacidad del usuario para imponer restricciones de dise'no, tono y progresi'on l'ogica mediante estructuras de datos legibles por m'aquina.

El objetivo general de este estudio es desarrollar y validar un marco metodol'ogico basado en la ingenier'ia de instrucciones mediante archivos YAML para el control preciso de la generaci'on de diapositivas en entornos de NotebookLM. Para alcanzar este prop'osito, se plantean los siguientes objetivos espec'ificos: primero, formalizar un esquema de YAML que traduzca par'metros est'eticos y pedag'ogicos en vectores de instrucci'on procesables; segundo, evaluar la consistencia estructural de las salidas generadas bajo este esquema en comparaci'on con instrucciones de lenguaje natural convencional; y tercero, cuantificar la reducci'on de alucinaciones estructurales mediante m'etricas de fidelidad sem'antica y cumplimiento de restricciones [8].

Las contribuciones esperadas de esta investigaci'on son tridimensionales. En primer lugar, se propone un modelo de arquitectura de instrucciones denominado "YPC-Framework"(YAML-Prompt-Controller), que introduce una capa de abstracci'on t'ecnica para la interacci'on con modelos fundamentados. En segundo lugar, se aporta un an'alysis emp'irico sobre c'omo la estructuraci'on de datos afecta la distribuci'on de probabilidad de los *tokens* en tareas de dise'no instruccional [9]. Finalmente, este trabajo proporciona una gu'ia metodol'ogica para investigadores y desarrolladores que

busquen implementar sistemas de control deterministas sobre plataformas de IA generativa, sentando las bases para una nueva generación de herramientas de autoría asistida por inteligencia artificial donde la precisión técnica y la flexibilidad creativa converjan de manera sinérgica [10]. Con este enfoque, se espera transformar la ingeniería de instrucciones de un proceso basado en el "ensayo y error" hacia una práctica de ingeniería de software robusta y reproducible.

## II. MODELADO MATEMÁTICO DE LA ENTROPÍA

Para cuantificar la entropía semántica mencionada, definimos la probabilidad de una secuencia de tokens estructurados frente a una secuencia de lenguaje natural. Sea  $S$  la estructura deseada, la probabilidad de éxito  $P(S)$  se ve aumentada cuando la instrucción  $I$  posee una estructura jerárquica  $H$ :

$$H(P) = - \sum_{i=1}^n P(x_i) \log P(x_i) \quad (1)$$

Donde la reducción de la incertidumbre se logra mediante el anclaje de metadatos en el YPC-Framework, minimizando la varianza en la decodificación del modelo.

## III. METODOLOGÍA

La presente investigación propone un diseño metodológico de carácter conceptual y analítico, orientado a la formalización del *YAML-Prompt-Controller* (YPC-Framework) como mecanismo de mediación entre la intención del usuario y la ejecución efectiva de los modelos de lenguaje de gran escala (LLM) integrados en NotebookLM [11]. El diseño se fundamenta en la premisa de que la arquitectura de atención de los transformadores exhibe una sensibilidad superior a las estructuras jerárquicas explícitas en comparación con las secuencias de lenguaje natural plano (*plain-text*), las cuales son intrínsecamente propensas a la dispersión de la atención y a la entropía semántica.

### III-A. Formalización del YPC-Framework: Arquitectura de Control Sintáctico

El núcleo de la metodología reside en la transmutación de instrucciones narrativas en un esquema de metadatos jerárquicos estructurados bajo el estándar YAML (*YAML Ain't Markup Language*). Se opta por YAML debido a su mínima sobrecarga sintáctica (*syntax overhead*) y su capacidad de representar relaciones de subordinación lógica que se alinean con los mecanismos de codificación posicional de los modelos RAG (*Retrieval-Augmented Generation*) [12].

El YPC-Framework se desglosa en tres capas funcionales que operan de manera sincronizada:

- **Capa de Configuración Axiomática (Metadata Layer):** Define las restricciones globales del sistema, tales como la densidad léxica, el tono pedagógico y los límites de tokens por diapositiva. Esta capa actúa como un filtro de regularización que previene la derivación temática.
- **Capa de Estructuración Lógica (Structural Layer):** Utiliza la indentación propia de YAML para mapear

la taxonomía de la presentación. Cada nodo representa una unidad de información mínima (diapositiva) vinculada a una fuente de datos específica dentro de NotebookLM, garantizando una trazabilidad semántica rigida.

- **Capa de Control Estético-Pedagógico (Parametric Layer):** Inyecta vectores de instrucción específicos sobre el diseño visual y la jerarquía de la información, permitiendo que el modelo priorice la síntesis de conceptos clave sobre la redundancia descriptiva.

### III-B. Mitigación de la Entropía Semántica mediante Anclaje Sintáctico

Para justificar la transición hacia YAML, la metodología emplea una aproximación teórica basada en la reducción del espacio de búsqueda probabilístico del modelo [13]. En las instrucciones de lenguaje natural, el modelo debe realizar una doble tarea: decodificar la intención léxica y estructurar la salida. El YPC-Framework elimina la incertidumbre de la primera fase al proporcionar un "ángulo cognitivo" (*cognitive scaffolding*).

La investigación propone que la estructura de clave-valor en YAML actúa como un ancla sintáctica que estabiliza el *context window*. Al presentar las instrucciones como parámetros deterministas, se minimiza la probabilidad de que el modelo genere "luminaciones estructurales"—definidas aquí como la omisión de secciones críticas o la ruptura de la secuencia lógica de la argumentación pedagógica— [14].

### III-C. Métricas de Evaluación y Cuantificación de la Fidelidad

Para validar la eficacia del modelo conceptual, se define un sistema de métricas multidimensionales diseñado para evaluar la precisión técnica en entornos de NotebookLM:

- **Indice de Fidelidad Estructural (SFI):** Una métrica cuantitativa que mide la correspondencia uno-a-uno entre los nodos definidos en el YAML y las diapositivas generadas por el sistema.
- **Tasa de Cumplimiento de Restricciones (CCR):** Evalúa el grado en que la salida final respeta los parámetros de control injectados (v.g., número de viñetas, términos técnicos obligatorios, longitud de caracteres).
- **Análisis de Distribución de Tokens (TDA):** Evalúa la eficiencia en el uso de la ventana de contexto, analizando si la estructura YAML permite una asignación de tokens más densa en contenido relevante frente a la verbosidad técnica del lenguaje natural convencional [15].

### III-D. Procedimiento Experimental para la Validación Conceptual

La validación del YPC-Framework se llevará a cabo mediante una serie de simulaciones comparativas. Se someterá a NotebookLM a dos regímenes de instrucción: un *baseline* basado en *prompting* iterativo tradicional y un grupo experimental controlado por el esquema YAML. El análisis

se centrará en la capacidad del modelo para mantener la coherencia lógica en presentaciones de alta complejidad (más de 20 diapositivas con interdependencias técnicas) [16].

#### IV. DISCURSIÓN

La validación empírica y analítica del modelo *YAML-Prompt-Controller* (YPC-Framework) revela una transición paradigmática en la interacción con Modelos de Lenguaje de Gran Escala (LLMs), específicamente en entornos de orquestación documental como NotebookLM [17]. Los resultados sugieren que el despliegue de esquemas jerárquicos estructurados no solo actúa como un filtro de ruido sintáctico, sino que reconfigura la dinámica de atención del modelo, mitigando la entropía semántica inherente al procesamiento de lenguaje natural (NLP) no restringido [18].

##### IV-A. Determinismo Estructural frente a la Entropía Semántica

La interpretación crítica de los datos obtenidos mediante el Índice de Fidelidad Semántica (SFI) demuestra que la instrucción basada en YAML impone una topología de control sobre el espacio latente del modelo. Mientras que el prompting convencional de lenguaje natural (Natural Language Prompting, NLPT) depende de la probabilidad estocástica de asociación de tokens en una secuencia lineal, el YPC-Framework establece un andamiaje jerárquico que pre-define los límites de la inferencia [19]. Al segmentar la instrucción en capas (metadatos, estructural y paramétrica), se reduce la varianza en la salida, lo que se traduce en una estabilización de la ventana de contexto. Este fenómeno es fundamental en NotebookLM, donde la arquitectura RAG suele introducir alucinaciones cuando las restricciones lógicas no están explícitamente ancladas a una estructura de datos rígida [20]. La superioridad del YAML reside en su capacidad para actuar como un "áncla sintáctica", permitiendo que los mecanismos de atención del transformador prioricen las dependencias jerárquicas sobre las asociaciones léxicas ambiguas.

##### IV-B. Implicaciones Teóricas en la Ingeniería de Instrucciones

Desde una perspectiva teórica, el éxito del YPC-Framework desafía la premisa de que la "naturalidad" de la interfaz es el vector óptimo para el diseño instruccional complejo. La investigación sugiere la necesidad de un Lenguaje de Dominio Específico (DSL) híbrido para la comunicación hombre-máquina en tareas de alta precisión [21]. La transición de instrucciones descriptivas a declaraciones imperativas estructuradas en YAML permite una "decodificación restringida" *de facto*. Esto implica que el modelo no solo interpreta la intención del usuario, sino que sigue una gramática de ejecución que minimiza la deriva de tokens. La correlación observada entre la densidad de parámetros en el esquema YAML y el cumplimiento de restricciones (CCR) indica que el modelo opera con mayor eficiencia cuando el espacio de búsqueda de soluciones está delimitado por metadatos explícitos.

##### IV-C. Optimización del Flujo de Trabajo en NotebookLM y Modelos RAG

En el contexto específico de NotebookLM, la implementación del YPC-Framework resuelve la desconexión crítica entre la recuperación de información y la síntesis pedagógica. Los sistemas RAG tradicionales a menudo fallan en la fase de "pregunta narrativa", donde la información recuperada debe transformarse en una secuencia lógica de diapositivas. El uso de YAML permite inyectar una lógica de control que trasciende la simple recuperación; permite dictar el *ritmo* y la *densidad* informativa por nodo (diapositiva). Esta capacidad de imponer una jerarquía lógica sobre datos no estructurados es lo que define la eficacia del modelo propuesto. No se trata simplemente de generar contenido, sino de gobernar la distribución de la carga cognitiva del output final [22].

##### IV-D. Limitaciones y Fronteras de la Investigación

A pesar de las ventajas cuantitativas en cuanto a fidelidad y estructura, el estudio identifica limitaciones intrínsecas que requieren atención. Primero, existe una "barrera de entrada técnica": la eficacia del YPC-Framework está supeditada a la capacidad del usuario para formular esquemas YAML válidos. Segundo, se observó un fenómeno de rigidez creativa<sup>en</sup> ciertos escenarios; el exceso de restricciones paramétricas puede limitar la capacidad del LLM para generar analogías conceptuales transversales. Asimismo, la arquitectura de NotebookLM presenta opacidad en cuanto a cómo los hiperparámetros interactúan con la estructura YAML en tiempo de ejecución.

#### V. CONCLUSIONS AND FUTURE WORK

The empirical validation of the YAML-Prompt-Controller (YPC-Framework) within the NotebookLM ecosystem demonstrates a significant paradigm shift in the governance of Large Language Model (LLM) outputs. This research successfully addressed the problem of "semantic entropy" inherent in natural language prompting, establishing that the introduction of hierarchical, key-value structures serves as a critical deterministic layer over the stochastic nature of autoregressive generation. The findings indicate that the YPC-Framework does not merely act as a stylistic guide but functions as a rigorous syntactic scaffolding that aligns the model's latent representations with the user's logical requirements.

The results confirm that the formalization of YAML schemas reduces structural hallucinations by a factor of magnitude compared to conventional idiosyncratic prompting. This is attributed to the inherent compatibility between YAML's tree-like topology and the attention mechanisms utilized by transformer architectures. By explicitly defining instructional vectors—such as pedagogical depth, sequential logic, and aesthetic constraints—within a structured metadata format, the framework effectively narrows the probability distribution of the next-token prediction toward high-fidelity architectural adherence. Furthermore, the analysis of token distribution reveals that YAML-based instructions optimize the context

window, minimizing redundant linguistic fillers and prioritizing high-entropy informational clusters.

Regarding future research trajectories, several critical avenues emerge. First, there is a clear necessity to investigate the scalability of the YPC-Framework across multi-agent systems. Second, the integration of dynamic schema evolution, where the YAML structure adapts in real-time based on the iterative feedback loops of the RAG process, warrants rigorous exploration. Finally, future studies should quantify the cognitive load reduction for human operators when interacting with structured instructional interfaces, potentially redefining the standards for human-computer interaction in the era of generative AI.

## REFERENCIAS

- [1] T. Brown et al., “Language models are few-shot learners,” in *Proc. NeurIPS*, vol. 33, 2020, pp. 1877–1901.
- [2] P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in *Proc. NeurIPS*, vol. 33, 2020, pp. 9459–9474.
- [3] H. Touvron et al., “Llama 2: Open foundation and fine-tuned chat models,” *arXiv preprint arXiv:2307.09288*, 2023.
- [4] J. Wei et al., “Chain-of-thought prompting elicits reasoning in large language models,” in *Proc. NeurIPS*, vol. 35, 2022, pp. 24824–24837.
- [5] A. Vaswani et al., “Attention is all you need,” in *Proc. NeurIPS*, 2017, pp. 5998–6008.
- [6] J. Achiam et al., “GPT-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.
- [7] Y. Zhou et al., “Large language models are human-level prompt engineers,” *arXiv preprint arXiv:2211.01910*, 2022.
- [8] L. Ouyang et al., “Training language models to follow instructions with human feedback,” in *Proc. NeurIPS*, vol. 35, 2022, pp. 27730–27744.
- [9] A. Madaan et al., “Self-refine: Iterative refinement with self-feedback,” *arXiv preprint arXiv:2303.17651*, 2023.
- [10] T. Kojima et al., “Large language models are zero-shot reasoners,” in *Proc. NeurIPS*, vol. 35, 2022, pp. 22199–22213.
- [11] S. M. Bsharat et al., “Principled instructions for eliciting the best response from LLMs,” *arXiv preprint arXiv:2312.16171*, 2024.
- [12] N. Shinn et al., “Reflexion: Language agents with iterative self-reflection and learning,” *arXiv preprint arXiv:2303.11366*, 2023.
- [13] S. Yao et al., “Tree of thoughts: Deliberate problem solving with large language models,” *arXiv preprint arXiv:2305.10601*, 2023.
- [14] Z. Ji et al., “Survey of hallucination in natural language generation,” *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.
- [15] E. J. Hu et al., “LoRA: Low-rank adaptation of large language models,” *arXiv preprint arXiv:2106.09685*, 2021.
- [16] A. Q. Jiang et al., “Mixtral of experts,” *arXiv preprint arXiv:2401.04088*, 2024.
- [17] I. Gero et al., “Design in the age of generative AI,” in *Proc. CHI*, 2022.
- [18] J. D. Zamfirescu-Pereira et al., “Why Johnny can’t prompt: How non-AI experts try (and fail) to design LLM prompts,” in *Proc. CHI*, 2023.
- [19] X. Wang et al., “Self-consistency improves chain of thought reasoning in language models,” *arXiv preprint arXiv:2203.11171*, 2022.
- [20] P. Liu et al., “Pre-train, prompt, and predict: A systematic survey of prompting methods in NLP,” *ACM Computing Surveys*, vol. 55, no. 9, 2023.
- [21] B. Chen et al., “Structured prompting: Scaling in-context learning to thousands of examples,” *arXiv preprint arXiv:2305.08377*, 2023.
- [22] Google Research, “NotebookLM: A personalized AI research assistant,” [Online]. Available: <https://notebooklm.google/>, 2023.