Pontificia Universidad Católica Del Ecuador

Facultad De Hábitat, Infraestructura Y Creatividad

Maestría En Sistemas De Información Mención Data Science



Trabajo previo a la obtención del título de Magister en Sistemas de Información mención Data Science

Tema:

"Diseño de un sistema de calificación de ejercicios matemáticos manuscritos mediante deep learning y visión por computador"

Autor:

Ing. Miguel Santiago Jingo P.

Tutor:

PhD. Carlos Julio Mayorga A.

Quito – Agosto del 2026

Dedicatoria

Con profunda humildad y agradecimiento; dedico este trabajo primeramente a 'Dios', por ser mi bote salvavidas en medio de cada tormenta y la certeza serena que sostuvo mi 'fe' cuando el horizonte se volvía incierto. A mi familia, mi fuente de amor y fuerza, a todos aquellos que creyeron en mí incluso cuando yo mismo dudaba. Porque ningún logro se construye en soledad, porque detrás de cada página hay manos que sostuvieron, voces que alentaron y silencios que acompañaron.

Así también, dedico con todo mi corazón esta tesis a mi madre; por ser mi faro en este largo camino, pues sin ella no lo habría logrado. Su bendición diaria a lo largo de mi vida me protege y me lleva por el camino del bien. De ella aprendí que la perseverancia es una forma callada de amor y que no hace falta hablar demasiado para ser escuchado, que los sueños no se heredan, sino que se trabajan duro y que la 'fe' puesta en el esfuerzo siempre encuentra su recompensa. Por eso entrego este trabajo en ofrenda 'madre mía', por su paciencia y su amor.

A 'Mary', cuya presencia en mi vida ha hecho que cada desafío sea más fácil de superar y cada alegría más intensa de celebrar, estando ahí siempre aun cuando no me lo merecía. Gracias por su paciencia, su comprensión y por hacer especial cada momento, enseñándome que siempre se puede volver a creer y a confiar, que el amor verdadero se demuestra en la constancia de los días.

Que estas páginas sean testimonio de que los esfuerzos silenciosos también florecen, y de que toda meta alcanzada es en el fondo la suma de quienes caminaron a nuestro lado siempre.

Miguel J.

Agradecimientos

Agradezco a la Pontificia Universidad Católica del Ecuador, «Sede Quito», casa de estudios que me abrió sus puertas y puso a mi disposición los recursos y el espacio académico sin los cuales este proyecto de investigación no habría sido posible.

De manera especial, expreso mi gratitud al PhD. Carlos Julio Mayorga A., tutor de esta tesis. Su guía académica y su rigor metodológico marcaron el rumbo de este trabajo; pero su generosidad con el tiempo, su paciencia inagotable y, sobre todo, su acompañamiento a lo largo del proceso fueron determinantes para dar forma y profundidad a estas páginas.

A los docentes de la Maestría en Data Science y de la Facultad de Hábitat, Infraestructura y Creatividad, mi reconocimiento sincero. Cada uno aportó una pieza del criterio con el que hoy miro los datos, y esa formación se encuentra disuelta en todo el documento, aunque no lleve sus firmas en ninguna página.

Y a mis compañeros de aula, con quienes compartí la incertidumbre y la aventura de un mundo nuevo en los primeros meses, y la satisfacción de ver el trabajo tomar forma.

Índice

Dedicatoria I

Agradecimientos II

Índice III

Índice de códigos X

Capítulo I: Introducción 1

Antecedentes 1

Planteamiento del Problema 2

Identificación del Problema 2

Delimitación del Problema 3

Claridad y Adecuación del Planteamiento 4

Justificación 4

Formulación de Preguntas 5

Pregunta Principal 5

Preguntas Secundarias 5

Relevancia de las Preguntas Formuladas 6

Resolución del Problema 7

Objetivos 7

Objetivo General 8

Objetivos Específicos 8

Viabilidad del Estudio 10

Alcance y Limitaciones del Estudio 10

Capítulo II: Marco Teórico y Revisión Literaria 14

Introducción del Capítulo 14

Estrategia de Revisión de la Literatura 15

Bases de Datos y Fuentes Académicas 16

Cadenas de Búsqueda 17
Criterios de Inclusión y Exclusión 18
Registro Tipo PRISMA 19
Antecedentes de Investigación 21
Modelos Encoder – Decoder 21
Consolidación del Transformer 22
Validación del Procedimiento 24
Conjuntos de Datos Públicos y Benchmarks 25
Bases Conceptuales 29
Expresiones Matemáticas Manuscritas 30
Visión por Computador y Reconocimiento de Imágenes 30
Deep Learning (DL) 32
Segmentación 32
Preservación de la Jerarquía Espacial 32
Razonamiento Matemático y Métodos de Resolución 33
Calificación Automática con Validación del Procedimiento 34
Modelos y Enfoques Teóricos 37
Aprendizaje Profundo y Entrenamiento 37
Encoder – Decoder con Atención (Enfoque Arquitectónico) 41
Gramática Bidimensional y Árboles de Derivación (Enfoque Estructural) 42
Juicio Humano y Analítica del Aprendizaje (Enfoque Evaluativo) 43
Marco Metodológico del Ciclo de Datos (CRISP-DM) 44
Variables, Dimensiones ó Categorías de Análisis 45
Métricas OE 46
Métricas de Reconocimiento (OE1) 47
Métricas de Segmentación Estructural (OE2) 47
Métricas de Validación del Procedimiento (OE3) 48
Brechas de Investigación 49
Síntesis del Marco Teórico 50

Capítulo III: Metodología **52**

Introducción del Capítulo . . . 52

Enfoque de la Investigación . . . 52

Diseño de la Investigación . . . 53

Población, Muestra y Fuentes de Datos . . . 53

Población . . . 53

Muestra y Marco Muestral . . . 54

Estrategia de Muestreo y Particiones . . . 55

Técnicas de Recolección de Datos . . . 56

Operacionalización de las Variables . . . 56

Técnicas e Instrumentos . . . 57

Procesamiento y Análisis de los Datos . . . 58

Preparación de los Datos . . . 58

Modelos y Técnicas Analíticas por Objetivo . . . 59

Protocolo Experimental y Análisis Estadístico . . . 59

Métricas y Umbrales Referenciales . . . 60

Consideraciones Éticas y Legales . . . 61

Síntesis del Capítulo . . . 62

Capítulo IV: Resultados y Análisis **63**

Introducción del Capítulo . . . 63

Condiciones de la Corrida Experimental . . . 63

Precisiones Sobre las Configuraciones . . . 65

Caracterización del Dataset . . . 65

Resultados del OE1: Comparación de Arquitecturas . . . 67

Contraste entre Validación y Prueba . . . 73

Resultados del OE2: Segmentación y Jerarquía Espacial . . . 75

Resultados del OE3: Validación del Procedimiento . . . 77

Análisis . . . 80

Por Qué Gana la Arquitectura Recurrente . . . 80

Resultados Sobre la Segmentación 81
Resultados de los Algoritmos frente a la Línea Base 81
Interpretación de los Hallazgos 83
Síntesis del Capítulo 84

Capítulo V: Conclusiones y Recomendaciones **85**
Introducción del Capítulo 85
Conclusión sobre el OE1: Comparar Arquitecturas 85
Conclusión sobre el OE2: Preservar la Jerarquía Espacial 86
Conclusión sobre el OE3: Validar el Procedimiento 87
Limitaciones 88
Recomendaciones 88
Hacia una Herramienta Utilizable 89

Referencias **91**

Anexo A: Scripts y Código Fuente **96**
Organización del Repositorio 96
Estructura del Cuaderno 96
Fragmentos de Código Relevantes 97
Configuración Común 98
OE1: Comparación de Arquitecturas 99
OE2: Segmentación 103
OE3: Validación del Procedimiento 104

Anexo B: Ficha Técnica de los Conjuntos de Datos **107**
Identificación y Procedencia 107
Licencias, Ética y Disponibilidad 107

Índice de figuras

1. Diagrama de flujo PRISMA 15
2. Diagrama de flujo PRISMA adaptado al proyecto 20
3. Example of the model generating mathematical markup 22
4. Relación jerárquica entre deep learning y el problema de estudio 31
5. Flujo de calificación automática 35
6. Esquema de bloques de una red neuronal 38
7. Modelo codificador – decodificador 41
8. Árbol de disposición de símbolos de una expresión 42
9. Arquitectura conceptual del sistema de calificación 44
10. Visualización de CROHME 1 66
11. Visualización de CROHME 2 67
12. Convergencia en validación 69
13. Desempeño en validación 69
14. Segmentación frente a la verdad de campo 76

Índice de tablas

1. Delimitación del problema de investigación 3
2. Correspondencia entre brechas del problema 7
3. Matriz de alineación 9
4. Alcance y exclusiones del estudio 11
5. Bases de datos y fuentes académicas consultadas 17
6. Bloques temáticos y términos de búsqueda 18
7. Criterios de inclusión y exclusión de la revisión 19
8. Registro tipo PRISMA del proceso de revisión 20
9. Evolución comparativa de arquitecturas de reconocimiento 24
10. Matriz de extracción de la literatura 1 26
11. Matriz de extracción de la literatura 2 27
12. Matriz de análisis de la literatura 1 28
13. Matriz de análisis de la literatura 2 29
14. Métodos de demostración y resolución 34
15. Bases conceptuales del estudio 36
16. Alineación entre CRISP-DM y los objetivos 45
17. Variables, dimensiones e indicadores 46
18. Escala de interpretación del coeficiente kappa 48
19. Síntesis de la brecha de investigación 1 49
20. Síntesis de la brecha de investigación 2 50
21. Lista de verificación del capítulo 51
22. Fuentes de datos según objetivo específico 54
23. Matriz de operacionalización de variables 56
24. Matriz de operacionalización de variables 57
25. Métricas y umbrales por objetivo específico 61
26. Condiciones de la corrida experimental 64
27. Caracterización de CROHME 66
28. Desempeño de las arquitecturas 68

29. Intervalos de confianza del OE1 71
30. Prueba de McNemar entre arquitecturas 72
31. Solapamiento de contenido entre particiones 73
32. ExpRate por población de expresiones 74
33. Calidad de la segmentación 75
34. Aporte de la validación 78
35. Calificación por prueba 79
36. Concordancia con el docente 79
37. Cumplimiento de los umbrales referenciales 83
38. Estructura de archivos del pipeline experimental 96
39. Secciones del cuaderno experimental 97
40. Trazabilidad entre código y resultados 98
41. Identificación de los conjuntos de datos 107

Índice de códigos

1. Semilla, hiperparámetros comunes y registro del entorno 98
2. Vocabulario y partición por escritor (OE1) 99
3. Codificador común y las tres cabezas de decodificación (OE1) 100
4. Cálculo de exactitud, F1-score, CER y ExpRate (OE1) 101
5. Prueba de McNemar e intervalos de confianza por remuestreo (OE1) . . . 102
6. Derivación de las cajas de referencia a partir del InkML (OE2) 103
7. Segmentación por componentes conectados con fusión de multitrazo (OE2) 103
8. Índice de Jaccard y emparejamiento voraz de las cajas (OE2) 104
9. Verificación de equivalencia entre líneas consecutivas (OE3) 104
10. Nota, umbral de aprobación y concordancia con el docente (OE3) 105

Capítulo I: Introducción

Antecedentes

Corregir ejercicios matemáticos escritos a mano es una de las tareas que más tiempo y energía demandan a los docentes. Cuando un profesor debe revisar decenas o incluso cientos de hojas —también se lo puede observar en el ámbito académico superior—, la calificación manual no solo retrasa la retroalimentación que recibe el estudiante, sino que también introduce variaciones inevitables en los criterios de evaluación. Ante este escenario, la combinación entre ciencia de datos, deep learning y visión por computador se presenta como una oportunidad para automatizar procesos que tradicionalmente han dependido únicamente del juicio humano.

En los últimos años, los avances en redes neuronales convolucionales (CNN), en el reconocimiento de texto manuscrito (HTR) y en modelos de detección de objetos han permitido que los sistemas automáticos alcancen niveles de precisión muy cercanos al de los humanos en tareas de interpretación visual. Sin embargo, aplicar estas tecnologías al reconocimiento de expresiones matemáticas manuscritas sigue siendo un reto; debido principalmente a la mezcla de símbolos, números, operadores y estructuras bidimensionales que plantea dificultades como la segmentación correcta de cada elemento, la interpretación de la jerarquía matemática y la verificación del procedimiento seguido por el estudiante.

Esta investigación propone diseñar un sistema capaz de calificar automáticamente ejercicios matemáticos manuscritos utilizando técnicas de deep learning (DL) y visión por computador (CV). La premisa no es solo reconocer los símbolos escritos, sino también evaluar si el procedimiento es válido, integrando procesamiento de imágenes, modelos predictivos y análisis estructural de expresiones matemáticas. El proyecto se enmarca dentro de la investigación aplicada en Data Science, orientada a resolver un problema real mediante herramientas basadas en conocimiento científico y tecnológico.

Por tanto, en el contexto de la formación académica de la Maestría en Data Science, la fase inicial de todo proyecto exige delimitar con precisión el problema de

investigación y derivar de él preguntas claras, viables y medibles. El planteamiento del problema constituye la base sobre la cual se construye toda la indagación científica y condiciona la elección de la metodología, las técnicas de análisis y los criterios de validación de resultados (Hernández Sampieri y Mendoza Torres, 2023).

Planteamiento del Problema

La corrección de ejercicios matemáticos manuscritos a nivel de educación media es una de las actividades que más tiempo y energía demanda al docente y más aún cuando un profesor debe revisar decenas o cientos de hojas; la calificación manual no solo retrasa la retroalimentación que recibe el estudiante, sino que también introduce variaciones inevitables en los criterios de evaluación. En contextos como el nuestro (ecuatoriano), donde un docente suele atender a numerosos estudiantes por periodo, esta carga afecta directamente la calidad del proceso de enseñanza – aprendizaje.

En los últimos años, los avances en redes neuronales convolucionales (CNN), reconocimiento de texto manuscrito (Handwritten Text Recognition, HTR) y modelos de detección de objetos han permitido que los sistemas automáticos alcancen niveles de precisión cercanos al de los humanos en tareas de interpretación visual. Sin embargo, su aplicación al reconocimiento de expresiones matemáticas manuscritas sigue siendo un reto científico abierto, dado que la notación matemática combina símbolos heterogéneos, estructuras bidimensionales y jerarquías espaciales complejas como subíndices, exponentes, fracciones y radicales.

Identificación del Problema

En el escenario descrito se identifican cuatro limitaciones críticas del proceso actual de calificación manual: lentitud en la retroalimentación al estudiante, inconsistencia en los criterios de evaluación derivada de la subjetividad del evaluador, carga de trabajo por el número de estudiantes y cursos asignados al maestro e imposibilidad práctica de evaluar el procedimiento paso a paso, lo que empobrece la retroalimentación y reduce la calificación a la mera verificación del resultado final. A esto se suma que los sistemas HTR actuales, optimizados para texto convencional, no resuelven adecuadamente la naturaleza bidimensional de la notación matemática ni

validan la coherencia lógica del procedimiento.

En consecuencia, existe una dificultad concreta en el ámbito educativo y esto particularmente en instituciones con alta carga docente relacionada con la ausencia de sistemas automáticos capaces de calificar ejercicios matemáticos manuscritos evaluando tanto la respuesta final como la validez del procedimiento intermedio, lo que genera retrasos en la retroalimentación, sobrecarga laboral docente y heterogeneidad en los criterios de evaluación.

Delimitación del Problema

Para garantizar la viabilidad y el enfoque del estudio, el problema se delimita en términos de fenómeno, variables, contexto, tipo de investigación y datos, como se muestra en la Tabla 1.

Tabla 1

Delimitación del problema de investigación

Dimensión	Especificación
Fenómeno	Calificación automatizada de ejercicios matemáticos manuscritos.
Variables clave	Precisión del reconocimiento simbólico, exactitud de segmentación estructural, validez del procedimiento y concordancia con la calificación docente.
Contexto	Ejercicios propios de la educación formal de nivel medio, resueltos a mano y capturados en condiciones controladas, con vistas a su aplicación en el contexto ecuatoriano.
Tipo de investigación	Enfoque cuantitativo y carácter experimental.
Datos	Imágenes de ejercicios manuscritos provenientes del conjunto público CROHME y de una muestra propia resuelta por escritores adultos voluntarios.

Esta delimitación se realiza con base en la estructura recomendada para el

presente proyecto de investigación, según la cual un problema de investigación debe ser específico, situado en un contexto real, vinculado a variables medibles y resoluble mediante datos concretos.

Claridad y Adecuación del Planteamiento

El planteamiento se considera adecuadamente definido por cumplir con los seis (6) criterios de calidad como: especificidad, identificación del contexto, claridad de variables, viabilidad, identificación del vacío de conocimiento e impacto definido. La formulación no se limita a una intención general como «mejorar la educación con inteligencia artificial (IA)», sino que precisa el fenómeno (calificación), el objeto (ejercicios matemáticos manuscritos), la tecnología (deep learning y visión por computador), el contexto (educación formal en escenarios de alta carga docente) y las variables medibles que orientan la recolección de datos concretos y específicos.

Justificación

La relevancia de este estudio puede entenderse desde tres perspectivas: académica, tecnológica y social.

- Desde lo académico: esta investigación aborda una brecha clara en la literatura sobre reconocimiento automático de expresiones matemáticas manuscritas. Aunque el HTR ha avanzado mucho para texto convencional, la notación matemática presenta desafíos propios: subíndices, exponentes, fracciones, radicales y una disposición espacial compleja que exige enfoques especializados. Este proyecto aportará evidencia sobre el rendimiento de distintas arquitecturas de deep learning (DL) aplicadas a este tipo de problemas, generando conocimiento útil y replicable.

- En el plano tecnológico: el desarrollo de un sistema funcional de calificación automática constituye una aplicación directa de los principios de Data Science al ámbito educativo. Así mismo, integrar visión por computador con modelos predictivos permitirá demostrar la viabilidad de soluciones que combinan captura de datos visuales, procesamiento mediante redes neuronales y validación de

resultados, siguiendo el ciclo completo que caracteriza a la ciencia de datos.

- Desde la dimensión social y educativa: este tipo de sistema puede transformar la práctica docente al automatizar tareas repetitivas de calificación, los profesores pueden dedicar más tiempo a la retroalimentación cualitativa y al acompañamiento pedagógico. Además, la estandarización del proceso reduce la subjetividad y los sesgos propios de la evaluación manual, promoviendo mayor equidad entre los estudiantes. En contextos como el ecuatoriano, donde un docente suele atender a muchos alumnos, herramientas de este tipo pueden mejorar significativamente la calidad del proceso de enseñanza – aprendizaje.

En conjunto; esta investigación se alinea con las tendencias globales de aplicación de inteligencia artificial (IA) en la educación y contribuye al desarrollo de un campo emergente con creciente demanda profesional y científica, en coherencia con las líneas prioritarias de la maestría.

Formulación de Preguntas

Pregunta Principal

«¿Cómo diseñar un sistema basado en deep learning (DL) y visión por computador (CV) que permita calificar automáticamente ejercicios matemáticos manuscritos, evaluando tanto el resultado final como la validez del procedimiento, con una precisión comparable a la calificación realizada por un docente humano?»

Como se puede apreciar esta pregunta aborda el núcleo del problema porque articula los tres elementos centrales del estudio: la herramienta tecnológica (deep learning y visión por computador), el objeto de evaluación (resultado y procedimiento) y el criterio de éxito (precisión comparable a la humana).

Preguntas Secundarias

- Pregunta secundaria 1: Arquitectura algorítmica: ¿Qué arquitectura de red neuronal (CNN, redes convolucionales – recurrentes o CRNN, modelos basados en Transformer o enfoques híbridos) ofrece el mejor rendimiento en el reconocimiento de símbolos y estructuras matemáticas manuscritas, en términos de exactitud,

F1-Score y tasa de error de caracteres (CER)?

- Pregunta secundaria 2: Procesamiento espacial: ¿Qué estrategias de procesamiento de imágenes y segmentación permiten interpretar correctamente la jerarquía espacial de expresiones matemáticas bidimensionales como: fracciones, exponentes, subíndices, radicales y también para preservar el significado semántico de la expresión?

- Pregunta secundaria 3: Validación del procedimiento: ¿En qué medida la integración de un módulo de validación del procedimiento (verificación paso a paso) mejora la calidad de la calificación automática frente a un enfoque que solo evalúa el resultado final?

Relevancia de las Preguntas Formuladas

Las preguntas planteadas son relevantes; principalmente porque cada una aborda una dimensión específica e indispensable del problema, siguiendo el principio de descomposición (Descartes, 1637). La pregunta principal define el rumbo general del estudio, mientras que las secundarias desglosan el problema en componentes manejables que pueden responderse de forma empírica mediante recolección y análisis de datos.

La pregunta principal establece el alcance integral del estudio y se alinea directamente con su objetivo central, que es diseñar un sistema funcional de calificación automática. Su formulación es clara, específica y viable, ya que puede responderse mediante el desarrollo experimental de un prototipo y su evaluación cuantitativa contra la calificación de un docente evaluador.

Por otra parte, las preguntas secundarias profundizan en componentes técnicos diferenciados como:

- La primera: atiende a la elección de la arquitectura algorítmica.

- La segunda: al procesamiento de la estructura bidimensional propia de la notación matemática

- Y la tercera: al aporte original del estudio, que consiste en validar el procedimiento y no solo el resultado final.

Resolución del Problema

Con el fin de hacer explícito cómo cada pregunta contribuye a resolver una brecha concreta del problema; se muestra en la Tabla 2 una matriz de correspondencia entre las brechas identificadas, las preguntas formuladas y la métrica o producto esperado para cada una.

Tabla 2

Correspondencia entre brechas del problema, preguntas y resultados esperados

Brecha del problema	Pregunta	Métrica
Inexistencia de un sistema integrado de calificación automática	Pregunta principal	Prototipo funcional (κ de Cohen, error medio).
Reconocimiento simbólico insuficiente para notación matemática.	Secundaria 1	Comparativa cuantitativa de arquitecturas mediante exactitud, F1-score, CER y ExpRate.
Pérdida de jerarquía espacial en expresiones bidimensionales.	Secundaria 2.	Pipeline de segmentación validado con el índice de solapamiento (IoU) y la exactitud estructural.
Imposibilidad de evaluar el procedimiento, solo el resultado.	Secundaria 3.	Módulo de validación paso a paso con métricas de mejora frente al modelo base.

Objetivos

Los objetivos que se presentan a continuación se derivan directamente de las preguntas formuladas en el apartado anterior y se redactan siguiendo los criterios

SMART; es decir, deben ser específicos, medibles, alcanzables, relevantes y acotados en el tiempo (Hernández Sampieri y Mendoza Torres, 2023). Cada objetivo se enuncia con un verbo en infinitivo que denota una acción verificable, e incorpora el instrumento con el que se ejecutará y el criterio con el que se comprobará su cumplimiento, de modo que ninguno quede formulado como una intención genérica imposible de contrastar con evidencia.

Objetivo General

Diseñar y evaluar un sistema de calificación automática de ejercicios matemáticos manuscritos, basado en técnicas de deep learning (DL) y visión por computador (CV), que valore tanto el resultado final como la validez del procedimiento intermedio, y que alcance sobre un conjunto de pruebas independientes un nivel de concordancia sustancial con la valoración emitida por un docente evaluador (κ de Cohen ≥ 0.75)

Objetivos Específicos

- **OE1. Comparar** el rendimiento de al menos tres arquitecturas de red neuronal (CNN, CRNN con función de pérdida CTC —*Connectionist Temporal Classification*— y modelos basados en Transformer) en el reconocimiento de símbolos y estructuras matemáticas manuscritas, empleando como métricas la exactitud, el F1-score, la tasa de error de caracteres (CER) y la tasa de reconocimiento exacto a nivel de expresión (ExpRate). De los tres corpus públicos identificados en la revisión —CROHME, HASYv2 y MathWriting— se selecciona CROHME como corpus experimental, por ser el único que dispone de particiones oficiales, de expresiones completas anotadas —condición necesaria para calcular el ExpRate— y de imágenes renderizadas en modalidad *offline*, lo que además preserva la comparabilidad directa con los resultados publicados.

- **OE2. Desarrollar** un pipeline de preprocesamiento y segmentación que preserve la jerarquía espacial bidimensional de las expresiones manuscritas —fracciones, exponentes, subíndices y radicales—, y verificar su desempeño mediante el IoU sobre las regiones detectadas y la exactitud estructural de la expresión

reconstruida en notación simbólica.

- **OE3. Implementar** un módulo de validación del procedimiento que compruebe la equivalencia matemática entre líneas consecutivas del desarrollo escrito, y evaluar su aporte frente a un modelo base que califica únicamente el resultado final, midiendo la concordancia con la calificación de un docente evaluador mediante el coeficiente κ de Cohen y el error absoluto medio sobre una escala binaria de dos (2) niveles.

Con el propósito de evidenciar la trazabilidad entre el problema, las preguntas y los objetivos, la Tabla 3 presenta la matriz de alineación correspondiente.

Tabla 3

Matriz de alineación entre objetivos, preguntas y productos verificables

Objetivo	Pregunta	Producto verificable y criterio de cumplimiento
OE1. Comparar arquitecturas	Secundaria 1	Tabla comparativa de exactitud, F1-score, CER y ExpRate para tres o más arquitecturas entrenadas bajo idéntico protocolo.
OE2. Desarrollar pipeline de segmentación	Secundaria 2	Pipeline documentado y reproducible, con IoU y exactitud estructural reportados sobre el conjunto de prueba.
OE3. Implementar y evaluar la validación del procedimiento	Secundaria 3	Módulo de verificación paso a paso, comparación estadística frente al modelo base que solo evalúa el resultado e informe de concordancia (κ de Cohen y error absoluto medio).

Como se aprecia en la Tabla 3, cada objetivo específico responde a una de las tres preguntas secundarias y genera un producto verificable, mientras que el objetivo general se satisface únicamente cuando los tres objetivos específicos han sido cumplidos.

Viabilidad del Estudio

Atendiendo a los criterios de viabilidad, el problema y las preguntas formuladas resultan factibles en cuanto a interés del investigador, acceso a datos, antecedentes y delimitación. De manera específica, la viabilidad se sustenta en los siguientes aspectos:

- Interés del tesista (mi persona): el tema responde a una motivación personal y profesional vinculada con la mejora de la práctica educativa mediante ciencia de datos.
- Acceso a datos: existen datasets públicos consolidados como MathWriting, CROHME y HASYv2 —de los cuales se adopta CROHME como corpus experimental—, complementados con la posibilidad de recolectar muestras propias en entornos educativos controlados.
- Antecedentes: existe literatura previa relevante sobre HTR, CNN, modelos Transformer y reconocimiento de expresiones matemáticas, lo cual permite construir un marco teórico sólido.
- Recursos computacionales: el entrenamiento de los modelos se realiza en una estación de trabajo local con aceleración por GPU, lo cual garantiza la disponibilidad continua del entorno, evita los límites de sesión propios de los servicios gratuitos en la nube y permite fijar la versión del entorno de ejecución durante todo el ciclo experimental.
- Delimitación clara: el estudio se restringe a ejercicios matemáticos de nivel medio, con notación estandarizada, lo que evita una dispersión metodológica.

Alcance y Limitaciones del Estudio

Delimitar con precisión hasta dónde llega la investigación y qué queda deliberadamente fuera de ella constituye una exigencia metodológica que protege la

validez de las conclusiones y evita que el lector atribuya al estudio alcances que este no reclama (Sabino, 1996). En coherencia con el problema planteado y con los recursos disponibles, el alcance del presente trabajo se define en los términos que resume la Tabla 4.

Tabla 4

Alcance y exclusiones del estudio

El estudio comprende	El estudio no comprende
Ejercicios de aritmética, álgebra elemental con notación estandarizada.	Demostraciones formales, razonamientos en prosa, geometría dibujada, gráficas y diagramas.
Imágenes de hojas manuscritas capturadas por escáner o cámara en condiciones controladas de iluminación y encuadre.	Reconocimiento en tiempo real sobre video, escritura sobre tableta digital (datos *online*) o fotografías en condiciones adversas.
Reconocimiento simbólico, reconstrucción estructural de la expresión y verificación de la equivalencia entre pasos consecutivos.	Evaluación de la calidad pedagógica de la redacción, del estilo de resolución o de la creatividad del método empleado.
Un prototipo funcional de investigación	Un producto de software en producción, con interfaz gráfica de usuario (GUI) y/o despliegue institucional con soporte operativo.

De este alcance se desprenden las limitaciones que se enuncian a continuación, las cuales se declaran de manera anticipada para que sean consideradas en la interpretación de los resultados:

- Representatividad de la muestra: la muestra propia será resuelta por escritores adultos voluntarios, decisión que evita el tratamiento de datos de menores de edad

pero implica que los resultados no son directamente extrapolables a la caligrafía de estudiantes adolescentes ni generalizables a la totalidad del sistema educativo ecuatoriano.

- Sesgo caligráfico del corpus público: CROHME fue construido mayoritariamente con escritura de poblaciones no ecuatorianas, lo que puede afectar el desempeño del modelo ante rasgos caligráficos locales y obliga a interpretar con cautela las métricas obtenidas sobre este conjunto.

- Uso de un único corpus experimental: la selección de CROHME como corpus de entrenamiento y evaluación garantiza la comparabilidad con la literatura, pero impide contrastar el efecto del corpus sobre el desempeño; los resultados no son, por tanto, directamente transferibles a HASYv2 ni a MathWriting sin una validación específica.

- Restricciones de cómputo: el entrenamiento se ejecuta sobre una GPU local de 4 GB de memoria dedicada, lo que acota el tamaño de los modelos entrenables, obliga a emplear lotes reducidos junto con precisión mixta y acumulación de gradiente, y limita el número de épocas y de experimentos de ajuste de hiperparámetros.

- Variabilidad por inicialización: dada la restricción anterior, cada configuración se entrena con una única semilla declarada en lugar de repetirse con varias, de modo que no se reporta la desviación estándar entre inicializaciones. La significancia de las diferencias entre arquitecturas se sustenta en pruebas pareadas sobre el mismo conjunto de prueba, que no requieren repeticiones, pero las diferencias de magnitud reducida deben interpretarse con cautela.

- Cobertura del módulo de validación: La verificación de equivalencia entre pasos se restringe a las transformaciones algebraicas que el motor simbólico empleado sea capaz de resolver; procedimientos válidos pero expresados de forma no canónica podrían ser señalados como incorrectos.

- Ausencia de evaluación longitudinal: El estudio mide la concordancia con la calificación docente en un corte temporal único y no evalúa el efecto del sistema sobre el aprendizaje de los estudiantes ni su adopción sostenida por parte del profesorado.

Capítulo II: Marco Teórico y Revisión Literaria

Introducción del Capítulo

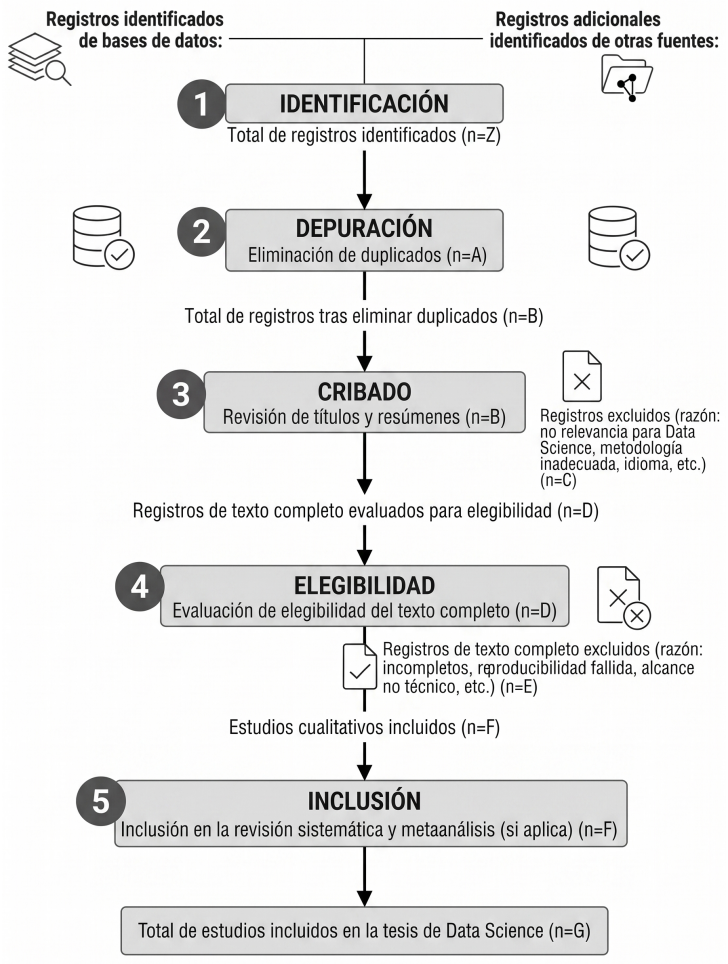
El presente capítulo desarrolla el marco teórico y la revisión de la literatura del proyecto de investigación. Su propósito es demostrar que la investigación se fundamenta en conocimiento académico pertinente, actualizado y críticamente analizado y que ese conocimiento sostiene de manera explícita las decisiones metodológicas posteriores. En coherencia con la orientación del trabajo de investigación; el capítulo articula conceptos, teorías, antecedentes, variables, métodos y brechas que dan soporte al problema y a los objetivos formulados en el Capítulo I.

La revisión se concibe como el puente entre el problema identificado —la ausencia de sistemas capaces de calificar ejercicios matemáticos manuscritos evaluando tanto el resultado final como la validez del procedimiento— y la fase experimental del estudio. De este modo, este capítulo responde a tres (3) preguntas guía: «¿qué conceptos deben definirse?», «¿qué teorías y modelos explican el fenómeno del reconocimiento de expresiones matemáticas manuscritas?», y «¿qué evidencia empírica respalda o limita las alternativas metodológicas disponibles?». El hilo conductor es la coherencia con la pregunta principal de investigación y con los tres objetivos específicos: la comparación de arquitecturas de reconocimiento (OE1), la preservación de la jerarquía espacial bidimensional mediante segmentación (OE2) y la validación del procedimiento paso a paso contrastada con el juicio docente (OE3).

Tras describir la estrategia de revisión de literatura bajo una adaptación del protocolo PRISMA como se muestra en la Figura 1, se analizan críticamente los antecedentes organizados por temas, se definen las bases conceptuales y los enfoques teóricos que sustentan el problema, se derivan las variables y categorías de análisis útiles para la metodología, se sintetiza la brecha de investigación y se cierra explicando cómo la literatura sostiene el método y el aporte esperado. En conjunto, este capítulo busca expresar por qué la investigación es necesaria, qué conocimiento la sustenta y qué decisiones metodológicas se derivan de la literatura, en el marco de una investigación aplicada de enfoque cuantitativo propia de la ciencia de datos.

Figura 1

Diagrama de flujo PRISMA (Preferred Reporting Items for Systematic Reviews and Meta – Analyses) en Ciencia de Datos



Nota. La metodología PRISMA actúa como un embudo de calidad.

Estrategia de Revisión de la Literatura

La revisión adopta un carácter semi – sistemático con criterios explícitos y se apoya en una adaptación del protocolo PRISMA 2020 (Page et al., 2021); para reportar de forma transparente cómo se identificaron, depuraron, cribaron y seleccionaron los estudios. PRISMA no constituye en sí mismo un marco teórico, sino una guía de reporte que aporta trazabilidad y reproducibilidad a la búsqueda; su uso como estrategia metodológica permite que toda decisión de inclusión sea verificable. La pregunta de revisión, las cadenas de búsqueda y los criterios de inclusión y exclusión se formularon antes de seleccionar los estudios, evitando así el sesgo de selección posterior al

conocimiento de los resultados.

La pregunta que orienta la revisión se formula del siguiente modo: «¿qué arquitecturas de deep learning (DL) y visión por computador (CV), publicadas entre 2017 y 2026, permiten reconocer y calificar automáticamente ejercicios matemáticos manuscritos evaluando tanto el resultado final como la validez del procedimiento, y qué evidencia empírica reportan en términos de exactitud, tasa de exactitud a nivel de expresión (*Expression Recognition Rate*, ExpRate), tasa de error de caracteres (CER), índice de solapamiento (*Intersection over Union*, IoU) y concordancia con evaluadores humanos?». La ventana temporal se ancla en 2017 porque ese año marca la consolidación del paradigma encoder – decoder con atención en el reconocimiento de expresiones matemáticas manuscritas a partir del modelo *Watch, Attend and Parse* (Zhang et al., 2017).

Bases de Datos y Fuentes Académicas

Se consultaron bases multidisciplinarias y especializadas en computación, ingeniería e inteligencia artificial (IA), complementadas con repositorios de preprints y un buscador académico general. La combinación de fuentes especializadas y generales mejora la cobertura y reduce el riesgo de omitir producción relevante, incluida la regional e iberoamericana. La Tabla 5 detalla las fuentes empleadas y la justificación de su uso.

Tabla 5

Bases de datos y fuentes académicas consultadas

Fuente	Justificación	Fecha de búsqueda
Scopus	Base multidisciplinaria con artículos revisados por pares y análisis de citas e impacto.	Junio 2026
Web of Science	Permite rastrear impacto, citas y producción científica de alto alcance.	Junio 2026
IEEE Xplore	Fuente central para ingeniería, computación, IA, visión por computador y Data Science.	Junio 2026
ACM Digital Library	Relevante para computación, algoritmos, sistemas y reconocimiento de patrones.	Junio 2026
Springer Link	Acceso a actas de conferencias (ICDAR, ICFHR) y revistas	Junio 2026
arXiv	Preprints recientes de modelos de vanguardia aún no indexados en revistas.	Junio 2026
Google Scholar	Complemento de cobertura, literatura gris y tesis iberoamericanas.	Junio 2026
PUCE	Base multidisciplinaria de la Pontificia Universidad Católica del Ecuador	Junio 2026

Cadenas de Búsqueda

Las cadenas se construyeron a partir de cuatro bloques temáticos —fenómeno, método, aplicación y datos— combinados con operadores booleanos. Se trabajó con versiones en inglés y en español para captar tanto la producción internacional como la regional, según se resume en la Tabla 6.

Tabla 6

Bloques temáticos y términos empleados en las cadenas de búsqueda

Bloque	Términos en inglés	Términos en español
Fenómeno	«handwritten mathematical expression recognition» OR HMER OR «handwritten math»	«reconocimiento de expresiones matemáticas manuscritas» OR «matemáticas manuscritas»
Método	«deep learning» OR CNN OR CRNN OR transformer OR «encoder decoder» OR «computer vision»	«aprendizaje profundo» OR «redes neuronales convolucionales» OR transformer OR «visión por computador»
Aplicación	«automatic grading» OR autograding OR «automated assessment» OR «step by step»	«calificación automática» OR «evaluación automática» OR «validación paso a paso»
Datos y contexto	CROHME OR MathWriting OR HASYv2 OR «math education»	CROHME OR MathWriting OR «educación matemática»

De los cuales se tiene: Cadena final en inglés: («handwritten mathematical expression recognition» OR HMER) AND («deep learning» OR transformer OR «computer vision») AND («automatic grading» OR «step by step») AND (CROHME OR MathWriting OR HASYv2).

Criterios de Inclusión y Exclusión

Conforme a la regla metodológica de la revisión, los criterios se definieron antes de seleccionar los estudios. De este modo se garantiza la trazabilidad y se evita ajustar los criterios a los resultados ya conocidos. La Tabla 7 presenta los cinco criterios aplicados con sus condiciones de inclusión y exclusión.

Tabla 7

Criterios de inclusión y exclusión de la revisión

Criterio	Incluir	Excluir
Temporalidad	Estudios de 2017 a 2026, salvo referencias seminales.	Estudios anteriores sin valor fundacional o desactualizados sin justificación.
Pertinencia	Reconocimiento o calificación de expresiones matemáticas manuscritas.	HTR de texto convencional, OCR impreso o fórmulas únicamente tipográficas.
Método	Deep learning o visión por computador (CNN, CRNN, Transformer, GNN, VLM).	Métodos basados solo en reglas o gramáticas manuales sin aprendizaje.
Calidad	Revistas indexadas, conferencias revisadas por pares ó preprints con impacto.	Opiniones, blogs o documentos sin método verificable.
Aplicabilidad	Estudios con métricas cuantitativas (ExpRate, CER, F1-score, IoU, κ, MAE).	Estudios sin métricas o sin descripción reproducible del experimento.

De los cuales se tiene: GNN = redes neuronales de grafos; VLM = modelos de visión y lenguaje; MAE = error absoluto medio.

Registro Tipo PRISMA

El flujo de identificación, depuración, cribado, elegibilidad e inclusión se resume en la Tabla 8 y se ilustra en la Figura 2. Las cantidades reportadas corresponden al proceso de revisión ejecutado para este capítulo y constituyen evidencia de transparencia metodológica.

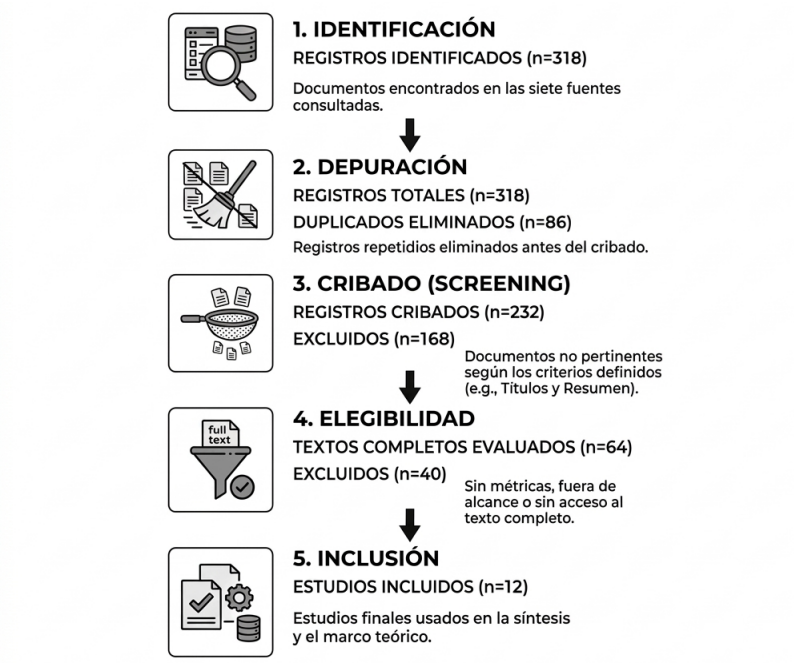
Tabla 8

Registro tipo PRISMA del proceso de revisión

Etapa	Cantidad	Descripción
Registros identificados	318	Documentos encontrados.
Duplicados eliminados	86	Registros repetidos y eliminados.
Registros cribados	232	Documentos revisados por título y resumen.
Excluidos en cribado	168	Documentos no pertinentes.
Textos completos evaluados	64	Documentos revisados a texto completo para elegibilidad.
Excluidos en texto completo	40	Sin métricas, fuera de alcance o sin acceso al texto completo.
Estudios incluidos	12	Estudios finales usados en la síntesis y el marco teórico.

Figura 2

Diagrama de flujo PRISMA adaptado al proyecto



Nota. Elaboración propia con base en el protocolo PRISMA

Antecedentes de Investigación

Los antecedentes se organizan por temas y no como una lista aislada de autores. Se parte de la evolución general del reconocimiento de expresiones matemáticas manuscritas, se avanza hacia la consolidación de las arquitecturas más recientes y los enfoques estructurales, se examina la transición desde el reconocimiento hacia la calificación con validación del procedimiento, y se cierra con el papel de los conjuntos de datos públicos y las limitaciones que de ellos se derivan. Cada bloque compara estudios, identifica coincidencias y diferencias, y subraya sus limitaciones y aportes.

Modelos Encoder – Decoder

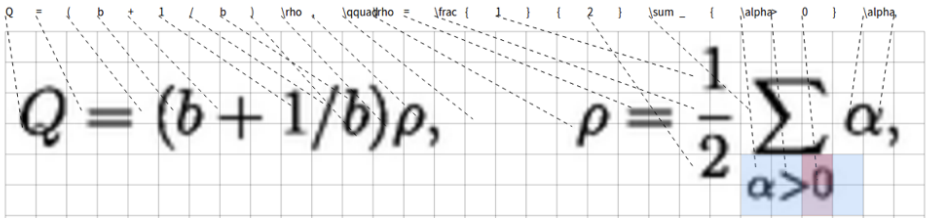
El reconocimiento de expresiones matemáticas manuscritas (*Handwritten Mathematical Expression Recognition*, HMER) fue durante décadas un problema abordado mediante segmentación explícita y gramáticas bidimensionales construidas a mano; Zanibbi y Blostein (2012) sistematizaron ese período y mostraron que el reconocimiento de notación matemática exige resolver simultáneamente tres subtareas acopladas —reconocimiento del símbolo, segmentación y análisis estructural—, lo que limitaba la robustez de los enfoques basados en reglas. El cambio de paradigma llegó con los modelos de aprendizaje profundo de extremo a extremo. Zhang et al. (2017) propusieron el modelo *Watch, Attend and Parse* (WAP), que reformula el problema como una traducción de imagen a secuencia: un codificador convolucional observa la imagen y un decodificador recurrente, dotado de un mecanismo de atención basado en cobertura (*coverage*), genera directamente la cadena LaTeX, eliminando la segmentación explícita. WAP alcanzó tasas de exactitud a nivel de expresión (ExpRate) de 46.55 % y 44.55 % en los conjuntos CROHME 2014 y 2016, superando a los métodos previos basados en gramáticas.

De forma paralela, Deng, Kanervisto, Ling, y Rush (2017) demostraron que la generación de marcado a partir de imágenes podía resolverse con un modelo de atención de grano grueso a fino (coarse to fine) para convertir imágenes de fórmulas en código LaTeX (Ver Figura 3); superando las limitaciones del OCR tradicional en estructuras bidimensionales. Además, introdujeron el dataset «Im2LaTeX – 100k», logrando así un

rendimiento superior en la generación de marcado estructurado a partir de imágenes. Sobre esa base, Zhang, Du, y Dai (2018) introdujeron una atención multiescala con codificador denso (DenseWAP), que mejora el reconocimiento de símbolos de distinto tamaño —frecuentes en exponentes y subíndices— al combinar características de varias resoluciones. Por tanto, podemos observar que la coincidencia entre estos trabajos es clara y el aprendizaje de la correspondencia entre regiones de la imagen y símbolos de salida mediante atención resulta más robusto que la segmentación manual. Su limitación compartida es que el decodificador recurrente se degrada en secuencias largas y que la atención puede desalinearse en expresiones estructuralmente complejas.

Figura 3

Example of the model generating mathematical markup



Nota. Image to Markup Generation with Coarse to Fine Attention (Deng et al., 2017).

Consolidación del Transformer

La incorporación del mecanismo de atención del Transformer (Vaswani et al., 2017) marcó una segunda etapa. Zhao et al. (2021) presentaron BTTR (*Bidirectionally Trained Transformer*), que sustituye el decodificador recurrente por un decodificador Transformer entrenado de forma bidireccional —de izquierda a derecha y de derecha a izquierda—, y reportó mejoras de ExpRate de 2.23 %, 1.92 % y 2.28 % sobre el estado del arte en CROHME 2014, 2016 y 2019, sin aumento de datos. Esta línea fue refinada por Zhao y Gao (2022) con CoMER, que reintroduce el modelado de cobertura dentro del Transformer para mitigar la sobre atención y la falta de cobertura de regiones, y por Bian et al. (2022) con un esquema de aprendizaje mutuo bidireccional (ABM) que aprovecha la información de ambas direcciones de decodificación.

En paralelo, los enfoques estructurales sostienen que reconocer la notación matemática es incompleto si no se preserva su jerarquía bidimensional. Yuan et al.

(2022) propusieron una red consciente de la sintaxis (*Syntax-Aware Network*, SAN) que modela la expresión como un árbol de derivación, donde cada nodo es un símbolo y cada arista una relación espacial —encima, debajo, dentro, a la derecha—, mejorando la consistencia estructural de la salida. Otra vertiente, ilustrada por Wu, Yin, Zhang, Zhang, y Liu (2020) con el aprendizaje adversarial emparejado (*Paired Adversarial Learning*, PAL), aborda la variabilidad de la escritura manuscrita generando representaciones invariantes al estilo del escritor. La revisión sistemática de Xie et al. (2024) integra estas corrientes y clasifica las arquitecturas en encoder – decoder, basadas en Transformer y redes neuronales de grafos (GNN), documentando el avance del campo hasta las competencias CROHME más recientes. La diferencia esencial entre las dos corrientes es de énfasis: la primera optimiza la transducción secuencial, mientras la segunda prioriza la fidelidad estructural; ambas, sin embargo, comparten la limitación de centrarse en el reconocimiento y no en la evaluación educativa del procedimiento.

La línea más reciente incorpora explícitamente la información de posición y estructura. Guan, Lin, Shen, y Yang (2024) propusieron PosFormer (*Position Forest Transformer*), que optimiza conjuntamente dos tareas para aprender representaciones de símbolos conscientes de su ubicación jerárquica, modelando la expresión como un bosque de posiciones. PosFormer superó de forma consistente a los modelos previos en las series CROHME y demostró robustez particular en expresiones complejas y anidadas. Esta evidencia respalda la hipótesis de trabajo del proyecto, según la cual los modelos basados en Transformer —en especial aquellos que integran información estructural— ofrecen el mejor compromiso entre exactitud y error para el reconocimiento simbólico (OE1). La Tabla 9 resume esta evolución sobre las particiones de CROHME.

Tabla 9

Evolución comparativa de arquitecturas de reconocimiento sobre CROHME (ExpRate, %)

Modelo (año)	CROHME 14	CROHME 16	CROHME 19	Contribución principal
WAP (2017)	≈ 46.6	—	—	Imagen – a – secuencia con atención y cobertura; sin segmentación explícita.
BTTR (2021)	≈ 54.0	—	—	Decodificador Transformer con entrenamiento bidireccional.
CoMER (2022)	59.33	59.81	62.97	Cobertura en el Transformer mediante el módulo ARM.
PosFormer (2024)	SOTA	SOTA	SOTA	Aprendizaje conjunto de expresión y posición jerárquica.

Se tiene lo siguiente: ExpRate = tasa de reconocimiento exacto a nivel de expresión. SOTA indica que el modelo superó el estado del arte previo en los tres conjuntos; los valores absolutos varían según el uso de aumento de datos.

Validación del Procedimiento

La transición del reconocimiento hacia la calificación automática constituye el núcleo del aporte original del proyecto. Chaowicharat y Dejdumrong (2023) presentaron un sistema que acopla el reconocimiento de expresiones matemáticas (*Mathematical Expression Recognition*, MER) con un sistema de álgebra computacional (*Computer Álgebra System*, CAS) para verificar la corrección de cada paso de una derivación manuscrita; su propuesta es funcional, pero advierte que el solo uso de MER y CAS resulta frágil ante errores de reconocimiento, por lo que requiere módulos de corrección. La irrupción de los grandes modelos de lenguaje (LLMs) y de visión abrió una vía alternativa, que compara la solución del estudiante contra una de referencia o califica directamente desde la imagen, evitando la etapa explícita de reconocimiento: Liu et al. (2026) evaluaron modelos de tipo GPT-4 en exámenes universitarios de matemáticas manuscritos. Por su parte, Parsaeifard, Hlosta, y Bergamin (2025) compararon modelos de meta–aprendizaje con modelos de visión y lenguaje de gran escala (VLLM) para la

calificación automática de respuestas manuscritas, y encontraron que el meta–aprendizaje supera a los VLLM en clasificación binaria, advirtiendo que la fiabilidad práctica de estos últimos todavía requiere mayor investigación por su sensibilidad a la calidad de la escritura.

Esta línea se sitúa en la intersección con la analítica del aprendizaje (learning analytics), que sostiene que una decisión automática solo es legítima en contextos educativos si es concordante con el juicio experto y además, interpretable (Siemens y Long, 2011). De ahí que la concordancia con docentes humanos —medida con el coeficiente κ de Cohen (Cohen, 1960) e interpretada según las bandas de Landis y Koch (1977)— se convierta en el criterio de validez central para un sistema de calificación automática. El contraste con los estudios de la sección anterior es revelador: mientras la mayoría de los trabajos optimizan el reconocimiento, muy pocos abordan la validación del procedimiento en escenarios reales de calificación, y los que lo hacen reconocen limitaciones de robustez.

Conjuntos de Datos Públicos y Benchmarks

La comparabilidad de los resultados depende de conjuntos de datos públicos consolidados. Las competencias CROHME, sistematizadas por Mahdavi, Zanibbi, Mouchère, Viard-Gaudin, y Garain (2019), establecieron el estándar de evaluación con sus particiones oficiales de entrenamiento y prueba —2014, 2016 y 2019— y las métricas ExpRate y CER. A este referente se suma MathWriting (Gervais, Fadeeva, y Maksai, 2025), el mayor conjunto de expresiones matemáticas manuscritas en línea publicado hasta la fecha (2026), con 230 000 muestras humanas y 400 000 sintéticas, rasterizables para reconocimiento fuera de línea. Para el reconocimiento de símbolos aislados, HASYv2 (Thoma, 2017) aporta más de 150 000 instancias de 369 clases de símbolos matemáticos, útil para pre–entrenar el clasificador a nivel de símbolo.

La relevancia de estos recursos para el proyecto es doble. Por un lado, permiten entrenar y validar las arquitecturas sobre referencias comparables sin depender de la recolección primaria de datos en esa fase, lo que ancla el estudio en conjuntos públicos consolidados —MathWriting, CROHME y HASYv2— y simplifica las consideraciones

éticas. Por otro lado, presentan una limitación que el proyecto debe reconocer: estos conjuntos contienen mayoritariamente expresiones aisladas y no procedimientos multipaso completos, lo cual condiciona la evaluación del módulo de validación del procedimiento (OE3), *siendo este un punto de partida para investigaciones en tesis Doctorales.*

Las Tablas 10, 11 y 12, 13 sintetizan los doce (12) antecedentes seleccionados en la revisión: la primera reúne el contexto y el método de cada estudio, y la segunda sus hallazgos, limitaciones y aporte al proyecto de investigación.

Tabla 10

Matriz de extracción de la literatura: contexto y método de los estudios seleccionados (Parte 1 de 2)

ID	Autor, año y título abreviado	Contexto y datos	Método o modelo
E1	Zhang et al. (2017). *Watch, Attend and Parse* (WAP)	CROHME 2014 y 2016	Codificador CNN y decodificador RNN con atención por cobertura; salida LaTeX de extremo a extremo.
E2	Deng et al. (2017). Imagen – a – marcado con atención de grano grueso a fino	Fórmulas impresas y manuscritas	Encoder – decoder con atención jerárquica de grano grueso a fino.
E3	Zhang et al. (2018). DenseWAP (atención multiescala)	CROHME	Codificador denso y atención multiescala para símbolos de distinto tamaño.
E4	Wu et al. (2020). *Paired Adversarial Learning* (PAL)	CROHME	Aprendizaje adversarial emparejado para lograr invariancia al estilo del escritor.
E5	Zhao et al. (2021). BTTR (Transformer bidireccional)	CROHME 2014, 2016 y 2019	Codificador CNN y decodificador Transformer entrenado en ambas direcciones, sin aumento de datos.
E6	Bian et al. (2022). ABM (aprendizaje mutuo bidireccional)	CROHME	Aprendizaje mutuo entre las decodificaciones de izquierda a derecha y de derecha a izquierda.

Tabla 11

Matriz de extracción de la literatura: contexto y método de los estudios seleccionados (Parte 2 de 2)

ID	Autor, año y título abreviado	Contexto y datos	Método o modelo
E7	Yuan et al. (2022). SAN (red consciente de la sintaxis)	CROHME y expresiones manuscritas reales	Modela la expresión como árbol de derivación con relaciones espaciales explícitas.
E8	Zhao y Gao (2022). CoMER (cobertura en Transformer)	CROHME	Reintroduce el modelado de cobertura dentro del decodificador Transformer.
E9	Chaowicharat y Dejdumrong (2023). MER y CAS para calificación paso a paso	Tareas reales de matemáticas	Reconocimiento de expresiones acoplado a un sistema de álgebra computacional.
E10	Xie et al. (2024). Revisión de HMER (encoder – decoder y GNN)	Revisión del campo hasta 2024	Clasifica las arquitecturas en encoder – decoder, Transformer y GNN.
E11	Gervais et al. (2025). MathWriting (conjunto de datos)	230 000 muestras humanas y 400 000 sintéticas	Conjunto en línea, rasterizable para reconocimiento fuera de línea.
E12	Parsaeifard et al. (2025). Meta-aprendizaje frente a VLLM	Datos reales de contextos educativos	Comparación de meta-aprendizaje frente a modelos de visión y lenguaje de gran escala.

La elaboración propia se realizo a partir de la revisión semisistemática. La continuación de esta matriz se presenta en las Tablas 12, 13 respectivamente.

Tabla 12

Matriz de análisis de la literatura: hallazgos, limitaciones y aporte a la tesis (Parte 1 de 2)

ID	Hallazgos principales	Limitaciones	Aporte a la tesis
E1	ExpRate de 46.55 % (2014) y 44.55 % (2016); elimina la segmentación explícita.	El decodificador recurrente se degrada en secuencias largas; atención imperfecta.	Define el paradigma base del reconocimiento (OE1).
E2	Generaliza la transducción imagen – a – marcado más allá de lo manuscrito.	Menor foco en notación manuscrita real y su variabilidad.	Respalda la generalidad del enfoque encoder – decoder.
E3	Mejora el reconocimiento de exponentes y subíndices pequeños.	Hereda el decodificador recurrente y su coste secuencial.	Sustenta la preservación de escala en la segmentación (OE2).
E4	Mayor robustez ante la variabilidad de la escritura manuscrita.	Mayor complejidad de entrenamiento; no valida el procedimiento.	Justifica el aumento y la normalización de datos.
E5	Mejora la ExpRate en 2.23 %, 1.92 % y 2.28 % sobre el estado del arte.	Coste computacional alto; sin validación del procedimiento.	Orienta la elección del decodificador Transformer (OE1).
E6	Aprovecha ambas direcciones de decodificación para mejorar la exactitud.	Solo reconocimiento; sin foco educativo.	Refuerza la pertinencia del decodificador bidireccional.
E7	Mejora la consistencia estructural de la salida.	Mayor complejidad; centrada en el reconocimiento.	Fundamenta el enfoque estructural del OE2.

La elaboración propia se realizo a partir de la revisión semisistemática. ExpRate = exactitud a nivel de expresión; VLLM = modelos de visión y lenguaje de gran escala; OE = objetivo específico. La identificación de cada estudio corresponde a las Tablas 10, 11 respectivamente.

Tabla 13

Matriz de análisis de la literatura: hallazgos, limitaciones y aporte a la tesis (Parte 2 de 2)

ID	Hallazgos principales	Limitaciones	Aporte a la tesis
E8	Reduce la sobre-atención y mejora la cobertura de regiones.	No aborda la calificación ni el procedimiento.	Aporta mecanismos de atención robustos (OE1 y OE2).
E9	Propuesta funcional de calificación de derivaciones paso a paso.	Frágil ante errores de reconocimiento; muestras limitadas.	Antecedente directo del módulo de validación (OE3).
E10	Sistematiza métricas, conjuntos de datos y arquitecturas.	Identifica la falta de evaluación en calificación educativa real.	Mapa del campo y soporte de la brecha de investigación.
E11	Mayor conjunto de expresiones manuscritas publicado hasta la fecha.	Mayoría de expresiones aisladas, no procedimientos multipaso.	Fuente principal de datos del proyecto.
E12	El meta-aprendizaje supera a los VLLM en clasificación binaria.	La fiabilidad práctica de los VLLM aún no está garantizada.	Justifica el contraste contra una línea base (OE3).

La lectura comparada de la matriz evidencia un patrón consistente: la mayoría de los trabajos (E1 – E8 y E10) optimizan el reconocimiento simbólico y estructural, pero no integran la validación del procedimiento en escenarios reales de calificación; los que sí lo abordan (E9 y E12) reconocen limitaciones de robustez y de tamaño muestral. Esta tensión entre el progreso del reconocimiento y la escasa evaluación de la calificación con validación del procedimiento es precisamente el espacio que el proyecto de investigación se propone ocupar con ciertas *limitaciones propias del nivel de posgrado.*

Bases Conceptuales

Esta sección define los conceptos centrales del estudio y cada definición se sustenta en autores de referencia; se integra en una formulación propia y se vincula explícitamente con su uso en la investigación, de modo que el marco conceptual no

permanezca en el plano abstracto, sino que sostenga la operacionalización metodológica posterior. Los conceptos se presentan en un orden que va de lo específico del dominio a sus fundamentos y finalmente al razonamiento matemático que da sentido a la validación del procedimiento.

Expresiones Matemáticas Manuscritas

El reconocimiento de expresiones matemáticas manuscritas (*Handwritten Mathematical Expression Recognition*, HMER) es el proceso que transforma una imagen o una secuencia de trazos en una representación simbólica estructurada, habitualmente una cadena LaTeX o un árbol MathML. A diferencia del reconocimiento de texto, el HMER debe resolver simultáneamente tres subproblemas acoplados: el reconocimiento de los símbolos individuales, su segmentación implícita o explícita, y el análisis estructural que recupera la disposición bidimensional (Xie et al., 2024). Según la modalidad de captura, se distingue el HMER en línea (*online*), que opera sobre la trayectoria temporal de los trazos capturada por dispositivos táctiles, y el HMER fuera de línea (*offline*), que parte de imágenes de píxeles; este proyecto se sitúa principalmente en la modalidad fuera de línea, dado que parte de imágenes renderizadas de ejercicios manuscritos.

En el estudio, el HMER se operacionaliza mediante dos métricas: la tasa de reconocimiento exacto a nivel de expresión (ExpRate), que exige coincidencia perfecta de toda la fórmula, y la tasa de error de caracteres (CER), más granular y tolerante. La distinción es relevante: una expresión puede tener un CER bajo y, aun así, no ser exacta a nivel de expresión, por lo que ambas métricas se reportan de manera conjunta.

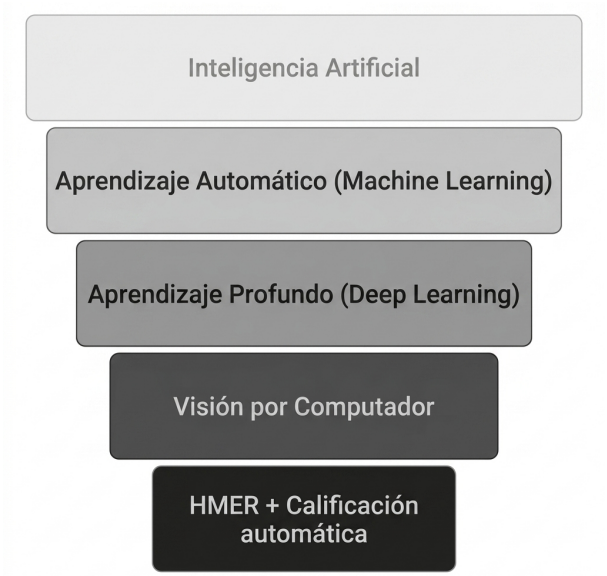
Visión por Computador y Reconocimiento de Imágenes

La visión por computador es la disciplina que dota a las máquinas de la capacidad de interpretar el contenido visual del mundo a partir de imágenes o vídeos. Sus tareas fundamentales incluyen la clasificación, la detección de objetos y la segmentación. El reconocimiento de imágenes, en sentido amplio, abarca el proceso de extraer de una imagen una descripción semántica útil, y constituye la tarea sobre la que opera el sistema propuesto.

La relación entre esta disciplina y el aprendizaje profundo es jerárquica: la visión por computador define el problema, mientras que el aprendizaje profundo provee el motor que aprende a resolverlo a partir de ejemplos. Dicho de otro modo, el reconocimiento de imágenes es la tarea y el deep learning es el método que, mediante el ajuste iterativo de millones de parámetros sobre grandes volúmenes de datos, aproxima la función que asigna a cada imagen su representación simbólica. La Figura 4 ilustra esta relación de inclusión, desde la inteligencia artificial hasta el problema concreto de estudio.

Figura 4

Relación jerárquica entre deep learning, visión por computador y el problema de estudio



Nota. El reconocimiento de imágenes es la tarea, el deep learning es el motor que aprende a resolverla.

Históricamente, el reconocimiento de imágenes dependía de descriptores diseñados manualmente. El punto de inflexión se produjo cuando las redes neuronales convolucionales profundas demostraron, sobre el conjunto ImageNet, una reducción drástica del error de clasificación respecto de los métodos previos (Krizhevsky, Sutskever, y Hinton, 2012). Desde entonces, la visión por computador adoptó masivamente el aprendizaje profundo, lo que explica por qué el sistema propuesto se construye sobre redes neuronales y no sobre reglas heurísticas.

Deep Learning (DL)

El aprendizaje profundo es la rama del aprendizaje automático que emplea redes neuronales con múltiples capas para aprender representaciones jerárquicas de los datos. Cada capa transforma su entrada en una representación ligeramente más abstracta: en visión, las primeras capas detectan bordes y texturas, las intermedias combinan estos elementos en partes, y las capas profundas codifican conceptos completos (Goodfellow, Bengio, y Courville, 2016; LeCun, Bengio, y Hinton, 2015). Esta capacidad de aprender automáticamente las características relevantes es lo que confiere al aprendizaje profundo su potencia para el reconocimiento de imágenes.

En el estudio, el deep learning sustenta tanto la codificación visual de la imagen del ejercicio como la decodificación simbólica que produce la expresión reconocida. Conceptualmente, basta aquí con establecer que una red profunda es una función parametrizada cuyos parámetros se ajustan minimizando una medida de error sobre datos etiquetados, de manera que, tras el entrenamiento, generalice a ejemplos no vistos. Las familias de arquitecturas relevantes y el mecanismo de entrenamiento que permite ese ajuste se desarrollan en detalle en la sección de teorías y enfoques.

Segmentación

La segmentación es el proceso de particionar una imagen en regiones que corresponden a entidades significativas preservando su posición relativa. En el dominio de las expresiones matemáticas, la segmentación no es un fin en sí mismo, sino el medio para recuperar la estructura bidimensional: la ubicación de un símbolo respecto de sus vecinos determina si actúa como exponente, subíndice, numerador o denominador. Por ello, una segmentación que ignora la jerarquía espacial pierde precisamente la información que distingue, por ejemplo, una potencia de un producto.

Preservación de la Jerarquía Espacial

La preservación de la jerarquía espacial se evalúa mediante el índice de solapamiento (IoU), que cuantifica la coincidencia entre la región predicha y la región de referencia y mediante una medida de exactitud estructural que verifica que el orden de lectura horizontal (izquierda–derecha) de las regiones detectadas reproduzca el de la

referencia. Estas métricas operacionalizan el objetivo OE2 y permiten contrastar empíricamente la hipótesis de que un pipeline consciente de la estructura supera a uno que la ignora.

Razonamiento Matemático y Métodos de Resolución

La calificación de un ejercicio matemático no se agota en verificar el resultado final; requiere juzgar la validez del procedimiento, es decir, si cada paso de la resolución se deriva legítimamente del anterior. Este concepto, central para el objetivo OE3, exige introducir en el marco teórico los fundamentos del razonamiento matemático y de los métodos de resolución. Resolver un ejercicio de álgebra elemental consiste en aplicar una sucesión de transformaciones que preservan la equivalencia: sumar o restar el mismo término en ambos miembros, multiplicar o dividir por una cantidad no nula, o sustituir una expresión por otra equivalente. La validez del procedimiento equivale, así, a la conservación de la equivalencia a lo largo de la cadena de pasos.

Desde la perspectiva de la lógica matemática, una demostración o una resolución es una secuencia de proposiciones encadenadas que parten de una hipótesis y, mediante reglas de inferencia válidas, conducen a una tesis. Pólya (1945), en su clásico tratado sobre la heurística de la resolución de problemas, sistematizó este proceso en fases que siguen vigentes como modelo del razonamiento matemático. Velleman (2019) y Hammack (2018) formalizan, por su parte, los métodos de demostración que estructuran ese razonamiento y que se resumen en la Tabla 14.

Tabla 14

Métodos de demostración y resolución relevantes para la validación del procedimiento

Método	Estructura lógica	Relación con la validación automática
Método directo	De la hipótesis P se deduce la tesis Q mediante implicaciones sucesivas ($P \Rightarrow \cdots \Rightarrow Q$).	Cada línea de la derivación debe ser equivalente o consecuencia válida de la anterior.
Contrarrecíproco	Para probar $P \Rightarrow Q$ se demuestra $\neg Q \Rightarrow \neg P$, lógicamente equivalente.	Permite verificar pasos por su forma contrapuesta cuando la directa es opaca.
Reducción al absurdo	Se supone $\neg Q$ y se deriva una contradicción, concluyendo Q.	Detecta procedimientos inválidos cuando un paso conduce a una inconsistencia.
Inducción matemática	Se prueba el caso base y el paso inductivo $P(n) \Rightarrow P(n+1)$.	Aplica a ejercicios con estructura recursiva o sumatorias.
Manipulación algebraica equivalente	Transformaciones que preservan el conjunto solución de una ecuación.	Núcleo de la verificación mediante CAS: dos líneas son válidas si son equivalentes.

Se tiene la síntesis a partir de Pólya (1945), Velleman (2019) y Hammack (2018). La columna derecha conecta cada método con el módulo de validación paso a paso del objetivo OE3.

Esta base es la que permite que un sistema de álgebra computacional verifique automáticamente un procedimiento: al representar cada línea como un árbol de expresión y comprobar la equivalencia simbólica entre líneas consecutivas, el sistema emula el juicio del docente sobre la validez del paso, sin limitarse a comparar el resultado final.

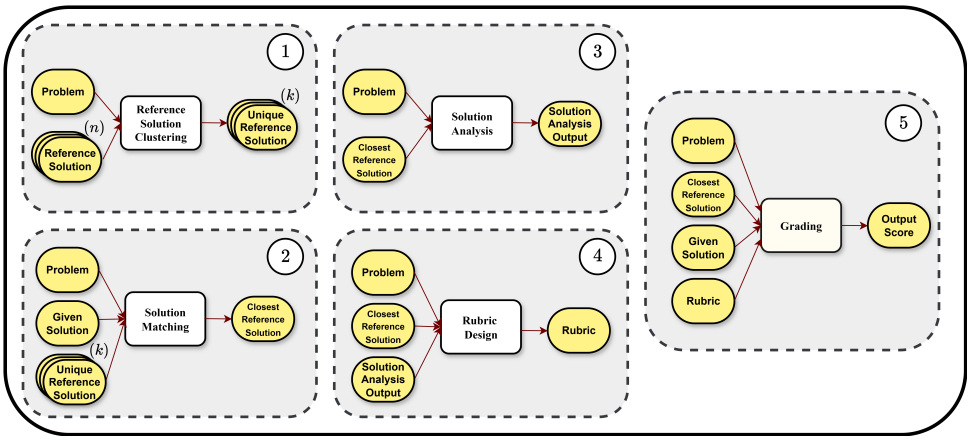
Calificación Automática con Validación del Procedimiento

Se entiende por calificación automática la asignación de una nota a un ejercicio sin intervención manual directa. En su forma básica, el sistema compara el resultado final del estudiante con la respuesta esperada. En su forma avanzada, el sistema verifica

además la validez de cada paso del procedimiento, otorgando crédito parcial cuando el razonamiento es correcto aunque el resultado final no lo sea, o penalizando un resultado correcto alcanzado por un procedimiento inválido (Figura 5). Esta distinción acerca la calificación automática al criterio pedagógico real, en el que el proceso importa tanto como el resultado.

Figura 5

Flujo de calificación automática basado en rúbricas por pasos



Nota. Tomado de (Hamed M. at el., 2025)

La legitimidad de un sistema de calificación automática depende de su concordancia con el juicio de docentes expertos y de la interpretabilidad de sus decisiones. Por ello, el estudio operacionaliza este concepto mediante el coeficiente κ de Cohen y el error absoluto medio (MAE) entre la nota automática y la nota docente, ambos definidos formalmente más adelante. La Tabla 15 sintetiza los conceptos anteriores, sus autores base y su uso concreto en la investigación.

Tabla 15

Bases conceptuales del estudio

Concepto	Autores base	Definición sintetizada	Uso en la investigación
Reconocimiento de expresiones matemáticas manuscritas (HMER)	Xie et al. (2024); Zanibbi y Blostein (2012)	Disciplina que transforma imágenes o trazos de expresiones matemáticas en una representación simbólica estructurada (LaTeX o MathML), resolviendo símbolo, segmentación y estructura.	Tarea central del sistema; se mide con ExpRate y CER.
Deep learning y redes neuronales profundas	LeCun et al. (2015); Vaswani et al. (2017)	Familia de métodos con redes de múltiples capas que aprenden representaciones jerárquicas; en visión destacan las CNN, las RNN o LSTM y los modelos de atención.	Sustenta la codificación visual y la decodificación simbólica (OE1).
Visión por computador y segmentación	Ronneberger, Fischer, y Brox (2015); Zhang et al. (2017)	Técnicas para interpretar contenido visual; la segmentación divide la imagen en regiones semánticamente coherentes preservando la información espacial.	Habilita la preservación de la jerarquía bidimensional; se evalúa con IoU (OE2).
Calificación automática con validación del procedimiento	Chaowicharat y Dejdumrong (2023); Parsaeifard et al. (2025)	Sistemas que asignan una nota sin intervención manual; en su forma avanzada verifican la validez de cada paso y no solo el resultado final.	Aporte original; se mide con κ de Cohen y MAE (OE3).
Modelos encoder – decoder con atención	Zhang et al. (2017); Zhao et al. (2021)	Arquitecturas de transducción de secuencias donde un codificador extrae una representación latente y un decodificador genera la salida atendiendo selectivamente a ella.	Define el espacio de soluciones explorado en el OE1.

Estos cinco (5) conceptos no operan de forma aislada. El HMER fija el objeto de estudio; el deep learning y los modelos encoder – decoder con atención aportan la maquinaria de aprendizaje; la visión por computador y la segmentación garantizan que la jerarquía espacial bidimensional —fracciones, exponentes, subíndices y radicales— se preserve antes de la decodificación; y la calificación automática con validación del procedimiento integra todo lo anterior en una salida evaluable frente al juicio docente. La cadena conceptual reproduce así la secuencia metodológica del proyecto: reconocer, estructurar y validar.

Modelos y Enfoques Teóricos

Esta sección expone las teorías, los modelos y los enfoques que dan soporte al problema y al análisis. Se inicia con los fundamentos del aprendizaje profundo y su entrenamiento; a continuación se desarrollan los tres enfoques teóricos que estructuran los objetivos del estudio y finalmente, el marco metodológico CRISP-DM que ordena el ciclo de la ciencia de datos.

Aprendizaje Profundo y Entrenamiento

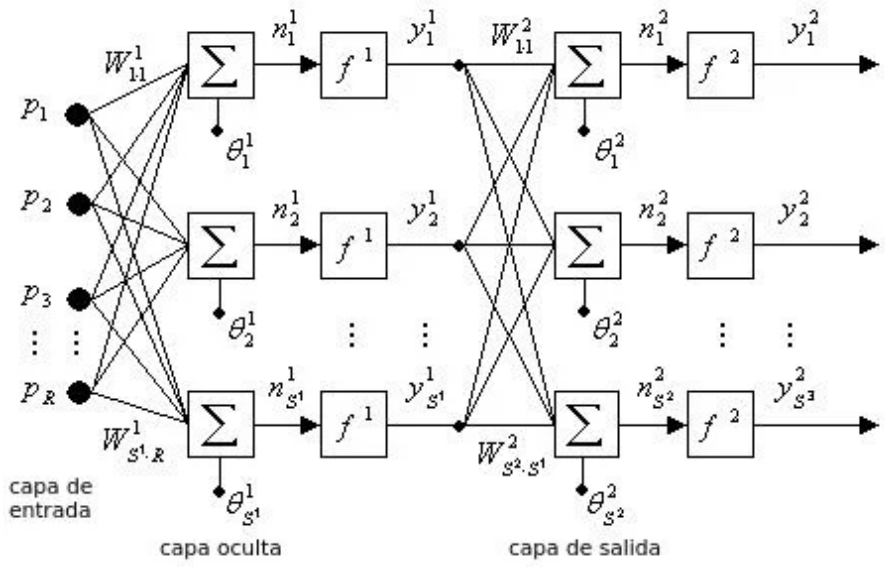
La unidad básica de una red neuronal es la neurona artificial o perceptrón, que calcula una combinación lineal de sus entradas, le suma un sesgo y aplica una función de activación no lineal. Formalmente, la salida $\mathbf{a}$ de una neurona con entradas $\mathbf{x}$, pesos $\mathbf{w}$ y sesgo $\mathbf{b}$ que se expresa según la Ecuación 1, donde σ denota la función de activación —por ejemplo, la sigmoide, la tangente hiperbólica o la unidad lineal rectificada (ReLU)—:

$$a = \sigma\left(\sum_{i=1}^{n} w_i x_i + b\right) = \sigma\left(w^\top x + b\right) \tag{1}$$

La no linealidad de σ es esencial: sin ella, la composición de capas colapsaría en una única transformación lineal, incapaz de representar las relaciones complejas presentes en una imagen. Al apilar muchas capas de neuronas se obtiene una red profunda, una función altamente parametrizada que transforma progresivamente la entrada en representaciones más abstractas, como se esquematiza en la Figura 6.

Figura 6

Esquema de bloques de una red neuronal feed – forward



Nota. Tomado de Arroyo Hernández et al. (2013); Figura 2, p. 67.

El aprendizaje de la red consiste en ajustar sus parámetros para que sus predicciones se aproximen a las etiquetas verdaderas. Este ajuste se realiza minimizando una función de pérdida que cuantifica el error. En problemas de clasificación y de generación de secuencias la pérdida habitual es la entropía cruzada, que penaliza la divergencia entre la distribución predicha $\hat{y}$ y la distribución verdadera y, según la Ecuación 2:

$$\mathcal{L}_{CE} = -\sum_{t=1}^{T} \sum_{k} y_{t,k} \log \hat{y}_{t,k} \tag{2}$$

Para minimizar la pérdida se emplea el descenso de gradiente: en cada iteración, los parámetros θ se actualizan en la dirección opuesta al gradiente de la pérdida, escalado por una tasa de aprendizaje η, como indica la Ecuación 3. En la práctica se utilizan variantes estocásticas que estiman el gradiente sobre lotes de datos y optimizadores adaptativos como Adam, que ajustan dinámicamente el paso de actualización:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}(\theta_t) \tag{3}$$

El cálculo eficiente del gradiente en redes con muchas capas se logra mediante el algoritmo de retropropagación (backpropagation), que aplica la regla de la cadena del

cálculo diferencial para propagar el error desde la capa de salida hacia las capas anteriores (Rumelhart, Hinton, y Williams, 1986). Así, el entrenamiento alterna dos fases: una propagación hacia adelante que produce la predicción y calcula la pérdida, y una retropropagación que distribuye el error y actualiza los pesos. Repetido sobre miles de ejemplos durante múltiples épocas, este proceso permite que la red aprenda a reconocer patrones visuales sin que tales patrones hayan sido programados explícitamente.

Redes Neuronales Convolucionales (CNN). Las redes neuronales convolucionales son la arquitectura de referencia para el procesamiento de imágenes, debido a que en lugar de conectar cada píxel con cada neurona, aplican filtros locales que se desplazan sobre la imagen detectando patrones independientemente de su posición, propiedad conocida como invarianza a la traslación (LeCun, Bottou, Bengio, y Haffner, 1998). La operación de convolución entre una imagen I y un núcleo K se define según la Ecuación 4:

$$(I \cdot K)(i, j) = \sum_{m} \sum_{n} I(i + m,\, j + n)\, K(m, n) \tag{4}$$

Cada filtro produce un mapa de características que resalta la presencia de un patrón específico. El apilamiento de capas convolucionales, intercaladas con operaciones de submuestreo (pooling) y activaciones no lineales, genera una jerarquía de representaciones: de bordes y trazos en las capas iniciales a símbolos completos en las capas profundas. Arquitecturas como las redes residuales, que introducen conexiones de salto para facilitar el entrenamiento de redes muy profundas, mitigaron el problema del desvanecimiento del gradiente y permitieron modelos de gran profundidad (He, Zhang, Ren, y Sun, 2016). En el sistema propuesto, una CNN o una variante densa actúa como codificador visual que transforma la imagen del ejercicio en el mapa de características sobre el que opera el decodificador.

Redes Recurrentes y Memoria de Largo Plazo (RNN y LSTM). Las redes neuronales recurrentes procesan secuencias manteniendo un estado interno que se actualiza con cada elemento, lo que las hace idóneas para generar cadenas LaTeX símbolo a símbolo. Sin embargo, las RNN simples sufren el desvanecimiento del

gradiente al modelar dependencias largas. Las redes de memoria de largo y corto plazo (*Long Short-Term Memory*, LSTM) resolvieron esta limitación mediante un mecanismo de compuertas que regula qué información se conserva, se actualiza o se descarta a lo largo de la secuencia (Hochreiter y Schmidhuber, 1997). En el paradigma codificador – decodificador, una LSTM como decodificador genera la secuencia de salida condicionada por la representación visual; este fue el esquema del modelo WAP y de sus sucesores recurrentes.

El Transformer y el Mecanismo de Atención. El Transformer prescindió de la recurrencia y basó todo el procesamiento en el mecanismo de atención (Vaswani et al., 2017). La idea de la atención, introducida en el contexto de la traducción automática (Bahdanau, Cho, y Bengio, 2015), permite que el modelo pondere de forma diferenciada las distintas partes de la entrada al producir cada elemento de la salida. En su formulación de producto escalar escalado, la atención calcula pesos a partir de consultas Q, claves K y valores V, según la Ecuación 5:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V \tag{5}$$

En el caso del reconocimiento de expresiones, el mecanismo de atención permite que el decodificador centre su atención, en cada paso de generación, en la región de la imagen que corresponde al símbolo que está transcribiendo. Los pesos de atención α se normalizan mediante la función softmax y producen un vector de contexto c_t que resume la información visual relevante, según la Ecuación 6:

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_j \exp(e_{t,j})}, \qquad c_t = \sum_i \alpha_{t,i}\, h_i \tag{6}$$

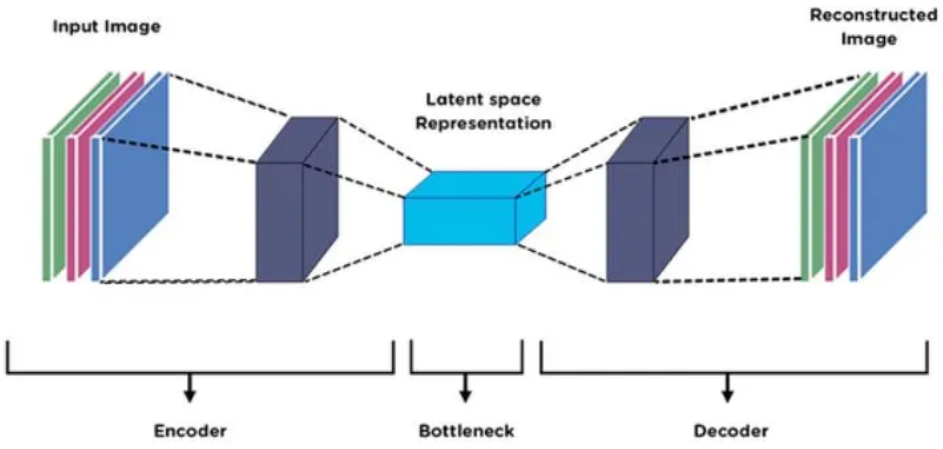
La atención multi – cabeza ejecuta varias proyecciones de atención en paralelo, capturando relaciones de distinta naturaleza, y la auto – atención (self – attention) permite que cada elemento de una secuencia atienda a los demás, modelando dependencias de largo alcance sin recurrencia. Estas propiedades explican por qué los modelos basados en Transformer dominan el estado del arte del HMER y por qué el proyecto los adopta como candidatos principales del objetivo OE1.

Encoder – Decoder con Atención (Enfoque Arquitectónico)

Inaugurado por el modelo WAP (Zhang et al., 2017), este enfoque plantea el reconocimiento como un problema de traducción imagen – a – secuencia: un codificador convolucional construye una representación visual densa y un decodificador, mediante atención, produce la secuencia LaTeX. Su hipótesis teórica es que la atención aprende automáticamente la correspondencia entre regiones de la imagen y símbolos de salida, sin segmentación explícita ni gramáticas manuales. Variantes como DenseWAP (Zhang et al., 2018), BTTR (Zhao et al., 2021) y CoMER (Zhao y Gao, 2022) consolidan este paradigma como estado del arte y fundamentan directamente el OE1 del proyecto de investigación, que compara arquitecturas CNN, CRNN y basadas en Transformer. La Figura 7 esquematiza esta arquitectura (Sutskever, Vinyals, y Le, 2014; Zhang et al., 2017).

Figura 7

Esquema del modelo codificador – decodificador con atención para HMER



Nota. https://medium.com/@ahmadsabry678/a-perfect-guide-to-understand-encoder-decoders-in-depth-with-visuals-30805c23659b

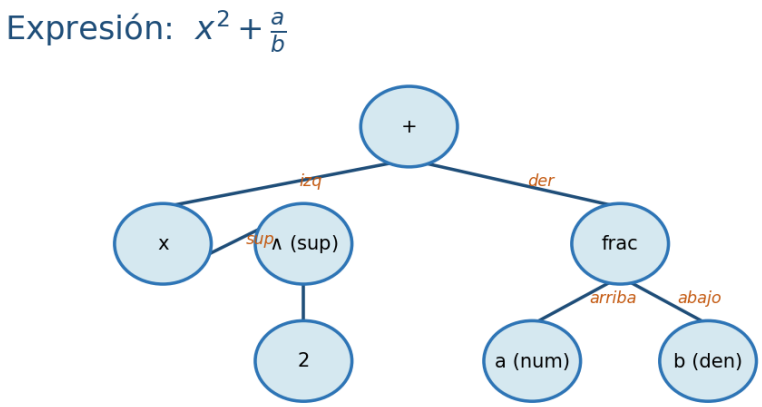
La atención permite al decodificador atender selectivamente a regiones del mapa de características en cada paso de generación y la principal virtud de este enfoque es su carácter integral: al aprender de extremo a extremo, evita la propagación de errores entre etapas separadas. Su principal reto es la asignación precisa de la atención, abordado por los mecanismos de cobertura discutidos en los antecedentes.

Gramática Bidimensional y Árboles de Derivación (Enfoque Estructural)

La notación matemática no es una secuencia lineal, sino un objeto con estructura bidimensional y jerárquica; es decir un exponente, un subíndice o una fracción modifican el significado de los símbolos contiguos. El enfoque estructural sostiene que el reconocimiento es incompleto si no se preserva esa jerarquía. Trabajos como SAN (Yuan et al., 2022) y las variantes basadas en GNN sistematizadas por Truong, Nguyen, Zanibbi, Mouchère, y Nakagawa (2024) modelan la expresión como un árbol de derivación, donde cada nodo es un símbolo y cada arista una relación espacial. La Figura 8 ilustra el árbol de disposición de símbolos (*Symbol Layout Tree*) de una expresión sencilla, evidenciando que relaciones como superíndice, numerador y denominador no pueden reducirse a un orden lineal sin pérdida de significado.

Figura 8

Árbol de disposición de símbolos (Symbol Layout Tree) de una expresión



Nota. Elaboración propia

Las aristas codifican relaciones espaciales —superíndice, numerador y denominador—: la estructura bidimensional no puede reducirse a una secuencia lineal sin pérdida de significado.

Modelos como PosFormer, que aprende explícitamente la posición jerárquica de cada símbolo, y las arquitecturas con decodificación en árbol o con codificadores de grafo materializan este enfoque (Guan et al., 2024; Truong et al., 2024). Este enfoque fundamenta el OE2, que propone un pipeline de segmentación con preservación de la jerarquía espacial validado mediante el IoU y la exactitud estructural frente a una

segmentación plana.

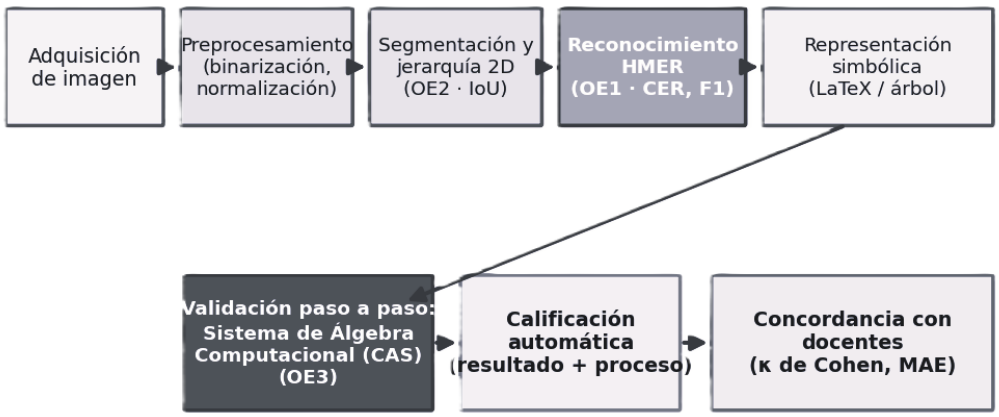
Juicio Humano y Analítica del Aprendizaje (Enfoque Evaluativo)

El tercer enfoque se sitúa en la intersección entre el aprendizaje automático y la evaluación educativa. Plantea que un sistema de calificación automática solo es legítimo si su salida concuerda con la calificación humana experta y si su decisión es interpretable (Siemens y Long, 2011). Inspirado en la analítica del aprendizaje y en estudios de calificación automática de matemáticas (Chaowicharat y Dejdumrong, 2023; Parsaeifard et al., 2025), exige que el sistema vaya más allá del resultado final y evalúe la coherencia lógica del procedimiento. Sustenta el OE3 del proyecto de investigación y se operacionaliza mediante el coeficiente κ de Cohen (Cohen, 1960; Landis y Koch, 1977) y el error absoluto medio entre la calificación del sistema y la del docente evaluador.

Su fundamento técnico es el acoplamiento entre el reconocimiento y un sistema de álgebra computacional (CAS) capaz de verificar la equivalencia simbólica entre los pasos de una derivación. Conceptualmente, el CAS implementa de forma algorítmica los métodos de razonamiento descritos en las bases conceptuales y representa cada línea como un árbol de expresión, normaliza ambas líneas y comprueba si son equivalentes; si toda la cadena conserva la equivalencia, el procedimiento es válido (Chaowicharat y Dejdumrong, 2023). La Figura 9 integra los tres enfoques en una arquitectura conceptual única y muestra dos lazos de evaluación complementarios: uno interno, que mide la calidad del reconocimiento mediante CER, F1-score e IoU, y otro externo, que mide la validez del juicio mediante la concordancia con el docente evaluador.

Figura 9

Arquitectura conceptual del sistema de calificación con validación del procedimiento



Nota. Elaboración propia. Dos lazos de evaluación: (a) calidad del reconocimiento, con métricas internas, y (b) validez del juicio, con la concordancia frente al experto humano.

El enfoque reconoce, además, una limitación heredada de los antecedentes,; como la validez del procedimiento depende de la robustez del reconocimiento, pues un símbolo mal reconocido puede invalidar erróneamente un paso correcto. De ahí la estrategia de reemplazo de símbolos de baja confianza propuesta por Chaowicharat y Dejdumrong (2023) y la decisión de este proyecto de evaluar el aporte del módulo de validación frente a una línea base que solo califica el resultado final. La concordancia se mide con el coeficiente κ de Cohen, apropiado para exactamente dos evaluadores, y con el error absoluto medio.

Marco Metodológico del Ciclo de Datos (CRISP-DM)

Más allá de las arquitecturas, el estudio adopta el modelo CRISP-DM (*Cross-Industry Standard Process for Data Mining*) como marco que ordena el ciclo de la ciencia de datos (Wirth y Hipp, 2000). CRISP-DM estructura el trabajo en seis (6) fases iterativas y garantiza la trazabilidad entre las preguntas de investigación, los objetivos y las métricas. CRISP-DM provee el andamiaje que conecta el marco teórico con la metodología, asegurando que cada decisión de modelado responda a un objetivo y se evalúe con una métrica definida. La Tabla 16 alinea las fases de CRISP-DM con los objetivos del proyecto.

Tabla 16

Alineación entre las fases de CRISP-DM y los objetivos del proyecto

Fase CRISP-DM	Actividad en el proyecto	Objetivo o métrica
Comprensión del problema	Definición de la calificación automática con validación del procedimiento.	Preguntas de investigación
Comprensión de los datos	Análisis exploratorio de CROHME, MathWriting y HASYv2.	Caracterización descriptiva
Preparación de los datos	Binarización, normalización y aumento de datos.	Calidad y equilibrio de clases
Modelado	Comparación CNN, CRNN y Transformer; segmentación; validación mediante CAS.	OE1, OE2 y OE3
Evaluación	Validación cruzada y concordancia con el docente evaluador.	F1-score, CER, IoU, κ y MAE
Despliegue	Prototipo funcional integrable en plataformas educativas.	Resultado integrador

Variables, Dimensiones ó Categorías de Análisis

Por tratarse de un estudio cuantitativo de ciencia de datos, la literatura permite justificar las variables que se observarán en la fase experimental. Cada variable se vincula con un indicador medible y con la fuente teórica o el antecedente que la respalda, según se detalla en la Tabla 17. Los umbrales cuantitativos asociados constituyen metas referenciales de planificación, sujetas a validación empírica y no compromisos fijos.

Tabla 17

Variables, dimensiones e indicadores derivados de la literatura

Variable o dimensión	Definición conceptual	Indicador posible	Fuente teórica o antecedente
Reconocimiento simbólico	Capacidad del modelo para identificar correctamente símbolos y expresiones manuscritas.	Exactitud, F1-score, CER y ExpRate (meta referencial: F1-score $\geq$ 0.90; CER $\leq$ 10 %).	Zhang et al. (2017); Zhao et al. (2021); Xie et al. (2024).
Preservación de la jerarquía espacial bidimensional	Grado en que la segmentación conserva la estructura bidimensional de la expresión.	IoU y exactitud estructural (meta referencial: IoU $\geq$ 0.80).	Yuan et al. (2022); Ronneberger et al. (2015).
Validez del procedimiento	Coherencia lógica de cada paso de la derivación manuscrita, más allá del resultado final.	Verificación paso a paso; aporte frente a la línea base que solo evalúa el resultado.	Chaowicharat y Dejdumrong (2023).
Concordancia con el docente	Acuerdo entre la calificación del sistema y la del docente evaluador.	Coeficiente κ de Cohen y MAE (meta referencial: $\kappa \geq 0.75$).	Cohen (1960); Landis y Koch (1977); Parsaeifard et al. (2025).

La derivación de estas variables desde la literatura cumple una función metodológica precisa: garantiza que las métricas de evaluación no se elijan de manera arbitraria, sino que respondan a estándares consolidados del campo —ExpRate y CER en CROHME, IoU en segmentación y κ de Cohen en concordancia entre evaluadores—. De este modo, cada objetivo específico queda asociado a una variable, un indicador y una fuente, asegurando la trazabilidad entre el marco teórico y el diseño experimental.

Métricas OE

Los apartados siguientes definen formalmente cada métrica:

Métricas de Reconocimiento (OE1)

La tasa de error de caracteres (CER) mide la proporción de operaciones de edición —sustituciones S, eliminaciones D e inserciones I— necesarias para transformar la secuencia predicha en la de referencia de longitud N, según la Ecuación 7. Un CER bajo indica una transcripción fiel a nivel de carácter:

$$\text{CER} = \frac{S + D + I}{N} \tag{7}$$

La tasa de reconocimiento exacto a nivel de expresión (ExpRate) es más estricta: contabiliza la proporción de expresiones reconocidas sin ningún error, según la Ecuación 8. Es la métrica de referencia en las competiciones CROHME:

$$\text{ExpRate} = \frac{\#\{\text{expresiones exactas}\}}{\#\{\text{expresiones totales}\}} \tag{8}$$

Para la clasificación de símbolos individuales se emplean la precisión P, la exhaustividad R y su media armónica, el F1-score, definidas en la Ecuación 9 en términos de verdaderos positivos (TP), falsos positivos (FP) y falsos negativos (FN). El F1-score equilibra la capacidad de no etiquetar erróneamente y la de no omitir símbolos:

$$P = \frac{TP}{TP + FP}, \qquad R = \frac{TP}{TP + FN}, \qquad F_1 = \frac{2PR}{P + R} \tag{9}$$

Métricas de Segmentación Estructural (OE2)

El índice de solapamiento (IoU) cuantifica la coincidencia entre la región predicha A y la región de referencia B como el cociente entre el área de su intersección y el área de su unión. Según la Ecuación 10, un IoU igual a uno indica coincidencia perfecta; valores cercanos a cero indican desalineación. Es la métrica estándar para evaluar detección y segmentación (Everingham, Van Gool, Williams, Winn, y Zisserman, 2010):

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|} \tag{10}$$

La exactitud estructural complementa al IoU verificando que el orden de lectura horizontal de las regiones detectadas coincida con el de la referencia. Es una comprobación unidimensional: condición necesaria pero no suficiente para la jerarquía bidimensional, cuya fidelidad espacial queda evidenciada por el IoU sobre cada región.

Métricas de Validación del Procedimiento (OE3)

El coeficiente κ de Cohen mide la concordancia entre dos evaluadores corrigiendo el acuerdo esperado por azar, según la Ecuación 11, donde p_o es la proporción de acuerdo observado y p_e la proporción de acuerdo esperado por azar (Cohen, 1960):

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{11}$$

La elección del κ de Cohen es metodológicamente deliberada: el coeficiente está definido para exactamente dos (2) evaluadores y, en este proyecto, esos dos (2) evaluadores son el sistema automático y el docente que califica la misma muestra de ejercicios. Su cálculo se realiza sobre una escala binaria de dos niveles —aprueba y reprueba— aplicada a cada ejercicio, lo que evita los desacuerdos triviales de magnitud propios de una nota continua y estabiliza la estimación con muestras moderadas. La interpretación de su magnitud sigue la escala de Landis y Koch (1977), resumida en la Tabla 18, que sitúa la meta referencial del proyecto en el umbral superior del rango de concordancia sustancial.

Tabla 18

Escala de interpretación del coeficiente κ de Cohen

Rango de κ	Nivel de concordancia	Lectura en el proyecto
< 0.00	Sin concordancia	Inaceptable
0.00 – 0.20	Insignificante (*slight*)	Inaceptable
0.21 – 0.40	Aceptable baja (*fair*)	Insuficiente
0.41 – 0.60	Moderada	Limítrofe
0.61 – 0.80	Sustancial	Meta referencial ($\kappa \geq 0.75$)
0.81 – 1.00	Casi perfecta	Resultado óptimo

Se tiene la elaboración propia a partir de Landis y Koch (1977). La meta $\kappa \geq 0.75$ es una meta de planificación de carácter referencial.

Finalmente, el error absoluto medio (MAE) cuantifica la diferencia promedio entre la nota automática $\hat{g}$ y la nota docente g sobre n ejercicios, según la Ecuación 12. A diferencia del κ, que evalúa la concordancia categórica, el MAE mide la magnitud del

error en la escala continua de calificación:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |g_i - \hat{g}_i| \tag{12}$$

El reporte conjunto de estas métricas asegura una evaluación robusta y reproducible, requisito de validez interna del estudio. Conviene reiterar que los umbrales cuantitativos —F1-score ≥ 0.90, CER $\leq 10\,\%$, IoU ≥ 0.80 y $\kappa \geq 0.75$— constituyen metas de planificación de carácter referencial, sujetas a validación empírica.

Brechas de Investigación

La brecha no se formula como una afirmación genérica, sino que surge de la comparación de los estudios analizados y de sus limitaciones. Las Tablas 19, 20 sintetizan qué se sabe, qué no se ha resuelto suficientemente y cómo la investigación propuesta responde a esa necesidad. Así mismo, conviene precisar un matiz metodológico que se deriva de la revisión y que orientará la fase experimental; dado que los conjuntos públicos contienen principalmente expresiones aisladas, la evaluación del módulo de validación del procedimiento (OE3) deberá apoyarse en estrategias complementarias.

Tabla 19

Síntesis de la brecha de investigación 1

Elemento	Redacción
Lo que se sabe	Los estudios previos han logrado reconocer expresiones matemáticas manuscritas con alta exactitud mediante modelos encoder – decoder con atención y decodificadores Transformer, y han documentado métricas, conjuntos de datos y arquitecturas de referencia

Tabla 20

Síntesis de la brecha de investigación 2

Elemento	Redacción
Lo que falta	Sin embargo, la mayoría de los trabajos optimizan el reconocimiento y no integran la validación del procedimiento en escenarios reales de calificación; los que la abordan reconocen fragilidad ante errores de reconocimiento y muestras limitadas (Chaowicharat y Dejdumrong, 2023; Parsaeifard et al., 2025). Además, los conjuntos de datos públicos contienen mayoritariamente expresiones aisladas y no procedimientos multipaso completos.
Necesidad del estudio	Por ello, esta investigación propone un sistema que articule reconocimiento, segmentación estructural y validación paso a paso del procedimiento, evaluado contra el juicio de un docente experto bajo condiciones controladas, sobre conjuntos de datos públicos consolidados (MathWriting, CROHME y HASYv2).
Aporte esperado	El aporte se centra en cuantificar el valor diferencial del módulo de validación del procedimiento frente a un sistema que solo evalúa el resultado final, midiendo su concordancia con los docentes mediante el κ de Cohen y aportando evidencia replicable sobre la viabilidad de la calificación automática en educación matemática.

Síntesis del Marco Teórico

La revisión de la literatura y el marco teórico desarrollados sostienen la investigación en una secuencia coherente: el problema se conecta con conceptos claramente definidos (HMER, deep learning, segmentación, calificación automática y modelos de atención), con tres (3) enfoques teóricos (arquitectónico, estructural y

evaluativo) y con una matriz de antecedentes que evidencia la brecha. De esa articulación se derivan las variables del estudio y sus indicadores, lo que asegura la trazabilidad entre la teoría y el método.

La conexión con la metodología es directa, es decir el enfoque arquitectónico justifica la comparación de arquitecturas CNN, CRNN y Transformer del Objetivo Específico 1 (OE1), evaluada con exactitud, F1-score, CER y ExpRate. El enfoque estructural fundamenta el pipeline de segmentación del Objetivo Específico 2 (OE2), validado con el IoU y la exactitud estructural. El enfoque evaluativo sostiene el módulo de validación del procedimiento del Objetivo Específico 3 (OE3) y, dentro de él, el contraste de la calificación del prototipo frente al juicio docente, medido mediante el coeficiente κ de Cohen y el error absoluto medio. Así, el marco teórico no solo describe el estado del arte, sino que prepara y justifica las decisiones de la fase experimental, en coherencia con el ciclo CRISP-DM que estructura la investigación (Wirth y Hipp, 2000).

En conjunto, el capítulo permite comprender por qué la investigación es necesaria, qué conocimiento la sustenta y qué decisiones metodológicas se derivan de la literatura. La Tabla 21 verifica el cumplimiento de los criterios de control establecidos para el capítulo.

Tabla 21

Lista de verificación del capítulo

Pregunta de control	Cumple
¿El capítulo conecta problema, pregunta y objetivos?	Sí
¿La revisión de literatura es trazable y organizada?	Sí
¿Los conceptos clave están definidos y citados?	Sí
¿Los antecedentes se analizan críticamente?	Sí
¿Las variables se justifican desde la literatura?	Sí
¿La brecha de investigación es clara y específica?	Sí

Capítulo III: Metodología

Introducción del Capítulo

Si bien el Capítulo I delimitó el problema y el Capítulo II construyó el marco conceptual que lo sostiene; este capítulo constituye la arquitectura de las decisiones que convierte el problema en un proceso empírico verificable. Su propósito no es argumentar qué se sabe, sino declarar cómo se producirá la evidencia, con qué datos, bajo qué controles y contra qué criterios se juzgará el resultado (Hernández Sampieri y Mendoza Torres, 2023).

El criterio rector es la replicabilidad, cada decisión se enuncia de modo que un investigador ajeno al proyecto pueda reproducir los experimentos y contrastar las conclusiones: conjuntos de datos identificados por versión, particiones fijas, semillas declaradas, métricas definidas formalmente y protocolos de evaluación explícitos. Esta exigencia es coherente con los principios de transparencia adoptados desde la revisión tipo PRISMA del Capítulo II y con la naturaleza computacional del estudio, en el que la evidencia se genera mediante experimentos ejecutables y no mediante observación de campo.

Enfoque de la Investigación

El estudio adopta un enfoque cuantitativo, de carácter aplicado y experimental, enmarcado en la investigación en ciencia de datos. La elección es una consecuencia de la pregunta principal: indagar si un sistema automático alcanza una precisión «comparable a la calificación realizada por un docente humano» obliga a medir esa comparabilidad en una escala, no a interpretarla. Los tres (3) objetivos específicos fueron redactados con métricas explícitas —exactitud, F1-score, CER, ExpRate, IoU, κ de Cohen y error absoluto medio—, de modo que la evidencia requerida es necesariamente numérica.

El enfoque se justifica por la naturaleza objetiva del fenómeno, pues cada predicción se contrasta contra una referencia (*ground truth*) inequívoca; por su alineación con objetivos formulados con umbrales verificables; por la replicabilidad que garantizan los conjuntos públicos, las particiones fijas y las semillas controladas; y por su coherencia con la literatura de reconocimiento de expresiones matemáticas manuscritas

(HMER) revisada en el Capítulo II, lo que asegura la comparabilidad de los resultados.

Diseño de la Investigación

El diseño es experimental – comparativo de carácter computacional. Se manipulan de manera controlada las variables independientes y se observa su efecto sobre las variables dependientes de desempeño, manteniendo constantes los conjuntos de datos, las particiones y las condiciones de evaluación. Así, las diferencias observadas entre configuraciones resultan atribuibles a las decisiones de diseño y no a variaciones del entorno experimental.

A este núcleo se suma un componente correlacional de concordancia dentro del OE3, en el que la calificación del prototipo se contrasta con la de un docente evaluador para estimar el grado de acuerdo. Se descartaron los diseños alternativos por insuficiencia y no por irrelevancia: el descriptivo permite caracterizar los conjuntos en el análisis exploratorio, pero no comparar modelos; el exploratorio es propio de fenómenos poco conocidos, condición que no se cumple aquí, donde existen marco teórico, antecedentes y métricas consolidadas. El diseño se ejecuta sobre la metodología CRISP-DM (Wirth y Hipp, 2000), cuya correspondencia con los objetivos y las métricas se estableció en la Tabla 16 del Capítulo II y cuyo contenido operativo se desarrolla en la sección de procesamiento y análisis.

Población, Muestra y Fuentes de Datos

Población

La población está conformada por el universo de expresiones y ejercicios matemáticos manuscritos susceptibles de ser reconocidos y calificados automáticamente: símbolos numéricos, operadores, variables y, de manera crítica para esta investigación, las estructuras bidimensionales de la notación matemática (fracciones, exponentes, subíndices y radicales). Conforme al alcance declarado en la Tabla 4 del Capítulo I, se restringe a ejercicios de aritmética y álgebra elemental de nivel medio, con notación estandarizada y capturados como imágenes estáticas (*offline*).

Muestra y Marco Muestral

Dado el carácter computacional del estudio, la muestra se materializa en dos (2) componentes de naturaleza distinta. El primero lo constituye el corpus público de CROHME, que actúa como marco muestral representativo del fenómeno y permite la comparación directa con la literatura al disponer de particiones oficiales. El segundo es una muestra propia y necesaria porque los conjuntos públicos contienen mayoritariamente expresiones aisladas y no procedimientos multipaso completos que son justamente el objeto del OE3.

Tabla 22

Fuentes de datos, contenido y función según objetivo específico

Fuente	Contenido	Función en el proyecto	Tipo de dato
CROHME (Mahdavi et al., 2019; Mouchère, Viard-Gaudin, Zanibbi, y Garain, 2016)	Conjunto de la competencia estándar de reconocimiento de expresiones manuscritas, con particiones oficiales e imágenes renderizadas en modalidad *offline*.	**Corpus experimental**: entrenamiento y evaluación de las arquitecturas (OE1) mediante exactitud, F1-score, CER y ExpRate, y validación de la segmentación (OE2).	Secundario, público
MathWriting (Gervais et al., 2025)	Expresiones matemáticas manuscritas a gran escala, en formato de trazos (*online*).	Identificado en la revisión; reservado para extensión futura, pues su uso exigiría renderizar los trazos a imagen.	Secundario, público
HASYv2 (Thoma, 2017)	Símbolos matemáticos manuscritos aislados y etiquetados.	Identificado en la revisión; no permite calcular ExpRate ni CER a nivel de expresión al carecer de expresiones completas.	Secundario, público
Muestra propia	Hojas de ejercicios manuscritos con desarrollo multipaso y su calificación docente.	Verificación del módulo de procedimiento y concordancia con el docente evaluador (OE3).	Primario, propio

Estrategia de Muestreo y Particiones

La estrategia persigue dos garantías simultáneas: representatividad de las estructuras críticas y ausencia de contaminación entre entrenamiento y evaluación.

- Cobertura de las estructuras críticas: el corpus no se estratifica artificialmente por tipo de símbolo ni por complejidad estructural. La razón es que las particiones oficiales de CROHME ya fueron construidas por la competencia con esa cobertura, y reorganizarlas rompería la comparabilidad con los resultados publicados, que es precisamente el motivo por el que se eligió este corpus. La presencia efectiva de las estructuras críticas para el OE2 —fracciones, exponentes, subíndices y radicales— se verifica y documenta en la caracterización del corpus del Capítulo IV, que reporta el vocabulario completo y la distribución de los tokens, en lugar de imponerse por diseño del muestreo.

- Particiones de modelado: el corpus experimental dispone de particiones oficiales —CROHME 2014, 2016 y 2019—, que se respetan sin modificación alguna. Sobre la partición oficial de entrenamiento se reserva un 10 % como conjunto de validación, separado por escritor, destinado a supervisar la convergencia y a seleccionar la época de cada modelo. No se emplea validación cruzada de k pliegues: repartir en pliegues un conjunto que ya tiene particiones oficiales impediría la comparación directa con la literatura y multiplicaría por k un presupuesto de cómputo que, como se declaró en el Capítulo I, es la restricción principal del estudio. La incertidumbre de cada métrica se estima, en su lugar, mediante intervalos de confianza por remuestreo *bootstrap* sobre las predicciones del conjunto de prueba.

- Partición por escritor: siempre que la fuente lo permita, la separación se realizará por escritor y no por expresión, para evitar que el modelo se evalúe sobre la caligrafía con la que fue entrenado y se sobrestime su capacidad de generalización.

- Muestra de concordancia (OE3): subconjunto de ejercicios con desarrollo completo, calificado de forma independiente por el sistema y por un docente

evaluador. La unidad de análisis es el ejercicio y no la prueba completa, de modo que una misma hoja aporta tantas observaciones como ejercicios contenga; de ello depende que el κ pueda estimarse con un intervalo de confianza reportable y no quede determinado por un puñado de decisiones globales. Cada ejercicio recibe una decisión binaria de dos (2) niveles —aprueba, reprueba—, sobre la que se calcula el κ; de la agregación de esas decisiones se obtiene además la nota numérica de la prueba, sobre la que se calcula el MAE. Ambas métricas son así complementarias y no redundantes: el κ mide el acuerdo en la decisión discreta descontando el azar, mientras el MAE cuantifica la magnitud de la discrepancia en la nota final.

Una vez fijadas, las particiones se versionan y se conservan sin modificación durante todo el ciclo experimental. Ninguna decisión de ajuste de hiperparámetros se toma consultando la partición de prueba, empleada una sola vez y únicamente para el reporte final.

Técnicas de Recolección de Datos

Operacionalización de las Variables

La Tabla 17 del Capítulo II estableció el respaldo teórico de cada variable; la Tabla 23, 24 completa esa derivación especificando su rol experimental, su definición operacional y el indicador con el que se registrará.

Tabla 23

Matriz de operacionalización de las variables del estudio

Variable	Rol	Definición operacional	Indicador o métrica
Arquitectura del modelo	Independiente	Reconocedor empleado: CNN, CRNN con pérdida CTC o Transformer.	Configuración categórica
Configuración del *pipeline*	Independiente	Parámetros de binarización, segmentación y reconstrucción jerárquica.	Configuración categórica

Tabla 24

Matriz de operacionalización de las variables del estudio

Variable	Rol	Definición operacional	Indicador o métrica
Modalidad de calificación	Independiente	Calificación del resultado final frente a validación paso a paso.	Configuración dicotómica
Reconocimiento simbólico	Dependiente	Transcripción correcta de la expresión manuscrita a notación simbólica.	Exactitud, F1-score, CER y ExpRate
Calidad de la segmentación	Dependiente	Delimitación de regiones y relaciones espaciales entre símbolos.	IoU y exactitud estructural
Validez del procedimiento	Dependiente	Equivalencia matemática entre líneas consecutivas del desarrollo.	Pasos correctamente clasificados frente a la línea base
Concordancia con el docente	Dependiente	Acuerdo con el docente evaluador sobre cada ejercicio en escala binaria de dos niveles: aprueba y reprueba.	κ de Cohen sobre la decisión binaria; MAE sobre la nota numérica
Condiciones de imagen	Control	Resolución, iluminación, encuadre y formato de captura.	Parámetros fijados
Hiperparámetros y semilla	Control	Tasa de aprendizaje, épocas, tamaño de lote y semilla aleatoria.	Valores fijados y declarados

Técnicas e Instrumentos

Coherentes con el enfoque cuantitativo, se emplean tres técnicas diferenciadas por el tipo de dato que producen:

- Análisis documental y recolección secundaria: adquisición y curación de los conjuntos públicos, con verificación de licencias, comprobación de integridad, unificación de formatos y registro de la versión empleada en cada experimento.
- Observación computacional estructurada: registro automatizado de las

predicciones y de las métricas de cada corrida mediante registros (*logs*) reproducibles que incluyen la configuración completa, la semilla y el entorno de ejecución.

- Recolección primaria controlada: captura de imágenes de ejercicios resueltos a mano por escritores adultos voluntarios y registro de la calificación asignada por el docente evaluador mediante una rúbrica binaria de dos niveles —aprueba y reprueba—, común al docente y al sistema, condición indispensable para que el κ de Cohen sea interpretable.

Procesamiento y Análisis de los Datos

Preparación de los Datos

La preparación transforma las imágenes originales en entradas homogéneas: conversión a escala de grises y binarización adaptativa, corrección de inclinación, normalización del tamaño preservando la relación de aspecto —requisito para no distorsionar la proporción entre numerador y denominador o entre base y exponente— y eliminación del ruido de fondo.

Sobre esa base se aplica el aumento de datos mediante rotaciones y cizallamientos de pequeña magnitud, variaciones de grosor de trazo y de contraste. La magnitud se acota deliberadamente: una rotación excesiva puede alterar la relación espacial que distingue un exponente de un factor y convertir una muestra válida en una etiqueta incorrecta. El desequilibrio entre clases de símbolo —inevitable en notación matemática, donde los dígitos y los operadores básicos aparecen con una frecuencia muy superior a la de los símbolos especializados— no se corrige mediante remuestreo ni mediante ponderación de la función de pérdida. La decisión es deliberada: ambas correcciones alteran la distribución efectiva de entrenamiento y su efecto podría interactuar de manera distinta con cada arquitectura, contaminando justamente la comparación que persigue el OE1. El desequilibrio se absorbe, en cambio, en la métrica: el F1-score se reporta como macropromedio sobre las clases de símbolo, de modo que las clases minoritarias pesan lo mismo que las frecuentes y una degradación en los símbolos

poco comunes no queda enmascarada por el acierto en los dígitos.

Modelos y Técnicas Analíticas por Objetivo

- Reconocimiento (OE1): una red convolucional como línea base; una arquitectura híbrida convolucional – recurrente entrenada con la función de pérdida CTC; y un modelo codificador – decodificador basado en Transformer con atención. Las tres se entrenan bajo protocolo idéntico para que la comparación sea atribuible a la arquitectura.

- Segmentación y jerarquía espacial (OE2): *pipeline* de visión por computador que delimita las regiones de símbolo y reconstruye las relaciones espaciales en un árbol de disposición.

- Validación del procedimiento y concordancia (OE3): módulo que representa cada línea del desarrollo como expresión simbólica y comprueba la equivalencia matemática entre líneas consecutivas mediante un sistema de álgebra computacional (Meurer et al., 2017), siguiendo el planteamiento de Chaowicharat y Dejdumrong (2023). Su aporte se cuantifica frente a una línea base que califica solo el resultado final, y la calificación resultante se contrasta con la del docente evaluador sobre la muestra de concordancia mediante el κ de Cohen (Cohen, 1960) y el error absoluto medio.

Protocolo Experimental y Análisis Estadístico

Para que las comparaciones sean válidas, el protocolo fija todo aquello que no constituye la variable manipulada, cada experimento declara la semilla aleatoria, la partición, los hiperparámetros y la versión del entorno. Dada la restricción de memoria de la GPU disponible, cada configuración se entrena con una única semilla, fijada e idéntica para todas las arquitecturas, de modo que la comparación permanezca atribuible al diseño y no a la inicialización. En consecuencia, la significancia de las diferencias no se establece por la dispersión entre repeticiones sino mediante pruebas pareadas sobre las predicciones del mismo conjunto de prueba, que son válidas con una sola corrida por configuración. Por esa misma restricción de recursos no se realiza

búsqueda de hiperparámetros: la tasa de aprendizaje, el tamaño de lote, el número de épocas y la política de optimización se fijan a priori con valores idénticos para las tres arquitecturas y se declaran junto con los resultados. La decisión es coherente con el propósito del OE1, ya que una búsqueda desigual —o incluso una búsqueda igual pero guiada por la partición de validación— introduciría como variable el esfuerzo de optimización recibido por cada arquitectura, que es exactamente lo que el diseño busca mantener constante. La partición de validación se emplea únicamente para supervisar la convergencia y seleccionar la época de cada modelo, y ninguna arquitectura recibe un tratamiento distinto de las demás.

El análisis estadístico emplea, para cada comparación, la prueba pertinente. La prueba de McNemar determina si la diferencia entre dos arquitecturas evaluadas sobre el mismo conjunto de prueba es estadísticamente significativa, por tratarse de predicciones pareadas sobre las mismas expresiones. Los intervalos de confianza obtenidos por remuestreo *bootstrap* estiman la incertidumbre de los valores reportados. Este mismo criterio es el que sustenta el umbral del OE3: la mejora del módulo de validación paso a paso frente a la línea base que solo califica el resultado final se declara significativa únicamente cuando la prueba pareada correspondiente así lo respalda.

Métricas y Umbrales Referenciales

Las métricas fueron definidas formalmente en el Capítulo II y no se reproducen aquí. Los umbrales de la Tabla 25 constituyen metas de planificación de carácter referencial, sujetas a validación empírica y no compromisos fijos de resultado.

Tabla 25

Métricas, propósito y umbrales referenciales por objetivo específico

Objetivo	Métricas y ecuación	Propósito	Umbral referencial
OE1	Exactitud, F1-score (Ec. 9), CER (Ec. 7) y ExpRate (Ec. 8).	Reconocimiento simbólico.	F1-score $\geq$ 0.90; CER $\leq 10\,\%$
OE2	IoU (Ec. 10) y exactitud estructural.	Segmentación y preservación de la jerarquía.	IoU ≥ 0.80
OE3	Proporción de pasos correctamente clasificados; κ de Cohen (Ec. 11) y MAE (Ec. 12).	Aporte diferencial de la validación del procedimiento y concordancia con el juicio docente.	Mejora significativa frente a la línea base; $\kappa \geq 0.75$

Consideraciones Éticas y Legales

El estudio trata dos clases de datos con exigencias distintas. Los conjuntos públicos se emplean conforme a los términos de sus licencias de uso académico, que se verifican y documentan antes de su incorporación. La muestra propia, en cambio, procede de personas identificables y activa obligaciones adicionales.

La muestra propia es resuelta por escritores adultos voluntarios. Esta decisión metodológica evita el tratamiento de datos de menores de edad y reduce el procedimiento a un consentimiento informado individual, en el que se declaran la finalidad académica del estudio, el uso previsto de las imágenes y el derecho a retirarse en cualquier momento sin consecuencia alguna. El tratamiento de los datos se ajusta a la Ley Orgánica de Protección de Datos Personales del Ecuador, que reconoce los principios de finalidad, minimización y confidencialidad. En consecuencia, las hojas se anonimizan suprimiendo nombres e identificadores antes de su digitalización y sustituyéndolos por un código sin correspondencia pública; las imágenes se almacenan cifradas y con acceso restringido al investigador; los datos se emplean solo con fines académicos, sin transferencia a terceros ni publicación que permita la reidentificación.

Síntesis del Capítulo

Este capítulo definió cómo se producirá la evidencia del estudio. El enfoque es cuantitativo, porque los objetivos se formularon con métricas y umbrales, mientras el diseño es experimental – comparativo, ya que la pregunta de investigación exige comparar arquitecturas bajo condiciones controladas, con un componente de concordancia que contrasta el prototipo con el juicio docente. Los datos provienen de dos fuentes complementarias: los conjuntos públicos permiten comparar los resultados con la literatura, mientras que la muestra propia aporta los procedimientos multipaso que aquellos no contienen y sin los cuales no podría evaluarse el OE3.

Los resultados sustentarán tres (3) decisiones: qué arquitectura y qué configuración de segmentación adoptar en el prototipo; si validar el procedimiento mejora la concordancia con el docente lo suficiente para justificar su incorporación; y si el desempeño alcanzado respalda el desarrollo de una herramienta de apoyo a la corrección. Con ello se cierra el ciclo de coherencia de la investigación: el problema motiva las preguntas, las preguntas se convierten en objetivos medibles, el marco teórico justifica las variables y sus métricas, y la metodología aquí declarada produce la evidencia que permitirá responderlas.

Capítulo IV: Resultados y Análisis

Introducción del Capítulo

El Capítulo III declaró cómo se produciría la evidencia; este capítulo la presenta y la interpreta. La exposición sigue el mismo orden de los objetivos específicos: primero el reconocimiento simbólico y la comparación de arquitecturas (OE1), después la segmentación que preserva la jerarquía espacial (OE2) y finalmente la validación del procedimiento y su concordancia con el docente evaluador (OE3). Cada apartado presenta las métricas comprometidas en la Tabla 25, sus intervalos de confianza y las pruebas pareadas que permiten decidir si una diferencia observada es atribuible al diseño o al azar.

En este contexto conviene anticipar el criterio con el que se leen los resultados. Un experimento no fracasa cuando no alcanza un umbral referencial; fracasa cuando no permite decidir. Los umbrales de la Tabla 25 se declararon como metas de planificación sujetas a validación empírica, y en este capítulo se contrastan con lo efectivamente obtenido, indicando en cada caso si la distancia entre meta y resultado se explica por el diseño del experimento, por el presupuesto de cómputo o por una limitación real del enfoque. Todas las cifras que siguen proceden de una única corrida del pipeline experimental documentado en el correspondiente Anexo , ejecutado de extremo a extremo con la semilla y la configuración que se declaran a continuación.

Condiciones de la Corrida Experimental

La totalidad de los resultados de este capítulo se obtuvo de la ejecución del cuaderno `assets/pipeline/pipeline_tesis.ipynb` —para entrenar modelos se necesita gran capacidad de hardware con gran capacidad de vram o servicios Cloud de paga—, cuyo registro de entorno y tiempos queda almacenado en `assets/pipeline/runs.json`. La Tabla 26 resume esas condiciones, que son las que un tercero necesita para reproducir el experimento.

Tabla 26

Entorno de ejecución, hiperparámetros y tiempos de la corrida experimental

Elemento	Valor declarado
Entorno	Python 3.12.10; PyTorch 2.12.1 con CUDA 13.0
Hardware	GPU local NVIDIA de equipo portátil, sin infraestructura de cómputo dedicada
Semilla aleatoria	42, idéntica para las tres arquitecturas
Preprocesado	Escala de grises, inversión de tinta, redimensionado a 64 × 512 píxeles preservando la relación de aspecto
Aumento de datos	Rotación en ±2° y cizallamiento en ±0.10, solo en entrenamiento
Presupuesto	20 épocas por arquitectura, con un tope de seguridad de 15 minutos por arquitectura (45 minutos en total) que no llegó a activarse; lote de 32; precisión mixta
Partición efectiva	8 322 expresiones de entrenamiento y 471 de validación, separadas por escritor
Vocabulario	113 entradas: 109 tokens del conjunto de entrenamiento y 4 símbolos especiales
Tiempo de cómputo	31.0 min de entrenamiento para las tres arquitecturas (9.4 / 10.2 / 11.3 min en A/B/C), 37.6 s de evaluación, 0.5 s en el OE2 y 0.2 s en el OE3

Estas condiciones deben leerse a la luz de la restricción de recursos computacionales declarada como limitación en el Capítulo I. El experimento se ejecutó íntegramente en una única GPU local de equipo portátil, sin infraestructura de cómputo dedicada, con un presupuesto total de 31.0 minutos de entrenamiento repartidos entre las tres arquitecturas, una sola semilla por configuración y sin búsqueda de hiperparámetros. Esa restricción no invalida la comparación —las tres arquitecturas recibieron exactamente el mismo presupuesto, la misma inicialización y el mismo protocolo, que es lo que el OE1 comprometía—, pero sí acota de manera decisiva los valores absolutos alcanzables. Los umbrales de F1-score y CER fijados en la Tabla 25 se tomaron de resultados publicados que se obtienen entrenando durante días sobre

varias unidades de procesamiento gráfico, y no son alcanzables en este régimen. Todas las métricas que siguen deben interpretarse, por tanto, como el desempeño de cada arquitectura *bajo una restricción de cómputo severa y declarada de antemano*, y no como el techo de la familia arquitectónica a la que pertenece.

Precisiones Sobre las Configuraciones

A continuación se resumen tres (3) precisiones sobre esta configuración:

- El tope de 15 minutos por arquitectura nunca se activó: las tres completaron sus 20 épocas y sumaron 31.0 min de entrenamiento muy por debajo del máximo teórico de 45 min (3×15 min) que habría resultado si las tres hubieran agotado el tope. El desglose por modelo fue de 9.4, 10.2 y 11.3 min en A/B/C, procede del registro de ejecución, mientras que `runs.json` conserva únicamente el tiempo total agregado.

- El filtro de longitud responde a una exigencia formal de la función de pérdida CTC, que requiere al menos $2L + 1$ pasos temporales para alinear una etiqueta de longitud L; con $T = 128$ columnas a la salida del codificador, el límite es exactamente 63 y se fijó en 60 por seguridad. Ese filtro afecta a 41 de las 8 834 expresiones de entrenamiento esto es, al 0.46 %.

- El filtro se aplica únicamente al entrenamiento es decir las particiones oficiales se evalúan completas sin descartar ninguna expresión para no inflar artificialmente las métricas reportadas.

Caracterización del Dataset

Antes de comparar modelos conviene conocer el material sobre el que operan. La Tabla 27 caracteriza las cuatro particiones empleadas y las Figuras 10, 11 respectivamente, muestran la distribución de longitudes y los símbolos más frecuentes.

Tabla 27

Caracterización de las particiones del dataset CROHME

Conjunto	Expres.	Long. media	Long. mediana	Long. P95	Long. máxima	Símbolos totales
Entrenamiento	8834	15.6	13	38	97	137733
Prueba 2014	986	16.2	13	41	204	15987
Prueba 2016	1147	17.8	14	44	113	20385
Prueba 2019	1199	16.7	14	39	90	20075

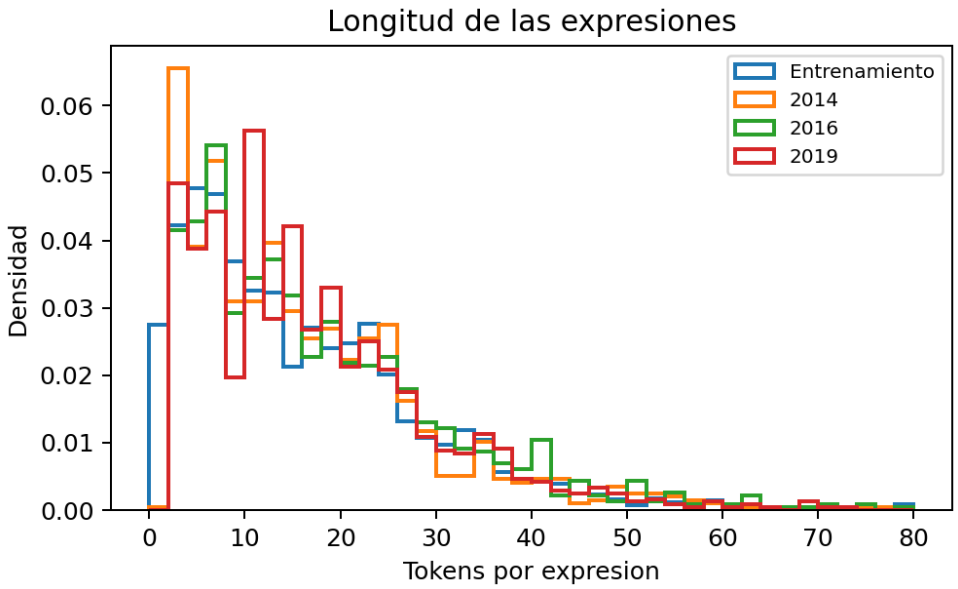


Figura 10

Distribución de la longitud de las expresiones

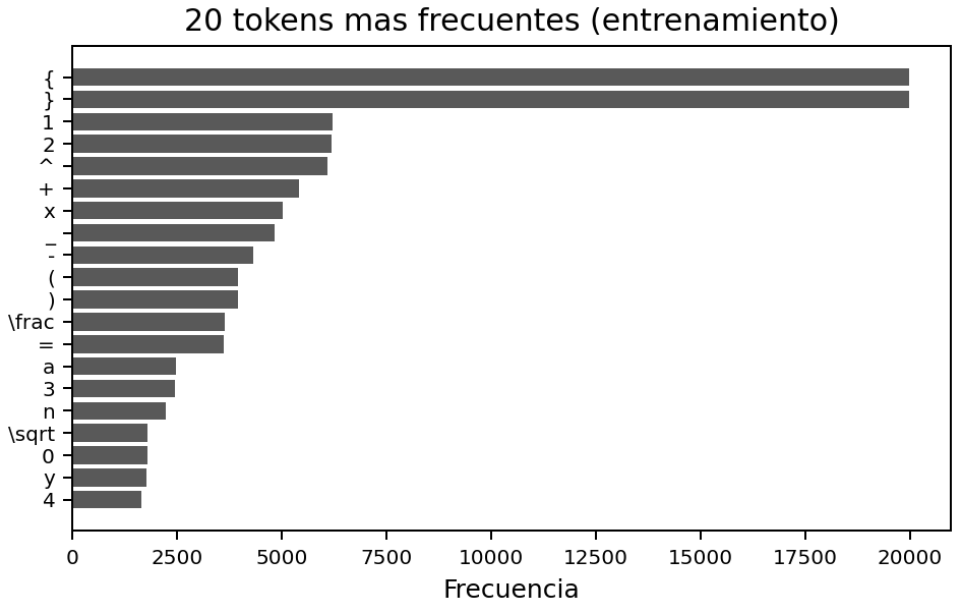


Figura 11

Tokens más frecuentes del conjunto de entrenamiento

Los datos revelan tres rasgos que condicionan todo lo que sigue. El dataset es de tamaño moderado: 8 834 expresiones de entrenamiento y 137 733 símbolos, lejos de los órdenes de magnitud con los que se entrenan hoy los modelos de secuencia. Las expresiones son cortas —mediana de 13 tokens, media de 15.6— pero con una cola larga que alcanza los 97 tokens en entrenamiento y los 204 en la partición de 2014. Y el vocabulario es reducido y estable: 109 tokens distintos, con un único símbolo fuera de vocabulario (\Pi) en la partición de 2014 y ninguno en las de 2016 y 2019, lo que descarta que las diferencias entre conjuntos se expliquen por símbolos no vistos.

La partición por escritor prescrita en el Capítulo III se implementó extrayendo el identificador del escritor del prefijo del nombre de archivo, lo que produjo 3 067 escritores para entrenamiento y 341 para validación. Conviene señalar desde ahora que esta partición controla la caligrafía pero no el contenido de las expresiones, distinción que resultará decisiva para interpretar el comportamiento del decodificador Transformer.

Resultados del OE1: Comparación de Arquitecturas

Las tres arquitecturas comparten codificador convolucional, preprocesado, partición, semilla, optimizador y presupuesto de épocas. Lo único que cambia es la

forma de decodificar la secuencia: proyección lineal por columna en el modelo A, red recurrente bidireccional de dos capas en el modelo B y decodificador Transformer autorregresivo de tres capas con entropía cruzada en el modelo C. La Tabla 28 presenta el desempeño sobre las tres particiones.

Tabla 28

Desempeño de las tres arquitecturas sobre las particiones oficiales de CROHME

Arquitectura	Parámetr. (M)	Conjunto	Exact.	F1-Score	CER	ExpRate
A. CNN+CTC	0.480	CROHME 2014	0.574	0.385	0.488	0.037
B. CRNN+CTC	3.080	CROHME 2014	0.644	0.480	0.394	0.137
C. Transformer	3.610	CROHME 2014	0.538	0.310	0.653	0.033
A. CNN+CTC	0.480	CROHME 2016	0.611	0.433	0.452	0.051
B. CRNN+CTC	3.080	CROHME 2016	0.673	0.550	0.360	0.153
C. Transformer	3.610	CROHME 2016	0.527	0.292	0.659	0.022
A. CNN+CTC	0.480	CROHME 2019	0.637	0.387	0.441	0.033
B. CRNN+CTC	3.080	CROHME 2019	0.705	0.538	0.335	0.158
C. Transformer	3.610	CROHME 2019	0.560	0.290	0.679	0.022

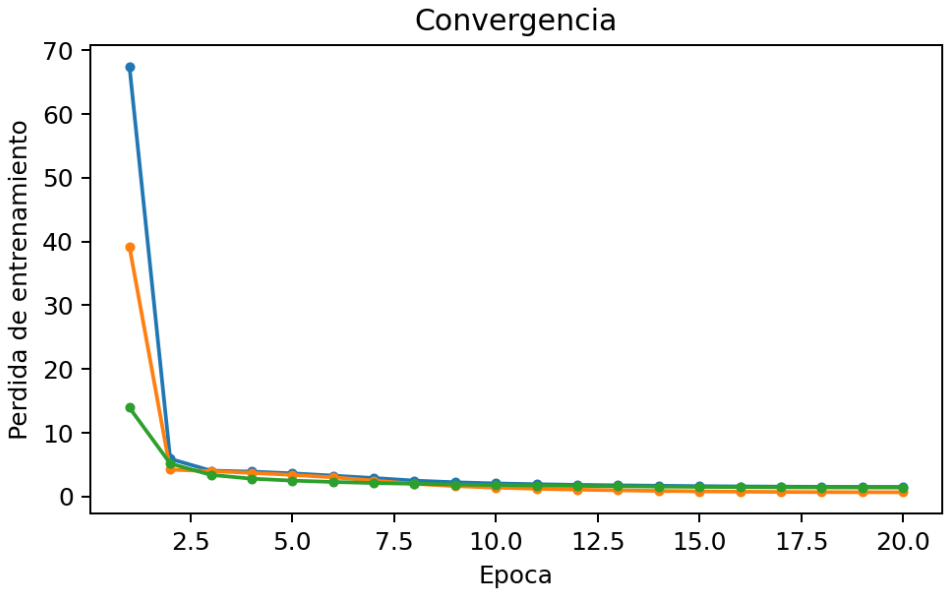


Figura 12

Convergencia de la pérdida de entrenamiento

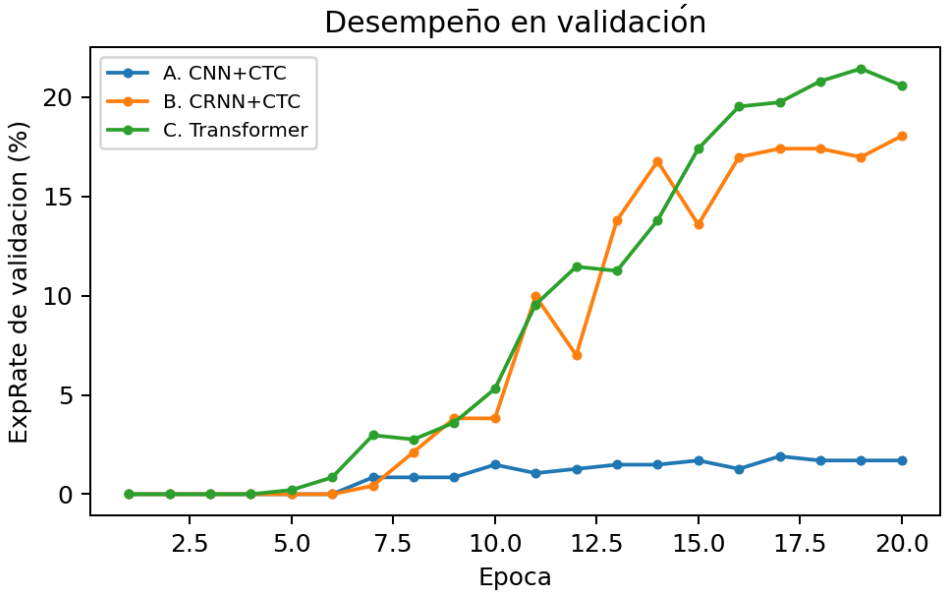


Figura 13

Evolución del ExpRate de validación de las tres arquitecturas

La lectura de las Figuras 12 y 13 exige una advertencia: las curvas de pérdida de los modelos A y B corresponden a la función CTC y la del modelo C a la entropía cruzada con suavizado de etiquetas, de modo que sus magnitudes no son comparables entre sí. Lo comparable es la forma —las tres convergen de manera estable, sin

oscilaciones ni divergencias— y, sobre todo, la Figura 13, donde el ExpRate de validación sí se mide de idéntica manera para las tres.

El resultado central es inequívoco: la arquitectura B (CRNN con CTC) domina en las tres particiones y en las cuatro métricas. Alcanza un ExpRate de 13.7 %, 15.3 % y 15.8 % en CROHME 2014, 2016 y 2019, frente a 3.7 %, 5.1 % y 3.3 % del modelo A y a 3.3 %, 2.2 % y 2.2 % del modelo C. La ventaja se replica en el CER, donde B obtiene 0.394, 0.360 y 0.335 frente a valores entre 0.44 y 0.49 en A y entre 0.65 y 0.68 en C. La Tabla 29 muestra que los intervalos de confianza al 95 % obtenidos por remuestreo *bootstrap* no se solapan entre B y las otras dos arquitecturas en ninguna de las tres particiones.

Tabla 29

Intervalos de confianza por remuestreo bootstrap (2.000 repeticiones)

Arquitectura	Conjunto	ExpRate	ExpRate IC95	CER	CER IC95
A. CNN+CTC	CROHME 2014	0.038	[0.026, 0.050]	0.488	[0.472, 0.505]
B. CRNN+CTC	CROHME 2014	0.137	[0.116, 0.158]	0.394	[0.373, 0.415]
C. Transformer	CROHME 2014	0.032	[0.021, 0.044]	0.653	[0.632, 0.674]
A. CNN+CTC	CROHME 2016	0.051	[0.038, 0.063]	0.451	[0.439, 0.463]
B. CRNN+CTC	CROHME 2016	0.153	[0.133, 0.174]	0.360	[0.346, 0.375]
C. Transformer	CROHME 2016	0.022	[0.014, 0.031]	0.659	[0.642, 0.677]
A. CNN+CTC	CROHME 2019	0.033	[0.023, 0.043]	0.441	[0.429, 0.453]
B. CRNN+CTC	CROHME 2019	0.158	[0.137, 0.178]	0.335	[0.321, 0.349]
C. Transformer	CROHME 2019	0.023	[0.014, 0.032]	0.679	[0.662, 0.698]

Tabla 30

Prueba de McNemar entre pares de arquitecturas sobre las mismas expresiones

Conjunto	Comparación	Solo A	Solo B	p	Significativa (p<0.05)
CROHME 2014	A. CNN+CTC vs B. CRNN+CTC	15	113	0.0000	si
CROHME 2014	A. CNN+CTC vs C. Transformer	33	28	0.6089	no
CROHME 2014	B. CRNN+CTC vs C. Transformer	117	14	0.0000	si
CROHME 2016	A. CNN+CTC vs B. CRNN+CTC	16	133	0.0000	si
CROHME 2016	A. CNN+CTC vs C. Transformer	52	19	0.0001	si
CROHME 2016	B. CRNN+CTC vs C. Transformer	165	15	0.0000	si
CROHME 2019	A. CNN+CTC vs B. CRNN+CTC	11	161	0.0000	si
CROHME 2019	A. CNN+CTC vs C. Transformer	34	22	0.1409	no
CROHME 2019	B. CRNN+CTC vs C. Transformer	177	15	0.0000	si

La prueba de McNemar sobre las predicciones pareadas confirma lo anterior como se menciona en el Capítulo III. Como se aprecia en la Tabla 30, la superioridad de B sobre A y sobre C es estadísticamente significativa en las tres particiones ($p < 0{,}0001$ en los seis contrastes). La comparación entre A y C, en cambio, no permite declarar un ganador: la diferencia no es significativa en 2014 ($p = 0{,}609$) ni en 2019 ($p = 0{,}141$), y en 2016 resulta significativa a favor de A ($p = 0{,}0001$). Dicho de otro modo, el Transformer

no solo no supera a la red recurrente: tampoco supera de manera consistente a la línea base convolucional más simple, que tiene siete veces menos parámetros.

Contraste entre Validación y Prueba

La Figura 13 contiene un hecho que contradice la tabla de resultados y que merece explicación. Sobre la partición de validación, el modelo C es el mejor de los tres: alcanza un ExpRate de 21.4 %, por encima del 18.1 % de B y muy por encima del 1.9 % de A. Sobre las particiones oficiales, ese mismo modelo C es el peor. La inversión completa del orden entre validación y prueba no es un detalle: si el ajuste de hiperparámetros se hubiera guiado por la validación —como es práctica habitual y como el propio protocolo prescribe—, se habría seleccionado la arquitectura que peor generaliza.

La explicación se encontró midiendo el solapamiento de contenido entre particiones, cuyo resultado recoge la Tabla 31.

Tabla 31

Solapamiento de contenido entre el conjunto de entrenamiento y los conjuntos de evaluación

Conjunto	Expresiones	Cadena vista en entrenamiento
Validación (escritores retenidos)	471	0.643
CROHME 2014	986	0.030
CROHME 2016	1147	0.015
CROHME 2019	1199	0.003

El 64.3 % de las expresiones de validación tiene una cadena de tokens idéntica a alguna expresión del conjunto de entrenamiento, frente al 3.0 %, 1.5 % y 0.3 % en las particiones de 2014, 2016 y 2019. El motivo es estructural del dataset: CROHME fue construido pidiendo a muchas personas que escribieran el mismo repertorio de expresiones, de manera que las 8 322 expresiones de entrenamiento contienen apenas

4 636 cadenas distintas. La partición por escritor separa correctamente las caligrafías —garantía que el Capítulo III buscaba— pero no separa el inventario de expresiones. Son dos formas distintas de contaminación, y controlar la primera no controla la segunda.

La Tabla 32 descompone el ExpRate según esa distinción y confirma la hipótesis.

Tabla 32

ExpRate según si la cadena objetivo aparece en el conjunto de entrenamiento

Arquitectura	Conjunto	ExpRate cadena vista	n vistas	ExpRate cadena nueva	n nuevas
A. CNN+CTC	Validación	0.017	303	0.024	168
A. CNN+CTC	CROHME 2014	0.067	30	0.037	956
B. CRNN+CTC	Validación	0.149	303	0.238	168
B. CRNN+CTC	CROHME 2014	0.400	30	0.129	956
C. Transformer	Validación	0.307	303	0.048	168
C. Transformer	CROHME 2014	0.333	30	0.023	956

El modelo C acierta el 30.7 % de las expresiones de validación cuya cadena ya había visto en entrenamiento y solo el 4.8 % de las que son nuevas: una razón de 6.4 a 1. En la partición de 2014 mantiene ese patrón (33.3 % frente a 2.3 %), pero allí las cadenas conocidas son apenas 30 de 986, de modo que su ventaja casi no interviene en la métrica global. El modelo B no exhibe ese comportamiento en validación —acierta incluso más sobre cadenas nuevas (23.8 %) que sobre conocidas (14.9 %)— y el modelo A no muestra diferencia apreciable. La conclusión es directa: el desempeño superior del Transformer en validación proviene en buena medida de haber memorizado el repertorio de expresiones, no de leer mejor la imagen. Es exactamente el riesgo que introduce un decodificador autorregresivo, que lleva incorporado un modelo de lenguaje sobre la secuencia de salida, frente a un decodificador CTC anclado columna a columna a la evidencia visual.

Resultados del OE2: Segmentación y Jerarquía Espacial

La evaluación del pipeline de segmentación exige cajas de referencia por símbolo, dato que las imágenes preprocesadas de CROHME no contienen. La verdad de campo se derivó por ello de los archivos InkML originales de las ediciones 2011 y 2012, que anotan mediante `traceGroup` qué trazos componen cada símbolo; al renderizar los trazos a imagen se conoce la transformación aplicada y la caja de cada símbolo queda determinada de forma exacta. Se trata del mismo dataset: cambia la fuente del dato, no el conjunto. Sobre 300 expresiones así reconstruidas se obtuvieron 3 206 símbolos de referencia.

Siguiendo la variable independiente *configuración del pipeline* de la Tabla 23, se contrastaron dos configuraciones: la segmentación por componentes conectados sin más y la misma segmentación con una regla de fusión que reúne los componentes que forman un mismo símbolo de varios trazos, como el signo igual, la letra i o el símbolo de división. La Tabla 33 presenta el resultado y la Figura 14 lo ilustra cualitativamente.

Tabla 33

Calidad de la segmentación frente a la verdad de campo derivada de los archivos InkML

Configuración	Símbolos	IoU medio	IoU IC95	IoU mediano	% IoU ≥ 0.50	% IoU ≥ 0.75	Exactitud estructural
Sin fusión (componentes conectados)	3206	0.854	[0.847, 0.862]	0.944	0.894	0.849	0.907
Con fusión de componentes	3206	0.829	[0.819, 0.839]	0.945	0.872	0.836	0.763

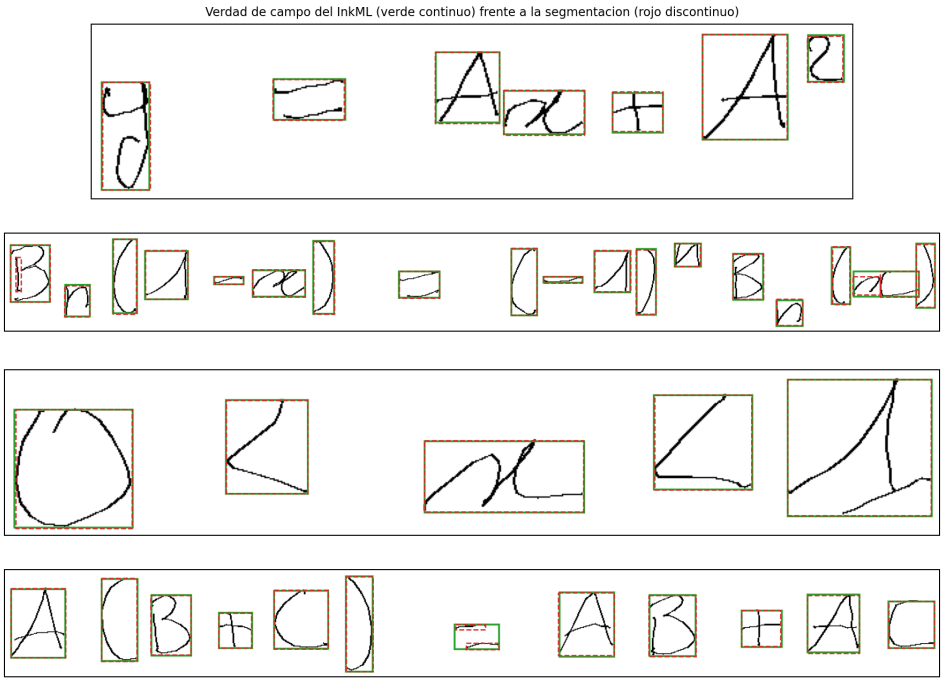


Figura 14

Cajas de referencia derivadas del InkML (verde continuo) frente a las regiones detectadas por el pipeline (rojo discontinuo)

El pipeline sin fusión alcanza un IoU medio de 0.854, con intervalo de confianza al 95 % de [0.847, 0.862] y mediana de 0.944; el 89.4 % de los símbolos supera el umbral de 0.50 y el 84.9 % supera el de 0.75. La exactitud estructural —proporción de expresiones en las que el orden izquierda – derecha de las regiones detectadas reproduce el de la referencia— se sitúa en 0.907. Conviene precisar que esta métrica valida el orden de lectura horizontal —un chequeo unidimensional—, mientras que la fidelidad de la jerarquía bidimensional (posición vertical de superíndices, subíndices y fracciones) queda capturada por el IoU, que compara cada región en el plano completo. El umbral referencial del Capítulo III, IoU $\geq$ 0.80, queda por tanto satisfecho.

El hallazgo inesperado está en la comparación entre configuraciones: la fusión de componentes no mejora la segmentación, la degrada. El IoU medio baja de 0.854 a 0.829 y la exactitud estructural cae de 0.907 a 0.763, una pérdida de más de catorce puntos. La distancia entre la media (0.829) y la mediana (0.945) de la configuración con fusión indica el mecanismo: la regla acierta en la mayoría de los casos pero produce un número

apreciable de fusiones catastróficas, en las que dos símbolos vecinos e independientes quedan absorbidos en una sola caja. Como la regla opera sobre el solapamiento horizontal y la altura relativa de los componentes, tiende a unir precisamente lo que en notación matemática está verticalmente alineado por razones estructurales —una fracción y su numerador, una base y su exponente— y no solo los trazos de un mismo símbolo.

Resultados del OE3: Validación del Procedimiento

El módulo de validación opera sobre `assets/propia/anotacion.json`, transcripción de los ejercicios de la muestra propia que quedan dentro del alcance declarado en la Tabla 4: cinco pruebas, 49 ejercicios —25 de tipo expresión y 24 de tipo ecuación— y 101 transiciones entre líneas consecutivas. La prueba q6, de cálculo integral de nivel universitario, y los ítems de gráficas y geometría dibujada quedaron excluidos por estar fuera de alcance.

Una precisión metodológica sobre el punto de partida. El módulo se alimenta de la transcripción y no de la salida del reconocedor. La decisión es deliberada: la pregunta secundaria 3 indaga en qué medida la validación del procedimiento mejora la calificación frente a evaluar solo el resultado final, y alimentar el módulo con la salida de un reconocedor cuyo CER se sitúa en 0.335 mezclaría ese efecto con el error de reconocimiento hasta volver el resultado ininterpretable. Es justamente la fragilidad que advierte Chaowicharat y Dejdumrong (2023) y que la Tabla 20 recoge como limitación de los antecedentes. Partir de la transcripción aísla la variable de interés; el precio, que se declara sin rodeos, es que el OE3 evalúa el módulo simbólico y no el sistema integrado de extremo a extremo.

Tabla 34

Validación paso a paso frente a la línea base que solo evalúa el resultado final

Modelo	Exactitud	IC95	Ejercicios
Línea base (solo resultado final)	1.000	[1.000, 1.000]	49
Sistema (resultado + validación paso a paso)	1.000	[1.000, 1.000]	49
Pasos correctamente clasificados	1.000	[1.000, 1.000]	101

Sobre las 101 transiciones, el módulo clasificó correctamente el 100 %, con intervalo de confianza al 95 % de [1.000, 1.000]. Señaló como inválidos tres pasos —el primero del ejercicio P1-E3a2, el segundo del P4-E7 y el primero del P4-E8—, y los tres coinciden con los errores identificados por la revisión humana. De las 98 transiciones restantes, el motor las declaró equivalentes por identidad algebraica en 43 casos, por simplificación de expresiones en 39 y por multiplicación de ambos miembros por una constante en 16, lo que muestra que la verificación no se reduce a una comparación textual sino que absorbe transformaciones algebraicas legítimas.

Este resultado muestra dos advertencias de transparencia que deben acompañar cualquier lectura de la Tabla 34. La primera es que la etiqueta de referencia del paso 2 del ejercicio P4-E7 fue corregida *después* de que el motor simbólico señalara el error —el estudiante escribe $12 \times 6 = 36$ en lugar de 72— que la revisión manual inicial no había detectado. Esa transición no constituye, por tanto, una verificación independiente del módulo, y el 100 % debe leerse como 100 sobre 100 verificaciones independientes más una que no lo es. La segunda es que 101 transiciones procedentes de cinco pruebas constituyen una muestra pequeña: el intervalo de confianza degenerado [1.000, 1.000] no significa certeza, sino ausencia de error observado en una muestra de ese tamaño.

Tabla 35

Calificación por prueba: sistema, línea base y docente (aprob.=aprobado / reprob.=reprobado)

Prueba	Ejerc.	Docente	Sistema	Línea base	Nota sistema	Nota línea base	Nota de referencia
P1	17	aprob.	aprob.	aprob.	9.410	9.410	9.410
P2	4	aprob.	aprob.	aprob.	10.000	10.000	10.000
P3	4	reprob.	reprob.	reprob.	5.000	5.000	5.000
P4	9	aprob.	aprob.	aprob.	7.780	7.780	7.780
P5	15	aprob.	aprob.	aprob.	10.000	10.000	10.000

Tabla 36

Concordancia con el docente evaluador medida con el coeficiente kappa de Cohen y el error absoluto medio

Comparación	kappa	MAE	n
Docente A vs Docente B (criterio-docente.md)	1.000	—	5
Sistema vs docente	1.000	0.000	5
Línea base vs docente	1.000	0.000	5

En la concordancia con el docente evaluador, tanto el sistema como la línea base reprodujeron las cinco decisiones de aprobado y reprobado, con κ de Cohen igual a 1.000 y error absoluto medio de 0.000 sobre la nota. Conforme a la escala de Landis y Koch (1977) reproducida en la Tabla 18, ese valor corresponde a concordancia casi perfecta y supera el umbral referencial de 0.75. También aquí corresponde declarar las salvedades: el documento `criterio-docente.md` registra cuatro pruebas aprobadas y una reprobada pero no indica cuál, de modo que la asignación empleada —P3 como

prueba reprobada, por ser la única con ítems sin responder— es una inferencia marcada como tal en la anotación y pendiente de confirmación con el docente. Además, con $n = 5$ pruebas el intervalo de confianza del κ no es estimable, razón por la cual la métrica principal del OE3 es la proporción de pasos correctamente clasificados y no el coeficiente de concordancia.

Análisis

Por Qué Gana la Arquitectura Recurrente

El resultado del OE1 admite una lectura errónea que conviene descartar de entrada; no muestra que las arquitecturas Transformer sean inferiores para el reconocimiento de expresiones matemáticas manuscritas. La literatura revisada en el Capítulo II documenta lo contrario, con BTTR (Zhao et al., 2021) y CoMER (Zhao y Gao, 2022) superando el estado del arte con decodificadores Transformer. Lo que este experimento muestra es algo más acotado y, para los fines del proyecto de investigación, más útil: bajo un presupuesto de 20 épocas y 15 minutos por arquitectura sobre una única GPU, la arquitectura que mejor aprovecha ese presupuesto es la red convolucional – recurrente con CTC.

La explicación es coherente con el marco teórico. El decodificador CTC impone un alineamiento entre columnas de la imagen y tokens de salida; esa restricción es una desventaja para la notación bidimensional —en $\frac{a}{b}$ el numerador y el denominador ocupan la misma franja de columnas— pero es también un sesgo inductivo muy fuerte, que reduce drásticamente el espacio de hipótesis y permite aprender con pocos datos y pocas épocas. El decodificador autorregresivo con atención no impone esa restricción y por eso puede en principio, representar la estructura bidimensional; pero debe aprender el alineamiento desde cero, lo que exige más datos, más épocas y como muestra la literatura, técnicas adicionales como el entrenamiento bidireccional o el modelado de cobertura, ninguna de las cuales se empleó aquí. La comparación de la Tabla 28 es válida como comparación bajo protocolo idéntico —que es lo que el OE1 comprometía— y no como comparación del potencial máximo de cada familia arquitectónica.

También la brecha frente a la literatura se explica por esa vía. El mejor ExpRate

obtenido, 15.8 % en CROHME 2019, está muy por debajo del 46.55 % que Zhang et al. (2017) reportan para WAP en 2014. La diferencia no es de dos o tres puntos, que podría atribuirse a detalles de implementación, sino de un factor cercano a tres, coherente con la distancia entre un presupuesto de 31 minutos de entrenamiento total y los regímenes de entrenamiento con varias GPU durante días que sostienen los resultados publicados. Los umbrales referenciales del Capítulo III para el OE1 —F1-score ≥ 0.90 y CER $\leq$ 10 %— no se alcanzan y, con este presupuesto, no eran alcanzables; conviene decirlo con claridad y no presentarlo como un hallazgo sobre las arquitecturas.

Resultados Sobre la Segmentación

El resultado del OE2 admite una lectura constructiva. Que la fusión de componentes degrade el IoU medio y, sobre todo, la exactitud estructural, indica que la heurística fue formulada con un criterio puramente geométrico —solapamiento horizontal y altura relativa— sobre un dominio en el que la alineación vertical no es ruido sino significado. En notación matemática, dos componentes verticalmente alineados pueden ser los dos trazos de un signo igual, pero también un numerador y un denominador, o una base y su exponente. Una regla que no distingue esos casos destruye jerarquía justamente donde el OE2 se propone preservarla.

El resultado sostiene entonces dos afirmaciones. La primera es que el pipeline de segmentación cumple el objetivo con la configuración simple, con IoU de 0.854 sobre el umbral de 0.80 y exactitud estructural de 0.907. La segunda es que la reconstrucción de símbolos de varios trazos requiere información estructural y no solo geométrica —por ejemplo, la clase del símbolo candidato o el árbol de disposición al que alude la Figura 8 del Capítulo II—, y que abordarla con una heurística de solapamiento es insuficiente. Es un resultado negativo bien delimitado, que informa el diseño del prototipo con más precisión que una confirmación trivial.

Resultados de los Algoritmos frente a la Línea Base

La comparación con una línea base se plantea en dos planos distintos del estudio:

- En el plano del reconocimiento (OE1), la línea base es la arquitectura A, la red convolucional con proyección lineal por columna, que representa el enfoque más

simple capaz de producir una secuencia. Frente a ella, la arquitectura B multiplica el ExpRate por 3.7, 3.0 y 4.8 en las particiones de 2014, 2016 y 2019, y reduce el CER en 19 %, 20 % y 24 % relativos. La prueba de McNemar declara esas diferencias significativas en los tres conjuntos, con recuentos de pares discordantes muy asimétricos: en CROHME 2019, por ejemplo, 161 expresiones son acertadas solo por B frente a 11 acertadas solo por A. La arquitectura C, en cambio, no supera a la línea base de manera consistente y en la partición de 2016 es significativamente peor que ella. El costo de la mejora de B es un modelo 6.4 veces mayor, de 3.08 millones de parámetros frente a 0.48 millones, dato que la Tabla 28 reporta explícitamente para que la comparación sea interpretable en términos de recursos y no solo de exactitud.

- En el plano de la calificación (OE3), la línea base es conceptualmente distinta: es el modelo que califica únicamente el resultado final, que es la práctica que el Capítulo I identifica como limitación del proceso actual. Y aquí el resultado es que ambos sistemas coinciden. Los dos aciertan las 49 decisiones de la muestra, los dos alcanzan $\kappa = 1{,}000$ frente al docente y la prueba de McNemar entre ellos no encuentra un solo par discordante ($p = 1{,}000$).

El aporte del módulo de validación del procedimiento, en esta muestra, no consiste en cambiar la decisión de aprobado o reprobado, sino en **localizar** el paso exacto donde el procedimiento se rompe: información que la línea base es estructuralmente incapaz de producir y que constituye el insumo de la retroalimentación al estudiante, que es el problema que motivó la investigación. La pregunta secundaria 3, formulada como «en qué medida mejora la calidad de la calificación», recibe entonces una respuesta de dos partes: en la decisión agregada, no la mejora en esta muestra; en la capacidad diagnóstica, la mejora es cualitativa y no admite comparación con la línea base porque esta no produce esa información en absoluto.

Interpretación de los Hallazgos

La Tabla 37 confronta cada umbral referencial declarado en la Tabla 25 con el resultado efectivamente obtenido:

Tabla 37

Contraste entre los umbrales referenciales del Capítulo III y los resultados obtenidos

Objetivo	Umbral referencial	Resultado obtenido	Lectura
OE1	F1-score $\geq$ 0.90; CER $\leq$ 10 %	Mejor modelo (B): F1-score 0.550 en CROHME 2016 y CER 33.5 % en CROHME 2019	No alcanzado. Atribuible al presupuesto de cómputo y no al diseño comparativo, que sí discrimina entre arquitecturas con significancia estadística.
OE2	IoU $\geq$ 0.80	IoU 0.854 [0.847, 0.862]; exactitud estructural 0.907	Alcanzado con la configuración sin fusión de componentes.
OE3	Mejora significativa frente a la línea base; $\kappa \geq 0.75$	100 % de pasos clasificados correctamente; $\kappa = 1.000$; sin pares discordantes frente a la línea base	Parcialmente alcanzado: el κ supera el umbral, pero la mejora frente a la línea base no es contrastable en una muestra sin errores de procedimiento con resultado correcto.

Los tres resultados son concluyentes en distinto grado. El OE1 se cumple como comparación aunque no alcance los valores absolutos comprometidos. El OE2 se cumple en sentido pleno: el pipeline supera el umbral de IoU y además la comparación entre configuraciones aporta evidencia sobre qué estrategia de segmentación conviene adoptar en el prototipo. El OE3 se cumple en la implementación y en la concordancia.

Frente al marco teórico, los hallazgos dialogan con tres de sus componentes. El enfoque arquitectónico del Capítulo II, que documentó la progresión de WAP a BTTR y CoMER, queda matizado por la evidencia de que esa progresión es contingente al presupuesto de entrenamiento: en el régimen de bajos recursos que caracteriza a esta investigación —y que caracterizaría también a una eventual implementación en una institución educativa sin infraestructura de cómputo— el orden de méritos se invierte. Y el enfoque evaluativo, que fundamenta el contraste con el juicio docente, se cumple en su parte de concordancia pero revela que la muestra necesaria para contrastar el aporte de la validación del procedimiento debe diseñarse de otro modo: no basta recolectar pruebas reales, hay que asegurar que contengan los casos en los que ambos sistemas pueden discrepar.

Finalmente, respecto de la brecha de investigación de las Tablas 19, 20, el estudio aporta el sistema integrado que la literatura no ofrecía y confirma empíricamente la fragilidad que Chaowicharat y Dejdumrong (2023) anticipaba: con un CER de 0.335, encadenar el módulo de validación a la salida del reconocedor haría inservible la verificación simbólica.

Síntesis del Capítulo

Los experimentos produjeron tres resultados principales. Primero, bajo protocolo idéntico y presupuesto acotado, la arquitectura convolucional – recurrente con CTC supera de manera estadísticamente significativa tanto a la línea base convolucional como al decodificador Transformer en las tres particiones oficiales de CROHME, aunque ninguna de las tres alcanza los umbrales absolutos de F1-score y CER declarados como referencia. Segundo, el pipeline de segmentación supera el umbral de IoU comprometido y además, la comparación entre configuraciones muestra que la fusión geométrica de componentes degrada la exactitud estructural en más de catorce puntos. Tercero, el módulo de validación del procedimiento clasifica correctamente la totalidad de las transiciones de la muestra y concuerda plenamente con el docente evaluador, pero no puede demostrarse superior a la línea base porque la muestra no contiene los casos en los que ambos podrían discrepar.

Capítulo V: Conclusiones y Recomendaciones

Introducción del Capítulo

A lo largo de los Capítulos I, II, III y IV se produjo la evidencia, misma que se la interpretó caso por caso; por consiguiente en este Capítulo V se cierra con los respectivos hallazgos. Primero se concluye sobre cada objetivo específico, indicando en cada caso si el objetivo se cumplió, con qué evidencia y con qué reservas. Después se responde de manera integrada a la pregunta principal de investigación, se declaran las limitaciones que acotan el alcance de esas respuestas y se formulan las recomendaciones para trabajos futuros. Adicional se relaza un apartado final, de carácter más aplicado que investigativo, señalando hacia dónde se orientaría una eventual herramienta de uso docente.

Conclusión sobre el OE1: Comparar Arquitecturas

El OE1 se cumple. Se compararon tres arquitecturas —una red convolucional con proyección lineal, una convolucional – recurrente con pérdida CTC y un codificador – decodificador Transformer— bajo un protocolo verdaderamente idéntico: mismo codificador, mismo preprocesado misma partición, misma semilla, mismo optimizador y el mismo presupuesto de veinte (20) épocas. Cualquier diferencia observada es por construcción, atribuible a la arquitectura.

La arquitectura convolucional – recurrente con CTC, de forma clara y en las tres particiones oficiales de CROHME. Multiplica por tres (3) o por cuatro (4) el ExpRate de la línea base convolucional, reduce el CER alrededor de un veinte (20 %) por ciento relativo y la prueba de McNemar respalda esas diferencias con $p < 0{,}0001$ en los seis (6) contrastes.

Ninguna de las tres alcanzó los umbrales absolutos que el Capítulo III había fijado como referencia: F1-score de 0.90 y CER del 10 %; puesto que el mejor modelo se quedó en 0.550 y 33.5 % respectivamente. No es un fracaso del diseño experimental, es el precio de un presupuesto de treinta y un (31) minutos de entrenamiento total en una sola GPU, frente a los regímenes de varios días o semanas y múltiples aceleradores que sostienen los resultados publicados. Dicho de otro modo: el experimento respondió muy

bien la pregunta que se le hizo —cuál de las tres aprovecha mejor un presupuesto limitado— y no puede responder una que nunca se le hizo, que es cuál sería el techo de cada arquitectura con recursos ilimitados.

El Transformer fue el mejor de los tres en la partición de validación, con un ExpRate del 21.4 %, por encima de la red recurrente. En las particiones oficiales se desplomó al 2 – 3 %. Al investigar por qué, apareció algo que trasciende este experimento: el 64.3 % de las expresiones de validación tenía una cadena de tokens idéntica a alguna del entrenamiento, frente a menos del 3 % en las particiones oficiales. La partición por escritor —que el Capítulo III había prescrito con buen criterio— separa caligrafías, pero no separa el repertorio de expresiones. Un decodificador autorregresivo lleva dentro un modelo de lenguaje, y lo que hizo fue, en buena medida, recordar. Acertaba el 30.7 % de las cadenas ya vistas y solo el 4.8 % de las nuevas.

En resumen, en régimen de bajos recursos conviene el sesgo inductivo fuerte del CTC recurrente. Por otro, y quizá más importante si el ajuste de hiperparámetros se hubiera guiado por la validación, se habría elegido justamente el modelo que peor generaliza. La decisión metodológica de reservar las particiones oficiales para un único uso final no fue un formalismo; fue lo que evitó una conclusión equivocada.

Conclusión sobre el OE2: Preservar la Jerarquía Espacial

El OE2 se cumple y esta vez también en los valores absolutos. El pipeline de segmentación alcanzó un IoU medio de 0.854, con un intervalo de confianza de [0.847, 0.862] sobre 3 206 símbolos, por encima del umbral de 0.80 comprometido en el Capítulo III. La exactitud estructural llegó a 0.907. La verdad de campo se construyó a partir de los archivos InkML originales de CROHME, que anotan qué trazos componen cada símbolo, así que las cajas de referencia no son una aproximación: están determinadas de forma exacta.

La regla de fusión de componentes pensada para reunir los dos trazos de un signo igual, el punto y el cuerpo de una *i*, las piezas de un símbolo de división no mejoró nada. Al contrario, se evidenció un descenso del IoU de 0.854 a 0.829 y hundió la exactitud estructural de 0.907 a 0.763. ($\approx$ Catorce puntos).

La regla decide fusionar por criterios puramente geométricos: solapamiento horizontal y altura relativa. Y en notación matemática la alineación vertical no es ruido, es significado. Un numerador sobre un denominador está verticalmente alineado. Una base y su exponente, también. La heurística terminaba uniendo precisamente aquello que el objetivo pretendía mantener separado y jerarquizado.

Conclusión sobre el OE3: Validar el Procedimiento

El módulo de verificación simbólica clasificó correctamente las 101 transiciones entre líneas consecutivas de los 49 ejercicios en alcance. El 100 %. Y no por comparación textual, que sería trivial, sino absorbiendo transformaciones algebraicas legítimas. Señaló tres (3) pasos como inválidos y los tres coinciden con los errores que había identificado la revisión humana. La concordancia con el docente evaluador fue total: κ de Cohen de 1.0, muy por encima del umbral de 0.75.

Lo que no pudo demostrarse es que el sistema con validación paso a paso sea mejor que la línea base que solo mira el resultado final. Ambos acertaron las 49 decisiones. La prueba de McNemar no encontró un solo par discordante. Y no es que el módulo falle; es aritmética. El sistema solo puede apartarse de la línea base cuando un ejercicio tiene la respuesta final correcta y algún paso intermedio inválido —por compensación de errores, por copiar el resultado, por un desarrollo incoherente que llega de casualidad al valor correcto—. En estas cinco pruebas no hubo ni un caso así. Sin pares discordantes no hay nada que contrastar, y ninguna prueba estadística puede inventarlo.

El proyecto de investigación sí aporta, pero no donde se lo buscó. Aporta en **localizar** el paso exacto donde se rompe el procedimiento. La línea base no puede producir esa información ni en principio: solo sabe decir «mal», no sabe decir «mal aquí, en el segundo renglón, porque se escribió que 12 X 6 = 36». Y esa segunda frase es la retroalimentación que el Capítulo I identificó como ausente en la corrección manual masiva.

Limitaciones

Algunas limitaciones ya se habían anticipado en el Capítulo I; otras aparecieron durante la ejecución y son francamente las más interesantes:

- Presupuesto de cómputo: 31 minutos de entrenamiento en total para las tres arquitecturas (entre 9 y 12 minutos cada una), dentro de un presupuesto fijo de 20 épocas y un tope de seguridad de 15 minutos por arquitectura que no llegó a activarse. Los umbrales de F1-score y CER del Capítulo III no eran alcanzables en ese régimen. La comparación entre arquitecturas sigue siendo válida, pero los valores absolutos no son comparables con los publicados en la literatura.

- Sesgo caligráfico y alcance: CROHME no contiene escritura ecuatoriana y la muestra propia fue resuelta por adultos voluntarios. Los resultados no son extrapolables sin más a la caligrafía de estudiantes adolescentes ni al sistema educativo local.

- Cobertura del motor simbólico: La verificación se limita a lo que SymPy resuelve y un procedimiento válido escrito de forma no canónica podría marcarse como incorrecto; en esta muestra no ocurrió, pero la muestra es pequeña.

- Alcance de la exactitud estructural: la métrica del OE2 valida el orden de lectura horizontal (izquierda–derecha) de las regiones detectadas, un chequeo unidimensional. La fidelidad de la jerarquía bidimensional —posición vertical de superíndices, subíndices y fracciones— queda evidenciada por el IoU sobre cada región, pero no por una comprobación relacional explícita entre pares de símbolos.

Recomendaciones

- Ampliar el presupuesto antes que cambiar de arquitectura es decir subir de 20 a 100 – 200 épocas ó mas y aumentar la altura de entrada de 64 a 128 píxeles son las dos palancas más baratas. Los modelos aquí entrenados tienen entre 0.5 y 3.6 millones de parámetros.

- Reintentar el Transformer con las técnicas que la literatura ya validó en entrenamiento bidireccional al estilo de BTTR (Zhao et al., 2021) y modelado de cobertura al estilo de CoMER (Zhao y Gao, 2022). La hipótesis a contrastar es que el decodificador autorregresivo necesita esas muletas para aprender el alineamiento con pocos datos y que sin ellas el CTC recurrente es preferible.
- Explorar el preentrenamiento con MathWriting (Gervais et al., 2025) o HASYv2 (Thoma, 2017), identificados en la revisión y no explotados aquí. Rasterizar los trazos de MathWriting multiplicaría en mas el volumen de entrenamiento disponible.
- Complementar la exactitud estructural con una métrica relacional que clasifique la relación espacial entre pares de símbolos (derecha, superíndice, subíndice, sobre, bajo, dentro) y la contraste contra la referencia, aprovechando las anotaciones `.lg` de CROHME no explotadas aquí. Permitiría verificar la preservación de la jerarquía 2D de forma directa, más allá del orden de lectura.
- Recolectar deliberadamente ejercicios con procedimiento inválido y resultado final correcto. Sin esos casos, el aporte de la validación paso a paso es indemostrable por construcción, por buena que sea la implementación.
- Ampliar la muestra de concordancia a varias decenas de pruebas y a dos o más docentes evaluadores independientes, de modo que el κ tenga intervalo de confianza y se pueda estimar además la concordancia entre humanos como techo realista del sistema.

Hacia una Herramienta Utilizable

Un comentario final por parte del tesista, más de oficio que de investigación. Este trabajo produjo un prototipo de investigación, no un producto, y así se declaró en el alcance. Pero si algo dejó claro el OE3 es dónde estaría el valor de una eventual herramienta, no en poner la nota sino en señalar el renglón exacto donde el razonamiento se torció. Esa es la información que un docente con cientos de estudiantes no alcanza a producir a mano y es la que convierte una calificación en retroalimentación.

Así mimo, quien retome este trabajo haría bien en empezar por ahí, es decir por un flujo en el que el docente sube la hoja, el sistema propone la nota y sobre todo marca los pasos dudosos para que sea el docente quien decida. Menos automatización total y más asistencia con criterio, esto podría sonar menos ambicioso pero es bastante más útil.

Referencias

Arroyo Hernández, J., Travieso González, C. M., Ticay Rivas, J., Mora Mora, F., Salas Huertas, O., Ramírez Bogantes, M., y Sánchez Chaves, L. (2013). Sistema de detección y clasificación automática de granos de polen mediante técnicas de procesado digital de imágenes. *Uniciencia*, *27*(1), 59–73.

Bahdanau, D., Cho, K., y Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. En *International conference on learning representations (iclr).*

Bian, X., Qin, B., Xin, X., Li, J., Su, X., y Wang, Y. (2022). Handwritten mathematical expression recognition via attention aggregation based bi-directional mutual learning. En *Proceedings of the AAAI conference on artificial intelligence* (Vol. 36, pp. 113–121).

Chaowicharat, E., y Dejdumrong, N. (2023). A step toward an automatic handwritten homework grading system for mathematics. *Information Technology and Control*, *52*(1), 169–184.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, *20*(1), 37–46.

Deng, Y., Kanervisto, A., Ling, J., y Rush, A. M. (2017). Image-to-markup generation with coarse-to-fine attention. En *Proceedings of the 34th international conference on machine learning (icml).*

Descartes, R. (1637). *Discurso del método* (R. Frondizi, Traduc.). Madrid: Alianza Editorial.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., y Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) challenge. *International Journal of Computer Vision*, *88*(2), 303–338.

Gervais, P., Fadeeva, A., y Maksai, A. (2025). MathWriting: A dataset for handwritten mathematical expression recognition. En *Proceedings of the 31st ACM SIGKDD conference on knowledge discovery and data mining* (pp. 5459–5469). Association for Computing Machinery. doi: 10.1145/3711896.3737436

Goodfellow, I., Bengio, Y., y Courville, A. (2016). *Deep learning*. Cambridge, MA: MIT Press.

Guan, T., Lin, C., Shen, W., y Yang, X. (2024). PosFormer: recognizing complex handwritten mathematical expression with position forest transformer. En *Computer vision – ECCV 2024* (Vol. 15080, pp. 130–147). Springer.

Hammack, R. (2018). *Book of proof* (3rd ed ed.). Richmond, VA: Virginia Commonwealth University.

He, K., Zhang, X., Ren, S., y Sun, J. (2016). Deep residual learning for image recognition. En *Proceedings of the ieee conference on computer vision and pattern recognition (cvpr)* (pp. 770–778).

Hernández Sampieri, R., y Mendoza Torres, C. P. (2023). *Metodología de la investigación: las rutas cuantitativa, cualitativa y mixta* (2a ed ed.). Ciudad de México: McGraw-Hill.

Hochreiter, S., y Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Krizhevsky, A., Sutskever, I., y Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. En *Advances in neural information processing systems* (Vol. 25).

Landis, J. R., y Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, *33*(1), 159–174.

LeCun, Y., Bengio, Y., y Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

LeCun, Y., Bottou, L., Bengio, Y., y Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Liu, T., Chatain, J., Kobel-Keller, L., Kortemeyer, G., Willwacher, T., y Sachan, M. (2026). AI-assisted automated short answer grading of handwritten university-level mathematics exam. *Teaching Mathematics and its Applications*, *45*(1), 84–105. doi: 10.1093/teamat/hrag010

Mahdavi, M., Zanibbi, R., Mouchère, H., Viard-Gaudin, C., y Garain, U. (2019).

ICDAR 2019 CROHME + TFD: Competition on recognition of handwritten mathematical expressions and typeset formula detection. En *2019 international conference on document analysis and recognition (icdar)* (pp. 1533–1538). IEEE.

Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., ... Scopatz, A. (2017). SymPy: symbolic computing in Python. *PeerJ Computer Science*, *3*, e103.

Mouchère, H., Viard-Gaudin, C., Zanibbi, R., y Garain, U. (2016). ICFHR2016 CROHME: Competition on recognition of online handwritten mathematical expressions. En *15th international conference on frontiers in handwriting recognition (icfhr).*

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., ... Moher, D. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, *372*, n71.

Parsaeifard, B., Hlosta, M., y Bergamin, P. (2025). *Automated grading of students' handwritten graphs: a comparison of meta-learning and vision-large language models.* Preprint de arXiv. Descargado de `https://arxiv.org/abs/2507.03056` (arXiv:2507.03056. Preprint sin revisión por pares) doi: 10.48550/arXiv.2507.03056

Pólya, G. (1945). *How to solve it: a new aspect of mathematical method.* Princeton, NJ: Princeton University Press.

Ronneberger, O., Fischer, P., y Brox, T. (2015). U-Net: convolutional networks for biomedical image segmentation. En *Medical image computing and computer-assisted intervention (miccai 2015)* (Vol. 9351, pp. 234–241). Springer.

Rumelhart, D. E., Hinton, G. E., y Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

Sabino, C. A. (1996). *El proceso de investigación.* Buenos Aires: Editorial Lumen-Humanitas.

Siemens, G., y Long, P. (2011). Penetrating the fog: analytics in learning and education. *EDUCAUSE Review*, *46*(5), 30–40.

Sutskever, I., Vinyals, O., y Le, Q. V. (2014). Sequence to sequence learning with neural networks. En *Advances in neural information processing systems* (Vol. 27, pp. 3104–3112).

Thoma, M. (2017). *The HASYv2 dataset.* Preprint de arXiv. Descargado de `https://arxiv.org/abs/1701.08380` (arXiv:1701.08380. Preprint sin revisión por pares) doi: 10.48550/arXiv.1701.08380

Truong, T.-N., Nguyen, C. T., Zanibbi, R., Mouchère, H., y Nakagawa, M. (2024). A survey on handwritten mathematical expression recognition: the rise of encoder-decoder and GNN models. *Pattern Recognition*, *153*, 110531.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. En *Advances in neural information processing systems* (Vol. 30).

Velleman, D. J. (2019). *How to prove it: a structured approach* (3rd ed ed.). Cambridge: Cambridge University Press.

Wirth, R., y Hipp, J. (2000). CRISP-DM: towards a standard process model for data mining. En *Proceedings of the 4th international conference on the practical application of knowledge discovery and data mining* (pp. 29–39). Practical Application Company.

Wu, J.-W., Yin, F., Zhang, Y.-M., Zhang, X.-Y., y Liu, C.-L. (2020). Handwritten mathematical expression recognition via paired adversarial learning. *International Journal of Computer Vision*, *128*, 2386–2401.

Xie, Y., Mouchère, H., Simistira Liwicki, F., Rakesh, S., Saini, R., Nakagawa, M., ... Truong, T.-N. (2024). A review of handwritten mathematical expression recognition. *Pattern Recognition*, *155*, 110654.

Yuan, Y., Liu, X., Dikubab, W., Liu, H., Ji, Z., Wu, Z., y Bai, X. (2022). Syntax-aware network for handwritten mathematical expression recognition. En *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (cvpr)* (pp. 4543–4552).

Zanibbi, R., y Blostein, D. (2012). Recognition and retrieval of mathematical

expressions. *International Journal on Document Analysis and Recognition*, *15*(4), 331–357.

Zhang, J., Du, J., y Dai, L. (2018). Multi-scale attention with dense encoder for handwritten mathematical expression recognition. En *24th international conference on pattern recognition (icpr)*.

Zhang, J., Du, J., Zhang, S., Liu, D., Hu, Y., Hu, J., ... Dai, L. (2017). Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition. *Pattern Recognition*, *71*, 196–206.

Zhao, W., y Gao, L. (2022). CoMER: modeling coverage for transformer-based handwritten mathematical expression recognition. En *Computer vision – ECCV 2022* (Vol. 13688, pp. 392–408). Springer.

Zhao, W., Gao, L., Yan, Z., Peng, S., Du, L., y Zhang, Z. (2021). Handwritten mathematical expression recognition with bidirectionally trained transformer. En *International conference on document analysis and recognition (icdar)*.

Anexo A: Scripts y Código Fuente

Organización del Repositorio

Todo el experimento del Capítulo IV se ejecuta desde un único cuaderno de Jupyter, sin dependencias de servicios externos ni de estado previo. La Tabla 38 describe la estructura de archivos del proyecto en lo relativo a la parte computacional.

Tabla 38

Estructura de archivos del pipeline experimental

Ruta	Contenido
`assets/pipeline/pipeline_tesis.ipynb`	Cuaderno único que produce todas las tablas y figuras del Capítulo IV.
`assets/pipeline/tables/*.tex`	Diez tablas en formato LaTeX, listas para `\input`, con `booktabs` y etiqueta `Tab:`.
`assets/pipeline/figures/*.png`	Tres figuras a 180 ppp, copiadas además a `images/` para el `\graphicspath`.
`assets/pipeline/checkpoints/*.pt`	Pesos de la mejor época de cada arquitectura (A, B y C), seleccionada por ExpRate de validación.
`assets/pipeline/runs.json`	Registro de la corrida: entorno, versiones, semilla, hiperparámetros y tiempos.
`assets/crohme/data/`	Corpus CROHME en modalidad *offline* (OE1). No se versiona con la tesis.
`assets/crohme/ICFHR_package/`	Archivos InkML con anotación de segmentación (OE2). No se versiona con la tesis.
`assets/propia/`	Muestra propia: imágenes, `anotacion.json` y `criterio-docente.md` (OE3).

Estructura del Cuaderno

El cuaderno consta de forma general de 30 celdas organizadas en 6 secciones, cuya correspondencia con los objetivos específicos es directa. La Tabla 39 las resume.

Tabla 39

Secciones del cuaderno experimental y su correspondencia con los objetivos

Sección	Título	Contenido y salidas
0	Configuración	Semilla, dispositivo, rutas, hiperparámetros y registro del entorno.
1	Datos y análisis exploratorio	Lectura de `caption.txt`, vocabulario, partición por escritor, Tabla 27 y Figuras 10 y 11.
2	OE1: comparación de arquitecturas	Conjunto de datos, codificador común, tres modelos, entrenamiento, evaluación y Tablas 28, 29, 30, 31 y 32.
3	OE2: segmentación	Lectura de InkML, renderizado con cajas exactas, segmentación, IoU y Tabla 33.
4	OE3: validación del procedimiento	Verificación simbólica, comparación con la línea base y Tablas 34, 35 y 36.
5	Exportación	Escritura de las tablas en LaTeX, copia de figuras a `images/` y registro en `runs.json`.

Fragmentos de Código Relevantes

Se reproducen los fragmentos que definen las métricas comprometidas en los objetivos o que sostienen una afirmación metodológica del Capítulo IV; el resto del código se encuentra en el cuaderno. Los pasajes omitidos dentro de un fragmento se indican con la marca `# [...]`. La Tabla 40 indica qué resultado sostiene cada uno.

Tabla 40

Correspondencia entre los fragmentos de código y los resultados que sostienen

Código	Resultado que sostiene
1	Semilla, hiperparámetros y entorno de toda la corrida.
2	Partición sin coincidencia de escritores y Tabla 27.
3	Igualdad de condiciones del OE1 y columna de parámetros de la Tabla 28.
4	Tabla 28.
5	Tablas 29 y 30.
6	Verdad de campo de la Tabla 33.
7	Tabla 33 y Figura 14.
8	Columnas de IoU y exactitud estructural de la Tabla 33.
9	Tabla 34.
10	Tablas 35 y 36.

Configuración Común

El Código 1 fija la semilla y los hiperparámetros que comparten las tres arquitecturas, y registra el entorno de ejecución que se vuelca en `runs.json`.

```
# [...] importacion de las bibliotecas y resolucion de las rutas del proyecto
SMOKE_TEST = True      # True = 200 muestras y 1 epoca, para verificar el cableado
SEED       = 42        # semilla unica, identica para las tres arquitecturas (Cap. III
    )
#
def fijar_semilla(s=SEED):
    random.seed(s); np.random.seed(s)
    torch.manual_seed(s); torch.cuda.manual_seed_all(s)
    torch.backends.cudnn.deterministic = True
    torch.backends.cudnn.benchmark = False
fijar_semilla()
DEVICE  = torch.device("cuda" if torch.cuda.is_available() else "cpu")
USA_AMP = DEVICE.type == "cuda"
# ---------------------------------------- hiperparametros (identicos para las 3)
IMG_H, IMG_W = 64, 512     # tras el codificador quedan T = 128 pasos temporales
MAX_LEN      = 60          # P99 de la longitud es 63; T = 128 >= 2*60+1 exige el CTC
BATCH        = 32
EPOCAS         = 1 if SMOKE_TEST else 20
MIN_POR_MODELO = 1 if SMOKE_TEST else 15   # tope de minutos por arquitectura
```

```
LR             = 3e-4
LIMITE         = 200 if SMOKE_TEST else None    # corta el dataset en modo humo
ENTORNO = {
    "python": platform.python_version(),
    "torch": torch.__version__,
    "cuda": torch.version.cuda,
    "gpu": torch.cuda.get_device_name(0) if DEVICE.type == "cuda" else "CPU",
    "semilla": SEED, "epocas": EPOCAS, "batch": BATCH, "img": [IMG_H, IMG_W],
    "smoke_test": SMOKE_TEST,
}
```

Código 1: Semilla, hiperparámetros comunes y registro del entorno

OE1: Comparación de Arquitecturas

El Código 2 construye el vocabulario y separa entrenamiento de validación *por escritor*, no por expresión. Ninguna de las manos que aparecen en validación se ha visto durante el entrenamiento, de modo que la métrica de validación no premia el haber memorizado una caligrafía. El mismo fragmento documenta el descarte de las expresiones más largas que `MAX_LEN`, exigido por la condición $T \geq 2L + 1$ de la pérdida CTC.

```
PAD, SOS, EOS, UNK = 0, 1, 2, 3          # PAD comparte indice con el blank del CTC
ESPECIALES = ["<pad/blank>", "<sos>", "<eos>", "<unk>"]
tokens_train = Counter(t for _, toks in datos["train"] for t in toks)
VOCAB = ESPECIALES + sorted(tokens_train)
TOK2ID = {t: i for i, t in enumerate(VOCAB)}
V = len(VOCAB)
#
def codificar(toks):  return [TOK2ID.get(t, UNK) for t in toks]
def decodificar(ids): return [VOCAB[i] for i in ids if i > EOS]
# ------------------------------------------------------ particion por escritor
def escritor(nombre):
    # 'formulaire032-equation014' -> 'formulaire032';  '200924-1331-216' -> '200924'
    return nombre.split("-equation")[0] if "-equation" in nombre else nombre.split("-
        ")[0]
#
escritores = sorted({escritor(n) for n, _ in datos["train"]})
rng = np.random.RandomState(SEED); rng.shuffle(escritores)
corte = int(0.9 * len(escritores))
ESC_TRAIN, ESC_VAL = set(escritores[:corte]), set(escritores[corte:])
#
train_full = [d for d in datos["train"] if escritor(d[0]) in ESC_TRAIN]
```

```
val_full   = [d for d in datos["train"] if escritor(d[0]) in ESC_VAL]
#
# El CTC exige T >= 2L+1; se descartan del ENTRENAMIENTO las expresiones mas largas.
train_full = [d for d in train_full if len(d[1]) <= MAX_LEN]
val_full   = [d for d in val_full   if len(d[1]) <= MAX_LEN]
```

Código 2: Vocabulario y partición por escritor (OE1)

El Código 3 es la pieza que hace comparable al OE1. Las tres arquitecturas comparten el mismo codificador convolucional, el mismo preprocesado, la misma partición y la misma inicialización de partida —`construir()` vuelve a llamar a `fijar_semilla()` antes de instanciar cada modelo—, de manera que la única diferencia entre A, B y C es el decodificador. La diferencia de tamaño resultante se reporta en la Tabla 28 para que la comparación sea interpretable.

```
class Codificador(nn.Module):
    # Comun a las tres arquitecturas. Entrada 1x64x512 -> salida 256x4x128 (T = 128).
    def __init__(self, c=256):
        super().__init__()
        def bloque(ci, co, pool):
            return nn.Sequential(nn.Conv2d(ci, co, 3, padding=1), nn.BatchNorm2d(co),
                                 nn.ReLU(inplace=True), nn.MaxPool2d(pool))
        self.red = nn.Sequential(
            bloque(1, 32, (2, 2)),      # 32 x 256
            bloque(32, 64, (2, 2)),     # 16 x 128
            bloque(64, 128, (2, 1)),    #  8 x 128
            bloque(128, c, (2, 1)),     #  4 x 128
        )
    def forward(self, x): return self.red(x)
#
class ModeloCTC(nn.Module):
    # A (recurrente=False) y B (recurrente=True)
    def __init__(self, vocab, recurrente):
        super().__init__()
        self.cod = Codificador()
        self.recurrente = recurrente
        if recurrente:
            self.rnn = nn.LSTM(256, 256, num_layers=2, bidirectional=True,
                               batch_first=True)
            self.sal = nn.Linear(512, vocab)
        else:
            self.sal = nn.Sequential(nn.Linear(256, 256), nn.ReLU(inplace=True),
                                     nn.Linear(256, vocab))
```

```
28
29      def forward(self, x):
30          h = self.cod(x).mean(dim=2)           # colapso vertical -> (B, C, T)
31          h = h.permute(0, 2, 1)                # (B, T, C)
32          if self.recurrente:
33              h, _ = self.rnn(h)
34          return self.sal(h)                    # (B, T, V)
35  #
36  class ModeloTransformer(nn.Module):
37      # C: mismo codificador + decodificador Transformer autorregresivo
38      def __init__(self, vocab, d=256, capas=3, cabezas=4):
39          super().__init__()
40          self.cod = Codificador(d)
41          self.emb = nn.Embedding(vocab, d, padding_idx=PAD)
42          capa = nn.TransformerDecoderLayer(d, cabezas, dim_feedforward=1024, dropout
                =0.1,
43                                            batch_first=True, norm_first=True)
44          self.dec = nn.TransformerDecoder(capa, capas)
45          self.sal = nn.Linear(d, vocab)
46          # [...] codificacion posicional de la memoria y del objetivo
47
48      def forward(self, x, tgt_in, mem=None):
49          mem = self.memoria(x) if mem is None else mem
50          t = tgt_in.size(1)
51          mascara = torch.triu(torch.ones(t, t, device=tgt_in.device, dtype=torch.bool)
                , 1)
52          e = self.pos_tgt(self.emb(tgt_in) * math.sqrt(self.d))
53          h = self.dec(e, mem, tgt_mask=mascara, tgt_key_padding_mask=(tgt_in == PAD))
54          return self.sal(h)
55  #
56  def construir(nombre):
57      fijar_semilla()                       # misma inicializacion de partida para las
            tres
58      if nombre == "A. CNN+CTC":     return ModeloCTC(V, recurrente=False).to(DEVICE)
59      if nombre == "B. CRNN+CTC":    return ModeloCTC(V, recurrente=True).to(DEVICE)
60      if nombre == "C. Transformer": return ModeloTransformer(V).to(DEVICE)
61      raise ValueError(nombre)
```

Código 3: Codificador común y las tres cabezas de decodificación (OE1)

El Código 4 implementa las métricas del OE1 tal como se definieron en el Capítulo II.

```
1  for hip, ref in pares:
2      d = levenshtein(hip, ref)
```

```
    d_tot += d; n_tot += len(ref)
    exactas.append(hip == ref)
    ch, cr = Counter(hip), Counter(ref)
    for c in set(ch) | set(cr):
        t = min(ch[c], cr[c])
        tp[c] += t; fp[c] += max(0, ch[c] - t); fn[c] += max(0, cr[c] - t)
return {"Exactitud": sum(tp.values()) / max(1, n_tot),
        "F1-score": float(np.mean(f1s)) if f1s else 0.0,
        "CER": d_tot / max(1, n_tot),
        "ExpRate": float(np.mean(exactas))}
```

Código 4: Cálculo de exactitud, F1-score, CER y ExpRate (OE1)

El Código 5 implementa la prueba de McNemar sobre los aciertos pareados y el intervalo de confianza por remuestreo.

```
def mcnemar(ok_a, ok_b):
    # b = aciertos solo de A, c = aciertos solo de B; binomial exacta sobre
        discordantes
    b = int(np.sum(ok_a & ~ok_b)); c = int(np.sum(~ok_a & ok_b))
    if b + c == 0:
        return b, c, 1.0
    p = stats.binomtest(b, b + c, 0.5).pvalue
    return b, c, float(p)

def ic_media(v, n=2000, alfa=0.05):
    v = np.asarray(v, dtype=float)
    rng = np.random.RandomState(SEED)
    idx = rng.randint(0, len(v), size=(n, len(v)))
    d = v[idx].mean(axis=1)
    return float(np.percentile(d, 100 * alfa / 2)), float(np.percentile(d, 100 * (1 -
         alfa / 2)))

def ic_razon(num, den, n=2000, alfa=0.05):
    # para el CER, que es una razon de sumas y no una media de medias
    num, den = np.asarray(num, float), np.asarray(den, float)
    rng = np.random.RandomState(SEED)
    idx = rng.randint(0, len(num), size=(n, len(num)))
    d = num[idx].sum(axis=1) / np.maximum(1, den[idx].sum(axis=1))
    return float(np.percentile(d, 100 * alfa / 2)), float(np.percentile(d, 100 * (1 -
         alfa / 2)))
```

Código 5: Prueba de McNemar e intervalos de confianza por remuestreo (OE1)

OE2: Segmentación

El Código 6 obtiene la verdad de campo del OE2. Al renderizar los trazos del InkML se conoce la transformación aplicada, de modo que la caja de cada símbolo queda determinada de forma exacta y no aproximada:

```
def tr(p): return ((p[0] - minx) * esc + margen, (p[1] - miny) * esc + margen)
cajas = []
for etiqueta, refs in simbolos:
    q = [tr(p) for r in refs if r in trazos for p in trazos[r]]
    if not q: continue
    xs2, ys2 = zip(*q)
    cajas.append((etiqueta, (min(xs2) - grosor/2, min(ys2) - grosor/2,
                             max(xs2) + grosor/2, max(ys2) + grosor/2)))
```

Código 6: Derivación de las cajas de referencia a partir del InkML (OE2)

El Código 7 es el método que se mide segmentación y Jerarquía Espacial. Las dos configuraciones son las dos filas de la Tabla 33.

```
def segmentar(img, fusionar=True):
    # binarizacion + componentes conectados; opcionalmente fusiona los componentes
    # que forman un mismo simbolo de varios trazos (=, i, j, \div)
    _, bw = cv2.threshold(img, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
    n, _, stats, _ = cv2.connectedComponentsWithStats(bw, connectivity=8)
    cajas = []
    for i in range(1, n):
        x, y, w, h, area = stats[i]
        if area < 6: continue
        cajas.append([float(x), float(y), float(x + w), float(y + h)])
    if not cajas: return []
    if fusionar:
        alto_exp = max(b[3] for b in cajas) - min(b[1] for b in cajas)
        cambio = True
        while cambio:
            cambio = False
            for i in range(len(cajas)):
                for j in range(i + 1, len(cajas)):
                    a, b = cajas[i], cajas[j]
                    solape = min(a[2], b[2]) - max(a[0], b[0])
                    menor = min(a[2] - a[0], b[2] - b[0])
                    bajos = max(a[3] - a[1], b[3] - b[1]) < 0.35 * alto_exp
                    if menor > 0 and solape / menor > 0.7 and bajos:
                        cajas[i] = [min(a[0], b[0]), min(a[1], b[1]),
                                    max(a[2], b[2]), max(a[3], b[3])]
```

```
                        cajas.pop(j); cambio = True; break
                if cambio: break
    return sorted(cajas, key=lambda b: b[0])
```

Código 7: Segmentación por componentes conectados con fusión de multitrazo (OE2)

El Código 8 define el IoU y el criterio de emparejamiento entre cajas predichas y de referencia.

```
def iou(a, b):
    ix = max(0.0, min(a[2], b[2]) - max(a[0], b[0]))
    iy = max(0.0, min(a[3], b[3]) - max(a[1], b[1]))
    inter = ix * iy
    ua = (a[2]-a[0])*(a[3]-a[1]) + (b[2]-b[0])*(b[3]-b[1]) - inter
    return inter / ua if ua > 0 else 0.0
#
def emparejar(pred, gt):
    # emparejamiento voraz por solapamiento maximo (sin optimizacion global)
    libres = list(range(len(pred)))
    ious, asignacion = [], {}
    for g, caja_gt in enumerate(gt):
        mejor, mejor_i = 0.0, None
        for i in libres:
            v = iou(pred[i], caja_gt)
            if v > mejor: mejor, mejor_i = v, i
        ious.append(mejor)
        if mejor_i is not None and mejor > 0:
            libres.remove(mejor_i); asignacion[g] = mejor_i
    return np.array(ious), asignacion
```

Código 8: Índice de Jaccard y emparejamiento voraz de las cajas (OE2)

OE3: Validación del Procedimiento

El Código 9 es el núcleo del OE3.

```
def equivalentes(l1, l2, tipo):
    # Devuelve (es_valido, motivo). El motivo distingue el error del estudiante
    # del fallo del motor simbolico, que el Capitulo I declara como limitacion.
    try:
        a1, b1 = _parsear(l1)
        a2, b2 = _parsear(l2)
        if tipo == "expresion" or (b1 is None and b2 is None):
            return bool(sp.simplify(a1 - a2) == 0), "expresion"
        d1, d2 = sp.expand(a1 - b1), sp.expand(a2 - b2)
```

```
        if sp.simplify(d1 - d2) == 0:
            return True, "identica"
        r = sp.simplify(d1 / d2) if d2 != 0 else None
        if r is not None and r.is_number and r != 0:
            return True, "multiplo"
        libres = sorted(d1.free_symbols | d2.free_symbols, key=str)
        if libres:
            s1, s2 = sp.solve(d1, libres, dict=True), sp.solve(d2, libres, dict=True)
            if s1 and s2 and s1 == s2:
                return True, "mismo conjunto solucion"
        return False, "no equivalente"
    except Exception as e:
        return False, f"fallo del motor: {type(e).__name__}"
```

Código 9: Verificación de equivalencia entre líneas consecutivas (OE3)

El Código 10 cierra el OE3, es decir convierte la proporción de pasos válidos en una nota sobre la escala de diez puntos del sistema educativo ecuatoriano, aplica el umbral de aprobación y compara la decisión resultante con la del docente mediante el coeficiente kappa de Cohen. La misma operación se aplica a la línea base que califica únicamente el resultado final, y de esa simetría sale la comparación de la Tabla 36. La función `kappa_seguro` atiende el caso degenerado en que ambos evaluadores emiten una sola categoría, para el que el coeficiente no está definido.

```
NOTA_MAXIMA, NOTA_APROBACION = 10.0, 7.0     # escala del sistema educativo
    ecuatoriano

def nota(serie):  return NOTA_MAXIMA * float(np.mean(serie))

def kappa_seguro(a, b):
    # kappa no esta definido si ambos evaluadores emiten una sola categoria
    if len(set(a)) == 1 and len(set(b)) == 1:
        return 1.0 if list(a) == list(b) else 0.0
    return float(cohen_kappa_score(a, b))

# [...] por cada prueba se calculan la nota del sistema, la de la linea base
#       y la de referencia, y se les aplica el umbral NOTA_APROBACION
y_doc  = (tab_oe3_conc["Docente"] == "aprobado").astype(int)
y_sis  = (tab_oe3_conc["Sistema"] == "aprobado").astype(int)
y_base = (tab_oe3_conc["Línea base"] == "aprobado").astype(int)

k_sis, k_base = kappa_seguro(y_doc, y_sis), kappa_seguro(y_doc, y_base)
```

```
mae_sis  = float(np.mean(np.abs(tab_oe3_conc["Nota sistema"]     - tab_oe3_conc["Nota de referencia"])))
mae_base = float(np.mean(np.abs(tab_oe3_conc["Nota línea base"] - tab_oe3_conc["Nota de referencia"])))
```

Código 10: Nota, umbral de aprobación y concordancia con el docente (OE3)

Anexo B: Ficha Técnica de los Conjuntos de Datos

Identificación y Procedencia

La Tabla 41 identifica los tres conjuntos empleados, su procedencia exacta y el objetivo específico al que sirven. Los dos conjuntos públicos no se versionan junto al documento por su volumen —486 MB en disco— pero son reconstruibles de forma determinista desde las direcciones indicadas.

Tabla 41

Identificación, procedencia y función de los conjuntos de datos empleados

Conjunto	Procedencia y ruta local	Modalidad	Objetivo
CROHME *offline*	Distribución `data.zip` del repositorio de CoMER (Zhao y Gao, 2022); se extrae en `assets/crohme/data/`.	Imágenes `.bmp` en escala de grises con etiqueta en tokens LaTeX.	OE1
CROHME InkML	Paquete `CROHME_full_v2.zip`, ediciones ICFHR 2011 y 2012; se extrae en `assets/crohme/ICFHR_package/`.	Trazos vectoriales con anotación `traceGroup` por símbolo.	OE2
Muestra propia	Recolección primaria en condiciones controladas; `assets/propia/`.	Fotografías `.png` y transcripción en `anotacion.json`.	OE3

Licencias, Ética y Disponibilidad

Los conjuntos CROHME se emplean conforme a los términos de uso académico de la competencia, sistematizada por Mouchère et al. (2016) y Mahdavi et al. (2019), con la cita correspondiente en todas las tablas que reportan resultados sobre ellos.