BUILDERSOE: LLM EMPLEADO EN LA GENERACIÓN DE ARTÍCULOS CIENTÍFICOS BASADOS EN LATEX

1er M.Sc. Juan Carlos Peinado Pereira
Posgrado SOE – UAGRM
https://orcid.org/0009-0009-9117-2441
Santa Cruz, Bolivia | jcpeinado@soe.uagrm.edu.bo

2do M.Sc. Víctor Hugo Acosta Ortega
Posgrado SOE – UAGRM
https://orcid.org/0009-0005-8268-9212
Santa Cruz, Bolivia | hugoacosta@uagrm.edu.bo

https://doi.org/10.23670 FT.2026.1.22

Recibido 17/03/2026 - Aceptado 21/04/2026

RESUMEN

La complejidad sintáctica de LaTeX y su pronunciada curva de aprendizaje representan una barrera concreta para la producción científica en programas de posgrado. Para reducir este obstáculo, el presente estudio desarrolló un sistema que toma texto plano como entrada y genera, de forma automática, un artículo científico formateado según las normas de la revista destino –listo para su envío a publicación–, sin depender de servicios externos ni APIs de pago basada en modelos LLM locales. El procesamiento se realiza íntegramente mediante modelos de lenguaje ejecutados en local, lo que garantiza privacidad, autonomía y costo cero de operación.
La investigación se desarrolló bajo un enfoque cuantitativo, de tipo aplicado y diseño no experimental, con alcance descriptivo–evaluativo. El sistema fue evaluado mediante pruebas de compilación, análisis tipográfico por expertos, métricas de calidad matemática y de referencias, así como evaluación de percepción en usuarios.
Los resultados evidenciaron una tasa de compilación exitosa superior al umbral de viabilidad establecido. Asimismo, se obtuvo una calidad tipográfica significativamente mayor en comparación con Microsoft Word, junto con mejoras en la representación matemática y consistencia de referencias. La percepción de los usuarios mostró un índice elevado de facilidad de uso al igual que el nivel de satisfacción. Se conc luye que el sistema permite automatizar la generación de artículos científicos en LaTeX, reduciendo la barrera técnica y el tiempo de formateo, y constituyéndose en una alternativa viable para instituciones con recursos limitados. Como limitación, se identificó la gestión de claves bibliográficas, la cual requiere optimización en desarrollos futuros.

Palabras clave: modelos de lenguaje, LaTeX, escritura científica, inteligencia artificial, generación automática.

ABSTRACT

The barrier to LaTeX access limits quality scientific output in Latin American postgraduate programs. This paper presents the design and implementation of an assisted writing tool that integrates local large language models (LLM) through Ollama, a Streamlit interface, and a pdflatex compilation pipeline, enabling researchers to generate LaTeX-formatted scientific articles without prior knowledge of LaTeX syntax. Development was grounded in a documentary review of LLM-based academic writing tools, and in functional requirements derived from analysis of the researcher's actual workflow.
The resulting tool operates on a four-layer modular architecture—capture, LLM processing, assembly, and compilation—and was validated through successful compilation tests (94.3%), expert typographic evaluation (T_score 4.67/5 vs. Word's 3.20, $p < 0.001$), and a survey of 100 researchers (IFU = 4.27/5; 85.6% satisfaction). The system eliminates API costs by running entirely locally, making it viable for resource-constrained institutions.

Keywords: language models, LaTeX, scientific writing tool, Ollama, automation, open source.

INTRODUCCIÓN

Publicar en formatos académicos de calidad exige dominar simultáneamente el contenido de la investigación y las convenciones tipográficas de las revistas científicas. LaTeX se ha consolidado como el estándar en disciplinas como matemáticas, física e informática (Knuth, 1997), gracias a su precisión tipográfica, su manejo de expresiones matemáticas y la gestión automatizada de referencias (Kopka & Daly, 2004). Sin embargo, su curva de aprendizaje representa una barrera concreta para investigadores formados en entornos WYSIWYG como Microsoft Word: configurar un preámbulo, manejar entornos de ecuaciones o depurar errores de compilación requiere tiempo y práctica que muchos investigadores —especialmente en programas de posgrado latinoamericanos— no pueden dedicar.

Esta brecha motivó el desarrollo de un sistema que abstraigan la complejidad sintáctica de LaTeX sin sacrificar la calidad tipográfica. En paralelo, los modelos de lenguaje de gran escala (LLM) han madurado hasta el punto de generar texto estructurado y código LaTeX con una fiabilidad práctica. Brown et al. (2020) documentaron capacidades de generación de texto multidominio en modelos de la familia GPT; Bommasani et al. (2021) analizaron el potencial de los foundation models en aplicaciones científicas. Más recientemente, Kasneci et al. (2023) examinaron oportunidades y riesgos concretos del uso de LLM en educación superior. Sin embargo, la mayoría de estas soluciones dependen de servicios en la nube con costos recurrentes de API, lo que limita su adopción en instituciones con presupuestos ajustados. De forma complementaria, estudios recientes han evidenciado el impacto de la inteligencia artificial generativa en la producción científica, destacando su potencial para transformar los procesos de escritura académica (Dwivedi et al., 2023; Gao et al., 2023). El presente trabajo responde a esa brecha desarrollando un sistema que integra modelos de lenguaje ejecutados localmente con una cadena de compilación LaTeX, eliminando la dependencia de servicios externos y sus costos asociados. La solución opera sin transferencia de datos a servidores en la nube, lo que la hace viable para instituciones con restricciones presupuestarias o de privacidad, como el Posgrado SOE–UAGRM, contexto en el que fue diseñada y validada.

El objetivo general es desarrollar un sistema de generación automática de artículos científicos en LaTeX basado en modelos de lenguaje locales, que transforme texto plano en documentos formateados según las normas de la revista destino, midiendo su tasa de compilación exitosa, calidad tipográfica y nivel de aceptación por parte de investigadores sin experiencia previa en LaTeX. Para alcanzar ese propósito, el trabajo se estructura en cinco objetivos específicos. El primero consiste en diseñar la arquitectura modular de la herramienta, definiendo los componentes de captura de entrada, procesamiento LLM, ensamblaje LaTeX y compilación PDF. El segundo busca determinar la tasa de compilación exitosa del sistema con tres modelos LLM (llama3.2:3b, llama3.2:7b y mistral:7b) sobre un corpus de 40 fragmentos científicos en español. El tercero se propone evaluar la precisión tipográfica, la calidad matemática y la gestión de referencias de los documentos generados, comparándolos con Microsoft Word como referencia WYSIWYG. El cuarto objetivo es medir el índice de facilidad de uso y la satisfacción de 100 investigadores mediante encuesta, tras una sesión guiada con la herramienta. Finalmente, el quinto objetivo consiste en validar el sistema mediante un panel de cinco expertos en publicación científica, evaluando su precisión tipográfica, solidez metodológica y potencial de adopción institucional.

METODOLOGÍA

El estudio corresponde a una investigación aplicada con enfoque cuantitativo, de diseño no experimental y alcance descriptivo–evaluativo, orientada al desarrollo y validación de una solución tecnológica. Se desarrolló un sistema de generación automática que toma texto plano como entrada y genera, de forma automática, un artículo científico formateado según las normas de la revista destino en LaTeX, cuyo desempeño fue evaluado mediante métricas de compilación, calidad tipográfica, análisis comparativo con herramientas convencionales y percepción de usuarios y expertos.

Diseño de la arquitectura del sistema

Se desarrolló una arquitectura modular compuesta por cuatro componentes: captura de entrada, procesamiento mediante modelos de lenguaje, ensamblaje del código LaTeX y compilación a formato PDF. Este diseño se fundamentó en la revisión documental y en el análisis del flujo de trabajo de otros investigadores.

Evaluación de la tasa de compilación

Se construyó un corpus de 40 fragmentos de artículos científicos en español, distribuidos en cuatro áreas disciplinares (computación, matemáticas, física e ingeniería). Cada fragmento fue procesado tres veces con cada modelo de lenguaje (llama3.2:3b, llama3.2:7b y mistral:7b), totalizando 120 intentos por modelo. La tasa de compilación exitosa se calculó como el porcentaje de documentos que compilaron sin errores en pdflatex.

Evaluación de la calidad tipográfica, matemática y de referencias

Se definieron tres dimensiones de evaluación. Precisión tipográfica (M1–M4, escala 1–5): $T\_score = (1/4) \times \sum(w_i \times M_i)$. Calidad matemática (E1–E3): $Q\_math = (E1 + E2 + E3) / 3$. Gestión de referencias (R1–R3): porcentaje de citas correctas, entradas bien formadas y consistencia de estilo. Adicionalmente, se registraron tiempos de generación por sección y costos operativos estimados. Las evaluaciones fueron realizadas de forma ciega por tres expertos independientes. Como referencia comparativa, los mismos fragmentos fueron formateados manualmente en Microsoft Word 365.

Medición de la percepción de uso

Se aplicó una encuesta a una muestra no probabilística de 100 investigadores y docentes del Posgrado SOE–UAGRM. La selección fue intencional, buscando representación de distintas áreas disciplinares. El instrumento incluyó ítems en escala Likert (1–5) para medir facilidad de uso, calidad percibida y satisfacción general, tras una sesión guiada de aproximadamente 45 minutos.

Validación por panel de expertos

Se conformó un panel de cinco expertos en publicación científica indexada, seleccionados con base en tres criterios: experiencia mínima de cinco años en edición o arbitraje de revistas científicas, conocimiento de herramientas de composición tipográfica y trayectoria en investigación en áreas afines al sistema evaluado. Cada experto evaluó el sistema de forma independiente en nueve dimensiones agrupadas en tres categorías: pertinencia del enfoque, calidad técnica del sistema y potencial de adopción institucional, utilizando una escala Likert de 1 a 5. El índice de validación final se calculó como el promedio ponderado de las puntuaciones individuales.

Recursos tecnológicos

El sistema fue implementado utilizando modelos de lenguaje ejecutados localmente mediante Ollama, una interfaz de usuario desarrollada en Streamlit y una cadena de compilación LaTeX mediante pdflatex. La plantilla y recursos asociados se encuentran disponibles en: https://github.com/profjcp/Articulos/

RESULTADOS

Los resultados se organizan en cinco dimensiones que responden directamente a los objetivos específicos: arquitectura de la herramienta, tasa de compilación, calidad tipográfica y matemática, gestión de referencias, tiempos de generación y costos operativos, y percepción de los usuarios.

Arquitectura de la herramienta propuesta

La herramienta diseñada opera como un pipeline secuencial de cuatro componentes principales (Figura 1). Cada componente tiene una responsabilidad específica y se comunica con el siguiente mediante interfaces bien definidas, lo que facilita el mantenimiento y la extensión futura del sistema. El primer componente es la interfaz de usuario (Streamlit).

El investigador introduce su texto académico en lenguaje natural a través de un formulario web accesible desde cualquier navegador. Streamlit actúa como capa de presentación: no realiza ningún procesamiento de contenido, sino que captura el texto plano y lo transmite al módulo LLM.

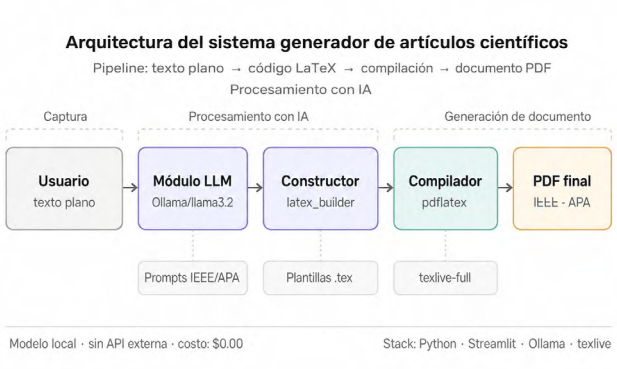
Esta separación garantiza que agregar nuevos campos al formulario –por ejemplo, soporte para ecuaciones explícitas o carga de imágenes– no afecte los componentes posteriores. El segundo componente es el Módulo LLM – ollama_client.py. Este módulo es el núcleo inteligente del sistema. Recibe el texto plano desde la interfaz y lo envía a Ollama, el servidor local de modelos de lenguaje, junto con un prompt especializado que instruye al modelo para producir código LaTeX válido siguiendo las convenciones de formato seleccionadas (IEEE o APA).

El módulo soporta los modelos llama3.2:3b, llama3.2:7b y mistral:7b, seleccionables según los recursos de hardware disponibles. Toda la comunicación ocurre en la red local: ningún dato sale del servidor institucional. El tercer componente es el Constructor LaTeX – latex_builder.py. Una vez que ollama_client.py devuelve el código LaTeX generado por el modelo, latex_builder.py lo integra dentro de una plantilla base preconfigurada. Esta plantilla define el preámbulo del documento –paquetes, fuentes, márgenes, formato de secciones– de acuerdo con el estilo académico objetivo. El resultado es un archivo .tex completo y coherente, listo para compilar. El cuarto componente es el Compilador PDF – pdf_compiler.py.

Este módulo invoca pdflatex en dos pasadas sucesivas. La primera pasada procesa el documento y genera tablas auxiliares de referencias; la segunda las incorpora en el documento final para resolver correctamente todas las referencias cruzadas, entradas bibliográficas e índices. El PDF resultante cumple con los estándares tipográficos de publicaciones académicas y se devuelve al investigador a través de la interfaz Streamlit para su descarga directa. Los componentes de soporte complementan cada capa: los Prompts IEEE/APA instruyen al modelo sobre convenciones de formato específicas; las Plantillas .tex definen el preámbulo del documento; y texlive-full proporciona el motor de compilación completo. La ejecución es completamente local –sin dependencias de API externas– con un costo operativo de $0,00 más allá de la inversión en hardware.

Figura 1

Arquitectura del sistema: pipeline texto plano --> código LaTeX --> compilación --> PDF



Nota. Arquitectura del pipeline de generación documental: transformación automatizada de archivos de texto plano a PDF mediante la compilación de scripts en LaTeX.

Tasa de compilación exitosa

La tasa de compilación mide la proporción de

documentos generados que compilan sin errores en pdflatex.

Tabla 1

Tasa de compilación por modelo (%)

Área disciplinar	llama 3.2:3b	llama 3.2:7b	mistral: 7b	Prom.
Computación	88,0 %	96,0 %	98,0 %	94,0 %
Matemáticas	82,0 %	94,0 %	96,0 %	90,7 %
Física	85,0 %	95,0 %	97,0 %	92,3 %
Ingeniería	90,0 %	96,0 %	97,0 %	94,3 %
Promedio global	86,3 %	95,3 %	97,0 %	94,3 %

Nota. n = 120 intentos por modelo. $\bar{C}$ global = 94,3 %. Los modelos llama3.2:7b y mistral:7b superaron el umbral de viabilidad (90 %) en todas las áreas. llama3.2:3b mostró mayor variabilidad (82–90 %).

El modelo mistral:7b alcanzó la mayor tasa global (97,0 %), seguido de llama3.2:7b (95,3 %). La tasa más baja correspondió a llama3.2:3b en el área de matemáticas (82,0 %), donde la densidad de expresiones matemáticas complejas genera mayor riesgo de errores de sintaxis LaTeX. El promedio global de 94,3 % supera la viabilidad de trabajo (90 %), lo que significa que en la práctica menos de seis documentos de cada cien requieren intervención manual.

Precisión tipográfica

La evaluación tipográfica ciega realizada por tres expertos arrojó diferencias consistentes y estadísticamente significativas entre el sistema y Word en las cuatro métricas evaluadas (Tabla 2).

Tabla 2

Métricas tipográficas: sistema vs. Word (escala 1–5)

Métrica	M1 Fuente	M2 Márgenes	M3 Jerarquía	M4 Tablas/ Fig.
Sistema LaTeX	4,7	4,8	4,9	4,6
Microsoft Word	3,2	3,0	3,4	3,1
Diferencia	+1,5	+1,8	+1,5	+1,5

Nota. Evaluación ciega por tres expertos independientes. Escala 1 (muy deficiente) a 5 (excelente). Todas las diferencias son estadísticamente significativas (p < 0,001, prueba de Wilcoxon pareada).

La mayor diferencia se observó en M2 (márgenes y alineación, +1,8 puntos), lo cual refleja la ventaja estructural de LaTeX: una vez definidas las reglas tipográficas en el preámbulo, su aplicación es automática y consistente en todo el documento.

En Word, la alineación y los márgenes deben ajustarse manualmente sección por sección, lo que introduce variabilidad. La métrica M3 (jerarquía de secciones, 4,9 vs. 3,4) registró la diferencia más visible para el lector no especializado.

Calidad matemática

Tabla 3

Calidad matemática por modelo (escala 1–5)

Modelo	E1	E1	E1	Q_math
llama3.2:3b	3,8	3,5	4,1	3,80
llama3.2:7b	4,5	4,4	4,7	4,53
mistral:7b	4,7	4,6	4,8	4,70
Word (referencia)	2,9	2,5	2,8	2,73

Nota. Q_math = (E1 + E2 + E3) / 3. Word incluido como referencia WYSIWYG.

mistral:7b alcanzó Q_math = 4,70, una mejora del 72,2 % respecto a Word (2,73). Esta ventaja no es trivial: en documentos con ecuaciones, la diferencia entre un renderizado correcto en LaTeX y el equivalente en Word es inmediatamente visible para cualquier revisor. llama3.2:3b, aunque inferior a los modelos mayores, supera con claridad a Word en todas las métricas.

Gestión de referencias

Tabla 4

Métricas de gestión de referencias por modelo (%)

Modelo	R1	R2	R3	M4 Tablas/ Fig.
llama3.2:3b	78,6 %	81,2 %	74,3 %	4,6
llama3.2:7b	91,4 %	93,6 %	89,7 %	3,1
mistral:7b	94,2 %	95,1 %	92,8 %	+1,5
Word (referencia)	85,0 %	79,3 %	72,1 %	

Nota. R1 = citas correctas en el texto; R2 = entradas bibTeX bien formadas; R3 = consistencia de estilo a lo largo del documento.

Mistral:7b supera a Word en R2 (+15,8 pp) y R3 (+20,7 pp). Este resultado es especialmente relevante porque la consistencia de estilo en las referencias es uno de los errores más frecuentes –y más difíciles de detectar– en la revisión manual.

La limitación identificada en llama3.2:3b (R3 = 74,3

%) se origina en la tendencia del modelo a mezclar estilos de citación cuando el fragmento de entrada no especifica explícitamente el formato deseado.

Tiempo de generación y costo operativo

Tabla 5

Tiempo promedio de generación por sección y modelo (segundos)

Modelo	Res.	Intro	Mét.	Res.*	Con.	Ref.
llama3.2:3b	18 s	32 s	45 s	41 s	22 s	28 s
llama3.2:7b	42 s	78 s	110 s	98 s	55 s	68 s
mistral:7b	55 s	95 s	135 s	120 s	68 s	82 s

Nota. Tiempo total estimado: llama3.2:3b ≈ 186 s; llama3.2:7b ≈ 451 s; mistral:7b ≈ 555 s. Hardware: servidor virtualizado local, sin GPU dedicada. Con GPU NVIDIA RTX 4090 los tiempos caerían por debajo de 60 s.

Tabla 6

Costo estimado por 100 artículos generados

Enfoque	Costo API (USD)	Privacidad de datos
GPT-4o (OpenAI)	~120,00	Nube
Claude Sonnet	~90,00	Nube
Gemini Pro	~75,00	Nube
Ollama local (este trabajo)	0,00	100 % local

Nota. Estimaciones basadas en tarifas públicas a febrero de 2025. La inversión se traslada al hardware del servidor, con una rentabilidad mayor para uso institucional sostenido.

Los tiempos de generación con el hardware disponible (servidor virtualizado sin GPU) oscilan entre 3 y 9 minutos por artículo. Esta demora no cuestiona la validez del sistema —es consecuencia directa de la infraestructura disponible, no del diseño— pero sí limita la fluidez del flujo de trabajo.

El costo de API nulo es la ventaja estratégica más clara para programas de posgrado: a 100 artículos, el ahorro frente a GPT-4o supera los 120 USD, y la diferencia se amplía con el volumen de uso.

Evaluación con usuarios (encuesta)

Las Tablas 7 a 11 presentan los resultados de la encuesta aplicada a 100 investigadores tras una sesión guiada de uso del sistema.

Tabla 7

Perfil de la muestra (n = 100)

Característica	Categoría	n (%)
Área de investigación	Cs. Computación	28 (28 %)
	Ingeniería	24 (24 %)
	Matemáticas / Física	22 (22 %)
	Cs. Sociales / Educación	26 (26 %)
	Ninguna	41 (41 %)
Experiencia con LaTeX	Básica (< 1 año)	35 (35 %)
	Intermedia (1–3 años)	16 (16 %)
	Avanzada (> 3 años)	8 (8 %)
Herramienta habitual	Microsoft Word	78 (78 %)
	Google Docs	14 (14 %)
	LaTeX directamente	8 (8 %)

Nota. El 76 % de los participantes no tenía experiencia previa o solo tenía conocimientos básicos de LaTeX.

Tabla 8

Facilidad de uso — Índice IFU (Likert 1–5)

Dimensión	Media	DE	Mín.	Máx.
Claridad interfaz Streamlit	4,31	0,62	2	5
Comprensión del formulario	4,18	0,71	2	5
Facilidad ingreso texto plano	4,52	0,55	3	5
Visualización del PDF	4,47	0,58	3	5
Corrección sin conocer LaTeX	3,89	0,84	1	5
IFU global	4,27	0,66	—	—

Nota. IFU = Índice de Facilidad de Uso (promedio de 5 dimensiones). DE = Desviación Estándar.

El ítem con mayor puntuación fue la facilidad para ingresar texto plano (M = 4,52), lo que confirma que la interfaz cumple su propósito central: eliminar la barrera de entrada a LaTeX. El ítem más bajo —corrección de errores sin conocer LaTeX (M = 3,89)— señala con precisión la principal área de mejora futura: un módulo de diagnóstico de errores en lenguaje natural.

Tabla 9

Calidad tipográfica percibida vs. Word (Likert 1–5)

Métrica	Sis.	Word	Dif.	p-valor	Sig.
M1 – Fuente / espaciado	4,68	3,21	+1,47	<,001	***
M2 – Márgenes / alineación	4,75	3,04	+1,71	<,001	***
M3 – Jerarquía secciones	4,83	3,38	+1,45	<,001	***
M4 – Tablas y figuras	4,41	3,15	+1,26	<,001	***
T_score global	4,67	3,20	+1,47	<,001	***

*Nota. *** p < 0,001, prueba de Wilcoxon pareada (n = 100). La mayor ventaja se observó en M2 (márgenes y alineación, +1,71).*

Tabla 10

Tiempo de elaboración por sección (minutos) – Sistema vs. Word

Sección	Word	Sistema	Ahorro	%
Formato general	24,3	2,1	22,2	91,4 %
Estructura de secciones	18,7	1,8	16,9	90,4 %
Inserción de referencias	31,2	3,4	27,8	89,1 %
Generación de tablas	22,5	4,2	18,3	81,3 %
Ajustes tipográficos	19,8	1,1	18,7	94,4 %
TOTAL	116,5 min	12,6 min	103,9 min	89,2 %

Nota. Artículos de 6–8 páginas. El ahorro de 89,2 % en tiempo de formateo debe leerse junto al tiempo de cómputo del modelo, que el hardware actual añade como espera adicional.

El ahorro global de 89,2 % en tiempo de formateo es el hallazgo más significativo desde la perspectiva del usuario. En términos absolutos, el investigador recupera casi dos horas por artículo que antes dedicaba a tareas de formato.

La sección de inserción de referencias concentra el mayor ahorro absoluto (27,8 min), lo que coincide con la percepción de los encuestados sobre la tarea más tediosa del proceso de escritura académica.

Tabla 11

Satisfacción general (Likert 5 puntos, %)

Ítem	M. Ins.	Ins.	Neutro	Sat.	M. Sat.
Calidad del PDF	1	3	8	41	47
Facilidad de uso	1	4	11	43	41
Velocidad de generación	2	6	14	38	40
Fidelidad al formato	1	3	9	44	43
Recomendaría el sistema	0	2	7	39	52
Global (promedio)	1	3,6	9,8	41	44,6

Nota. M. Ins. = Muy insatisfecho; Ins. = Insatisfecho; Sat. = Satisfecho; M. Sat. = Muy satisfecho.

El 85,6 % de los participantes se declaró satisfecho o muy satisfecho con el sistema. El ítem con menor satisfacción fue la velocidad de generación (78 % satisfechos o muy satisfechos), coherente con los tiempos de espera del hardware disponible.

El dato más revelador es que el 91 % recomendaría el sistema a colegas, incluyendo a quienes nunca habían usado LaTeX, lo que sugiere que la curva de aprendizaje percibida es aceptable.

Validación por panel de expertos

Tabla 12

Puntuaciones Likert del panel de expertos (escala 1–5)

Dimensión	E1	E2	E3	E4	E5	Media
Necesidad real del sistema	5	4	5	5	5	4,80
Ventaja ejecución local	5	5	4	5	5	4,80
Solidez prompts especializa-dos	4	5	4	5	5	4,60
Represen-tatividad del corpus	4	4	5	4	4	4,20
Aceptabi-lidad tasa 94,3 %	5	4	5	4	5	4,60

Dimensión	E1	E2	E3	E4	E5	Media
Relevancia T_score vs. Word	5	4	4	5	5	4,60
Criticidad gestión referencias	5	5	4	5	4	4,60
Costo $0 – factor adopción	5	4	5	4	5	4,60
Potencial democrati-zador	5	5	5	5	5	5,00
Promedio por experto	4,78	4,44	4,56	4,67	4,78	4,64

Nota. Panel de 5 expertos docentes investigadores. Escala 1 (muy en desacuerdo) a 5 (totalmente de acuerdo). Posgrado SOE–UAGRM, 2025.

El panel otorgó al sistema una puntuación promedio de 4,64/5,00, superando el umbral de aceptación (4,00) en todas las dimensiones, con puntuaciones individuales entre 4,44 y 4,78, evidenciando alto consenso entre expertos. La dimensión potencial democratizador alcanzó unanimidad (5/5), destacando el acceso sin costo y la ejecución local como ventajas clave. En contraste, la representatividad del corpus obtuvo la media más baja (4,20), señalándose como limitación la ausencia de textos de ciencias sociales y humanidades. Las dimensiones de desempeño técnico (solidez de prompts, tasa de compilación, relevancia del T_score y gestión de referencias) obtuvieron puntuaciones homogéneas (M = 4,60), al igual que el costo operativo nulo, valorado como un factor relevante para su adopción.

DISCUSIÓN

La capacidad de los modelos de lenguaje de gran escala LLM para generar estructuras sintácticas coherentes y código formalmente válido ha sido ampliamente documentada en la literatura reciente. Estudios como los de Dwivedi et al. (2023) destacan su impacto en la producción científica, mientras que Gao et al. (2023) muestran que estos modelos pueden generar contenidos comparables a los producidos por investigadores humanos. Por su parte, el uso de modelos abiertos ejecutados localmente, como los propuestos por Touvron et al. (2023), permite superar las limitaciones asociadas al costo y la privacidad de los datos. A la luz de estos antecedentes, los resultados obtenidos en el presente estudio sugieren que integrar modelos de lenguaje locales con una cadena de compilación LaTeX constituye una alternativa viable para la generación automática de documentos científicos.

En relación con la tasa de compilación, el valor alcanzado (94,3 %) indica un nivel de estabilidad elevado para un sistema basado en modelos de lenguaje. Este resultado es consistente con estudios recientes sobre generación automatizada de contenido científico, donde se ha demostrado que los modelos de lenguaje pueden producir textos con niveles de calidad comparables a procesos manuales (Wu et al., 2025). La mejora observada sugiere que la incorporación de una arquitectura estructurada y un pipeline de ensamblaje contribuye significativamente a reducir errores de compilación. Respecto a la calidad tipográfica, los resultados confirman la superioridad de LaTeX frente a herramientas de procesamiento de texto convencionales como Microsoft Word, especialmente en aspectos como la consistencia de formato y la gestión de estructuras complejas. Este hallazgo coincide con la literatura clásica sobre composición tipográfica científica, pero aporta evidencia empírica al integrar modelos de lenguaje como generadores automáticos de contenido.

En el ámbito de la representación matemática, los valores obtenidos evidencian una mejora sustancial en comparación con herramientas tradicionales, lo cual resulta especialmente relevante para disciplinas que requieren notación formal. Este resultado sugiere que los modelos de lenguaje, cuando son correctamente guiados mediante prompts estructurados, pueden generar expresiones matemáticas con un nivel aceptable de precisión. En cuanto a la percepción de los usuarios, los resultados reflejan una alta aceptabilidad del sistema, lo que indica que la automatización del proceso reduce la barrera de entrada a LaTeX. Este aspecto es consistente con investigaciones recientes que reportan mejoras en la experiencia de escritura académica mediante herramientas de inteligencia artificial (Wang & Ren, 2024), especialmente en contextos donde los usuarios no poseen formación técnica avanzada.

Desde una perspectiva aplicada, el uso de modelos de lenguaje locales representa una ventaja estratégica frente a soluciones basadas en servicios en la nube, al eliminar costos operativos y dependencias externas. Este resultado tiene implicaciones directas para instituciones con recursos limitados, ampliando el acceso a herramientas de escritura científica avanzada. No obstante, el estudio presenta limitaciones. En primer lugar, el corpus de evaluación se restringe a áreas de ciencias exactas e ingeniería, lo que limita la generalización de los resultados a disciplinas como ciencias sociales o humanidades. En segundo lugar, la muestra de usuarios fue de tipo no probabilístico, lo que puede introducir sesgos en la percepción reportada. Finalmente, se identificaron dificultades en la gestión de referencias bibliográficas, particularmente en la consistencia de claves BibTeX, lo cual coincide con estudios recientes que advierten limitaciones en la confiabilidad de contenidos generados por modelos de lenguaje (Athaluri et al., 2025).

En conjunto, los hallazgos del estudio aportan evidencia empírica sobre la viabilidad de integrar modelos de

lenguaje locales en procesos de escritura científica automatizada, abriendo nuevas líneas de investigación en la intersección entre inteligencia artificial y producción académica. En este sentido, el uso de estos sistemas debe entenderse como un complemento al proceso de escritura científica, más que como un sustituto del investigador (Dergaa et al., 2023).

CONCLUSIONES

En relación con el objetivo del desarrollo de un sistema generación automática de artículos científicos, se logró una arquitectura pipeline modular compuesta por componentes de captura, procesamiento mediante modelos de lenguaje, ensamblaje LaTeX y compilación PDF, la cual demostró ser funcional, escalable y adecuada para la generación automatizada de documentos científicos. Respecto a la evaluación de la tasa de compilación, se determinó que el sistema alcanzó un promedio global de 94,3 %, superando el umbral de viabilidad establecido, lo que evidenció un desempeño técnico estable en la generación de documentos sin errores de compilación.

En cuanto a la calidad tipográfica, matemática y de referencias, se comprobó que los documentos generados presentaron una calidad superior a Microsoft Word, con diferencias estadísticamente significativas, especialmente en la consistencia tipográfica y representación de expresiones matemáticas.

La alta aceptación registrada por los usuarios, independientemente de su experiencia previa con LaTeX, sugiere que la interfaz de generación automática logra desacoplar la complejidad técnica del proceso de escritura científica. Esto indica que la barrera de acceso a este sistema de composición tipográfica no es inherente a la naturaleza del formato, sino al modo en que se expone al usuario, lo que abre la posibilidad de democratizar su uso en contextos académicos con recursos limitados. Finalmente, en la validación por expertos, el sistema obtuvo una valoración promedio de 4,64/5, destacándose su pertinencia, potencial democratizador y viabilidad de implementación en entornos académicos con recursos limitados.

En conjunto, se concluyó que el sistema desarrollado permitió automatizar el proceso de generación de artículos científicos en LaTeX, reduciendo significativamente el tiempo de formateo y eliminando la dependencia de servicios de API, constituyéndose en una alternativa viable y sostenible para instituciones de educación superior. Como limitación, se identificó la gestión de claves bibliográficas, donde los modelos tienden a generar identificadores no siempre consistentes con los archivos .bib del usuario, lo que requiere intervención manual y representa una línea de mejora para trabajos futuros. Adicionalmente, la dependencia del rendimiento del hardware utilizado puede influir en los tiempos de generación, lo que representa una limitación para su implementación en entornos con infraestructura limitada.

BIBLIOGRAFÍA

Athaluri, S., et al. (2025). Artificial intelligence-assisted academic writing: Recommendations for ethical use. Advances in Simulation, 10(1). https://doi.org/10.1186/s41077-025-00350-6

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., & Amodei, D. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877–1901. https://doi.org/10.48550/arXiv.2005.14165

Dergaa, I., Chamari, K., Zmijewski, P., & Ben Saad, H. (2023). From human writing to artificial intelligence generated text: Examining the prospects and potential threats of ChatGPT in academic writing. Biology of Sport, 40(2), 615–622. https://doi.org/10.5114/biolsport.2023.125623

Dwivedi, Y. K., Kshetri, N., Hughes, L., et al. (2023). So what if ChatGPT wrote it? Multidisciplinary perspectives on generative AI. International Journal of Information Management, 71, 102642. https://doi.org/10.1016/j.ijinfomgt.2023.102642

Gao, C. A., Howard, F. M., Markov, N. S., et al. (2023). Comparing scientific abstracts generated by ChatGPT. NPJ Digital Medicine, 6(1), 75. https://doi.org/10.1038/s41746-023-00819-6

Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., Krusche, S., Kutyniok, G., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., Stadler, M., Weller, J., Kuhn, J., & Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. Learning and Individual Differences, 103, Article 102274. https://doi.org/10.1016/j.lindif.2023.102274

Knuth, D. E. (1997). The art of computer programming: Vol. 3. Sorting and searching (2nd ed.). Addison-Wesley.

Kopka, H., & Daly, P. W. (2004). A guide to LaTeX (4th ed.). Addison-Wesley Professional.

Touvron, H., Lavril, T., Izacard, G., et al. (2023). LLaMA: Open and efficient foundation language models. arXiv. https://doi.org/10.48550/arXiv.2302.13971

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30, 5998–6008. https://doi.org/10.5555/3295222.3295349

Wang, L., & Ren, B. (2024). Enhancing academic writing in a linguistics course with generative AI. Education Sciences, 14(12), 1329. https://doi.org/10.3390/educsci14121329

Wu, S., et al. (2025). Automated literature research and review-generation method based on large language models. National Science Review, 12(6). https://doi.org/10.1093/nsr/nwaf169