Instituto Internacional de Investigación y Desarrollo Tecnológico Educativo INDTEC, C.A.
DOI: https://doi.org/10.29394/Scientific.issn.2542-2987.2026.11.39.11.150-165
OAI-PMH: http://www.indteca.com/ojs/index.php/Revista_Scientific/oai

CTEM: Sistema de Generación de Grafos de Conocimiento para el Aprendizaje Matemático

Marco Aurelio Guamán Buestán[1], Nancy Maribel Eras Eras[1], Juan Marcelo Pérez Pérez[1], Jonnathan Danilo Vallejo Sacoto[1]
[1]Instituto Tecnológico Superior Sudamericano (ITS), Cuenca, Ecuador
maguaman2@sudamericano.edu.ec | https://orcid.org/0000-0001-9544-2220
nmeras@sudamericano.edu.ec | https://orcid.org/0009-0000-1509-9857
jperez@sudamericano.edu.ec | https://orcid.org/0000-0003-4753-3122
jdvallejo@sudamericano.edu.ec | https://orcid.org/0000-0002-5781-7834
Correspondencia: maguaman2@sudamericano.edu.ec

Resumen: Las dificultades en el aprendizaje matemático, documentadas en PISA para el Desarrollo en países de ingresos bajos y medios, exigen herramientas para visualizar los conceptos y sus relaciones. El estudio presentó CTEM (Connectivism Tool for Mathematics Education), un sistema colaborativo que genera grafos de conocimiento a partir de debates académicos grabados en audio para el aprendizaje matemático, y evaluó la calidad y la seguridad de su primera fase de desarrollo. La investigación fue aplicada, con enfoque cuantitativo, método evaluativo y diseño no experimental transversal. El sistema, con microservicios y procesamiento de lenguaje natural, se evaluó mediante la verificación funcional de su flujo principal, la descripción de un proceso de aseguramiento de la calidad (QA) en cuatro fases y una auditoría de seguridad basada en el Open Web Application Security Project y en el modelo de amenazas STRIDE. El flujo principal se completó de extremo a extremo en la verificación del equipo. La auditoría registró 97 amenazas en seis componentes; 70,1% se clasificó como crítico o alto, y la interfaz de programación de aplicaciones (API) REST concentró el mayor número (37,1%). Information Disclosure (27,5%) y Tampering (25,3%) fueron las categorías STRIDE más frecuentes entre las 91 clasificadas. Se concluye que la arquitectura permitió completar el flujo principal y que el proceso de QA quedó documentado, sin poder valorarse su efecto sobre los defectos; la capa de API y el servicio de transcripción requieren remediación antes de escalar el despliegue, y el impacto pedagógico queda pendiente de pruebas con estudiantes.

Palabras clave: grafos de conocimiento; procesamiento de lenguaje natural; aseguramiento de la calidad; auditoría de seguridad; aprendizaje matemático
Código de clasificación internacional UNESCO: 1203.10 - Enseñanza con ayuda de ordenador.
Clasificación OCDE-FOS: 1.2 - Ciencias de la computación e información; 5.3 - Ciencias de la educación.

CTEM: A Knowledge Graph Generation System for Mathematics Learning

Abstract: Difficulties in learning mathematics, documented by PISA for Development in low- and middle-income countries, call for tools that make concepts and their relationships visible. This study presented CTEM (Connectivism Tool for Mathematics Education), a collaborative system that generates knowledge graphs for mathematics learning from audio-recorded academic discussions, and evaluated the quality and security of its first development phase. The research was applied, with a quantitative approach, an evaluative method, and a non-experimental cross-sectional design. The evaluation of the system, which uses microservices and natural language processing, combined functional verification of its main workflow, the description of a four-phase quality assurance (QA) process, and a security audit based on the Open Web Application Security Project and the STRIDE threat model. In the team's verification, the main workflow was completed end to end. The audit recorded 97 threats across six components; 70.1% were classified as critical or high, and the REST application programming interface (API) accounted for the largest number (37.1%). Information Disclosure (27.5%) and Tampering (25.3%) were the most frequent STRIDE categories among the 91 threats classified. The study concludes that the architecture made it possible to complete the main workflow and that the QA process was documented, although its effect on defects could not be assessed. The API layer and the transcription service require remediation before deployment is scaled up, and the pedagogical impact has yet to be tested with students.

Keywords: knowledge graphs; natural language processing; quality assurance; security audit; mathematics learning
UNESCO International Classification Code: 1203.10 - Computer-assisted instruction.
OECD-FOS Classification: 1.2 - Computer and information sciences; 5.3 - Educational sciences.

CTEM: sistema de geração de grafos de conhecimento para a aprendizagem matemática

Resumo: As dificuldades na aprendizagem matemática, documentadas pelo PISA para o Desenvolvimento em países de renda baixa e média, exigem ferramentas para visualizar os conceitos e suas relações. O estudo apresentou o CTEM (Connectivism Tool for Mathematics Education), um sistema colaborativo que gera grafos de conhecimento para a aprendizagem matemática a partir de debates acadêmicos gravados em áudio, e avaliou a qualidade e a segurança de sua primeira fase de desenvolvimento. A pesquisa foi aplicada, com abordagem quantitativa, método avaliativo e delineamento não experimental transversal. O sistema, que utiliza microsserviços e processamento de linguagem natural, foi avaliado por meio da verificação funcional de seu fluxo principal, da descrição de um processo de garantia da qualidade (QA) em quatro fases e de uma auditoria de segurança baseada no Open Web Application Security Project e no modelo de ameaças STRIDE. O fluxo principal foi concluído de ponta a ponta na verificação realizada pela equipe. A auditoria registrou 97 ameaças em seis componentes; 70,1% foram classificadas como críticas ou altas, e a interface de programação de aplicações (API) REST concentrou o maior número (37,1%). Information Disclosure (27,5%) e Tampering (25,3%) foram as categorias STRIDE mais frequentes entre as 91 ameaças classificadas. Conclui-se que a arquitetura permitiu concluir o fluxo principal e que o processo de QA ficou documentado, sem que fosse possível avaliar seu efeito sobre os defeitos; a camada de API e o serviço de transcrição requerem remediação antes de se ampliar a escala da implantação, e o impacto pedagógico permanece pendente de testes com estudantes.

Palavras-chave: grafos de conhecimento; processamento de linguagem natural; garantia da qualidade; auditoria de segurança; aprendizagem matemática
Código de classificação internacional da UNESCO: 1203.10 - Ensino com auxílio de computador. **Classificação OCDE-FOS:** 1.2 - Ciências da computação e da informação; 5.3 - Ciências da educação.

Cómo citar este artículo:
Guamán, M. A., Eras, N. M., Pérez, J. M., & Vallejo, J. D. (2026). CTEM: Sistema de Generación de Grafos de Conocimiento para el Aprendizaje Matemático. *Revista Scientific, 11*(39), 150–165. https://doi.org/10.29394/Scientific.issn.2542-2987.2026.11.39.11.150-165

Fecha de Recepción:	Fecha de Aceptación:	Fecha de Publicación:
20-05-2025	21-07-2025	05-02-2026

Revista Scientific - Artículo de Investigación Empírica - Arbitrado - Volumen 11, Número 39 - Febrero-Abril 2026 - pp. 150-165
pp. BA2016000002 - e-ISSN: 2542-2987 - ISNI: 0000 0004 6045 0361

DOI: https://doi.org/10.29394/Scientific.issn.2542-2987.2026.11.39.11.150-165
OAI-PMH: http://www.indteca.com/ojs/index.php/Revista_Scientific/oai

1. Introducción

La educación matemática requiere herramientas que permitan visualizar los conceptos y las relaciones que existen entre ellos, de modo que el proceso de enseñanza-aprendizaje gane en comprensión y en significado. El programa PISA para el Desarrollo (PISA-D, por sus siglas en inglés), adaptación del Programa para la Evaluación Internacional de Estudiantes (PISA), fue lanzado en 2014 por la Organización para la Cooperación y el Desarrollo Económicos (OCDE) para que la evaluación resulte más accesible y pertinente en países de ingresos bajos y medios, y contó con la participación de siete países, entre ellos Ecuador (OECD, 2018).

Las cifras que la OCDE presentó en 2019 permiten dimensionar la brecha en matemáticas: mientras que, en promedio, 76% de los estudiantes de los países de la OCDE alcanzó al menos el nivel 2 de competencia, esa proporción fue de 2% en Zambia, uno de los países de PISA-D (OECD, 2019).

A ello se suma un problema de cobertura. En promedio, solo 43% de los jóvenes de 15 años de los países de PISA-D estaba matriculado desde el grado 7 y podía rendir la prueba, frente a 89% en los países de la OCDE (OECD, 2018). Por tanto, los resultados describen únicamente a quienes permanecen en el sistema escolar en los grados evaluados, y la situación de quienes quedan fuera es aún menos conocida. Las pruebas PISA examinan lo que los estudiantes saben en lectura, matemáticas y ciencias y lo que pueden hacer con ese conocimiento (OECD, 2019), por lo que un desempeño bajo en matemáticas plantea la pregunta de cómo apoyar la enseñanza de esta disciplina con recursos que acerquen los conceptos abstractos a la experiencia de los estudiantes.

Una línea de la literatura aborda las dificultades en el aprendizaje matemático mediante estrategias de aprendizaje activo, tanto en la formación docente como en la educación superior. Un estudio con 48 futuros docentes de primaria, que experimentan en sus clases de matemática estrategias de aprendizaje activo como el plegado de papel, el recorrido de galería y el recorrido matemático, parte de que el aprendizaje es una actividad con dimensiones cognitivas, sociales y físicas (Vale & Barbosa, 2023). En el nivel universitario, un curso de cálculo rediseñado para estudiantes de carreras de ciencia, tecnología, ingeniería y matemáticas (STEM) con preparación insuficiente se compara con secciones de enseñanza tradicional, y los estudiantes del modelo activo reportan un cambio positivo en su perspectiva sobre la matemática y un mayor compromiso con la asignatura (Kurepa et al., 2019).

Otra línea de la literatura examina factores como el interés por la asignatura y la disponibilidad de recursos en entornos de aprendizaje mejorado con tecnología. En un estudio preliminar con 40 estudiantes de Malasia, el interés por la matemática es relativamente alto, pero no se correlaciona de forma significativa con el desempeño en el conjunto de la muestra, aunque sí en los estudiantes de menor desempeño, en un entorno de aprendizaje mejorado con tecnología (Wong & Wong, 2019).

Un modelo de ecuaciones estructurales con 157 estudiantes de la Universidad de Ha'il examina, en otro contexto, cómo la disponibilidad de recursos y el interés por la asignatura influyen en la intención de adoptar el aprendizaje mejorado con tecnología (Alshammari, 2023). De ambos trabajos puede inferirse que conviene considerar el interés del estudiante y la disponibilidad de recursos al incorporar tecnología, aunque ninguno mide el efecto de una herramienta digital sobre el aprendizaje.

Una revisión reciente de los últimos cinco años documenta que la tecnología digital se ha incorporado a la enseñanza de la matemática de formas variadas: sistemas de álgebra computacional y paquetes de geometría dinámica, entornos de colaboración en línea y sistemas de inteligencia artificial, estos últimos un campo nuevo que la revisión solo aborda de forma breve (Engelbrecht & Borba, 2024). En educación superior rural, un estudio cualitativo con 20 estudiantes de segundo año de matemática, apoyado en el Modelo de Aceptación de Tecnología, encuentra actitudes positivas

DOI: https://doi.org/10.29394/Scientific.issn.2542-2987.2026.11.39.11.150-165
OAI-PMH: http://www.indteca.com/ojs/index.php/Revista_Scientific/oai

hacia las herramientas digitales, junto con dificultades al aprender con ellas (Gqoli, 2024).

Más allá de los resultados finales, las analíticas de aprendizaje permiten observar el proceso. Una revisión sistemática analiza cómo se conceptualiza el compromiso estudiantil y qué indicadores se obtienen de los registros digitales en la educación superior (Bergdahl et al., 2024); otra examina cómo estas analíticas apoyan el éxito en los estudios (Ifenthaler & Yau, 2020), y una tercera evalúa si los tableros orientados al estudiante siguen siendo predominantemente analíticos o se han vuelto pedagógicamente informados (Paulsen & Lindsay, 2024).

En el ámbito colaborativo, un estudio de diseño y desarrollo con tres rondas de consulta Delphi a cinco expertos deriva 15 principios y 43 pautas para el aprendizaje colaborativo en línea apoyado en analíticas de aprendizaje (Choi & Nam, 2025). En paralelo, como señalan Cerratto y McGrath (2021) en su revisión de 21 artículos publicados entre 2014 y 2019, existen asuntos éticos que acompañan a estas prácticas.

El uso de los dispositivos móviles también tiene un lado problemático. Un metaanálisis de 29 estudios, con 48.490 participantes, encuentra que el uso problemático del teléfono inteligente se asocia de forma negativa y pequeña con el rendimiento académico (Paterna et al., 2024).

A decir de Bell (2011), el conectivismo, propuesto como teoría del aprendizaje para la era digital, se presenta como sucesor del conductismo, el cognitivismo y el constructivismo, teorías con limitaciones, entre ellas la de no abordar el aprendizaje que se localiza en la tecnología y en las organizaciones. Con todo, el propio Bell advierte que, por sí solo, resulta insuficiente para orientar el aprendizaje apoyado en tecnología. Esa perspectiva se refleja en el nombre de la herramienta propuesta en este trabajo, Connectivism Tool for Mathematics Education (CTEM).

Su producto central es un grafo de conocimiento, es decir, una representación estructurada del conocimiento en la que las entidades y sus relaciones se modelan como nodos y conexiones. Una revisión sistemática sobre la construcción y la aplicación de estos grafos en educación los vincula con la personalización del aprendizaje, el diseño curricular, el mapeo de conceptos y los sistemas de recomendación de contenidos (Abu-Salih & Alotaibi, 2024).

Elaborar un grafo exige resolver tareas específicas, como la extracción y la desambiguación de entidades y la extracción de relaciones, descritas en una revisión sobre grafos de conocimiento científico-tecnológicos (Zhou et al., 2020). Además, como los grafos reales evolucionan, su completación debe poder incorporar entidades y relaciones que no aparecieron durante el entrenamiento (Liang et al., 2024), y la alineación de entidades permite emparejar las que aparecen en grafos distintos (Zhu et al., 2024).

Por otra parte, el procesamiento automático del habla se ha aplicado en contextos diversos. Con estudiantes de medicina se emplea una herramienta de reconocimiento del habla y de reconocimiento facial para analizar cómo las características lingüísticas del docente, como la velocidad y el volumen al hablar o el uso de preguntas, se relacionan con la concentración en clase (Chai et al., 2024).

Ante la necesidad de nuevas tecnologías que fortalezcan un proceso de enseñanza centrado en los estudiantes, este trabajo presenta CTEM como un sistema colaborativo de generación de grafos de conocimiento para el aprendizaje matemático: graba el audio de debates académicos, lo transcribe, extrae los conceptos y sus relaciones y construye un grafo que docentes y administradores visualizan. Entre las fuentes revisadas, los estudios de analíticas de aprendizaje sobre el compromiso estudiantil se apoyan sobre todo en registros digitales de actividad (Bergdahl et al., 2024) y el reconocimiento del habla se ha utilizado para analizar clases (Chai et al., 2024); CTEM, en cambio, aplica el reconocimiento del habla al audio de los debates, con el propósito de que docentes y administradores visualicen cómo los estudiantes conectan los conceptos matemáticos. Esta revisión no agota la literatura, por lo

Revista Scientific - Artículo de Investigación Empírica - Arbitrado - Volumen 11, Número 39 - Febrero-Abril 2026 - pp. 150-165
pp. BA2016000002 - e-ISSN: 2542-2987 - ISNI: 0000 0004 6045 0361

DOI: https://doi.org/10.29394/Scientific.issn.2542-2987.2026.11.39.11.150-165
OAI-PMH: http://www.indteca.com/ojs/index.php/Revista_Scientific/oai

que no se descarta la existencia de sistemas semejantes.

Un sistema como CTEM, que graba la voz de los estudiantes, añade exigencias que no son solo pedagógicas. El tratamiento de los datos de voz plantea requisitos de protección de datos cuya aplicación a la captura, el almacenamiento y el procesamiento del habla todavía no se comprende bien (Nautsch et al., 2019), por lo que conviene tratar el audio con especial cautela. Un sistema distribuido, con aplicaciones web y móvil, una interfaz de programación de aplicaciones (API, por sus siglas en inglés) y servicios que procesan el audio, multiplica los puntos de acceso que deben protegerse, y la literatura sobre seguridad en microservicios es amplia, pero dispersa (Berardi et al., 2022).

La calidad y la seguridad del software se abordan con prácticas conocidas. La integración continua se asocia con efectos positivos en el desarrollo de software (Soares et al., 2022) y se ha propuesto como vía para mejorar la calidad del software científico (Schubert & Argent, 2024). En seguridad, el modelo STRIDE (acrónimo de *Spoofing*, *Tampering*, *Repudiation*, *Information Disclosure*, *Denial of Service* y *Elevation of Privilege*) se ha utilizado para identificar amenazas en los componentes de un sistema (Palanivel & Selvadurai, 2014); además, la auditoría de este estudio se apoya en las referencias del Open Web Application Security Project (OWASP) sobre seguridad de interfaces de programación.

Evaluar ambos aspectos antes de probar la herramienta con estudiantes es, por tanto, una condición previa de la investigación pedagógica y no un trámite técnico, y define el foco de este estudio.

Frente a la ausencia de evidencia sobre una herramienta que todavía no se ha probado con estudiantes, se plantean las siguientes preguntas de investigación: ¿cómo opera la primera fase de CTEM como sistema de generación de grafos de conocimiento? ¿Se completa su flujo principal en una verificación funcional? ¿Cómo se documenta su proceso de aseguramiento de la calidad? ¿Qué perfil de amenazas, por nivel de riesgo, componente y categoría STRIDE, revela la auditoría de seguridad?

Con base en estas interrogantes, el objetivo general de este estudio es presentar CTEM, un sistema de generación de grafos de conocimiento para el aprendizaje matemático, y evaluar la calidad y la seguridad de su primera fase de desarrollo.

De manera específica, el estudio busca describir la arquitectura del sistema y el flujo principal mediante el cual genera los grafos a partir del audio de los debates, documentar el proceso de aseguramiento de la calidad (QA) aplicado y caracterizar las amenazas detectadas según su nivel de riesgo, su componente y su categoría STRIDE. Dado su carácter descriptivo, el estudio no formula hipótesis estadísticas.

2. Metodología

La investigación se desarrolló en la primera fase del proyecto CTEM y combinó un componente técnico con uno educativo. El componente técnico consistió en construir y evaluar un sistema que genera grafos de conocimiento a partir de debates en audio, y el educativo, en orientar ese sistema a la visualización de las conexiones entre conceptos clave de la enseñanza de las matemáticas. Las revisiones sobre construcción (Zhou et al., 2020), completación (Liang et al., 2024) y alineación de grafos de conocimiento (Zhu et al., 2024) se tomaron como marco conceptual para el diseño del sistema. Para su evaluación se combinaron prácticas de integración continua (Schubert & Argent, 2024; Soares et al., 2022) con un modelado de amenazas estructurado (Palanivel & Selvadurai, 2014; Van Landuyt & Joosen, 2022).

Se precisan a continuación el tipo, el enfoque, el método y el diseño del estudio, articulados entre sí. Por su finalidad, la investigación fue aplicada (Hernández-Sampieri & Mendoza, 2018), pues se orientó a construir y evaluar un artefacto de software funcional para resolver un problema concreto de la enseñanza de las matemáticas, y no a generar conocimiento teórico. El enfoque fue cuantitativo (Creswell & Creswell, 2018),

DOI: https://doi.org/10.29394/Scientific.issn.2542-2987.2026.11.39.11.150-165
OAI-PMH: http://www.indteca.com/ojs/index.php/Revista_Scientific/oai

porque los hallazgos de la evaluación se expresaron como frecuencias y porcentajes de amenazas por nivel de riesgo, por componente y por categoría STRIDE, procesados mediante estadística descriptiva y sin recolección de datos cualitativos de usuarios en esta fase.

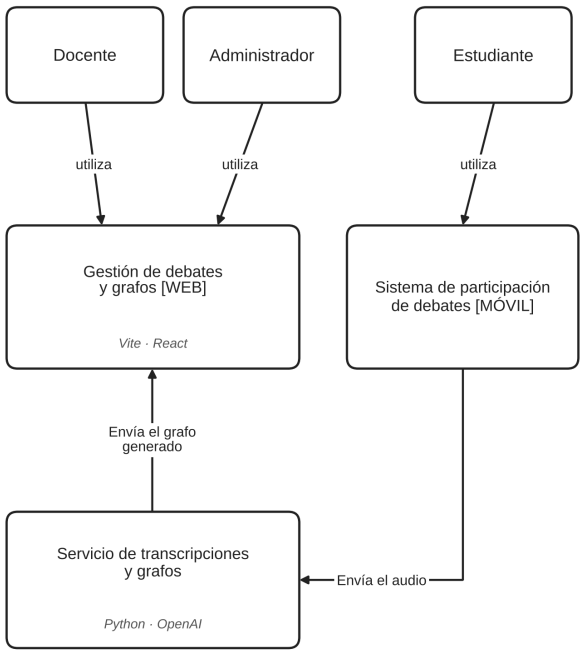
Como método se siguió la evaluación, que comparte metodologías y técnicas con la investigación, pero se orienta a valorar un objeto concreto para fundamentar decisiones sobre él (Cohen et al., 2018, cap. 5). En este estudio se valoró la calidad y la seguridad de un artefacto ya construido frente a criterios definidos de antemano. El alcance de la auditoría se delimitó a seis componentes que se evaluaron por separado, y los hallazgos se integraron después en una lectura global del riesgo. El diseño fue no experimental (Hernández-Sampieri & Mendoza, 2018) y transversal (Creswell & Creswell, 2018), pues la seguridad se registró en un único momento sobre el sistema y la calidad se documentó como proceso, sin manipular variables independientes ni comparar grupos, y su alcance fue descriptivo.

La evaluación se sustentó en dos referentes: los criterios de aceptación de QA definidos en la fase de análisis (pruebas unitarias, pruebas de integración y cobertura de código) y un modelado de amenazas estructurado según OWASP y STRIDE. Estos referentes orientaron la verificación de la funcionalidad y la seguridad del artefacto antes de fases experimentales con usuarios; los resultados de las pruebas de QA no se reportaron en este artículo. La verificación funcional, a cargo del equipo de desarrollo, consistió en recorrer el flujo principal (gestión de usuarios y roles, grabación de audio y generación del grafo) y registrar si se completaba o no; no se reportaron casos de prueba formales ni indicadores de desempeño.

2.1. Arquitectura de la Aplicación CTEM

CTEM se concibió para generar grafos de conocimiento a partir de debates académicos y apoyar así el aprendizaje matemático. El diagrama de contexto del sistema, que abarcó la aplicación web, la aplicación móvil y la lógica de negocio, se organizó en niveles según el modelo C4 (*Context, Containers, Components and Code*). Se documentaron el nivel 1, de contexto, y el nivel 2, de contenedores, y se acompañó cada uno con la descripción de las interacciones entre sus elementos.

Figura 1. Nivel 1 del modelo C4 de CTEM.



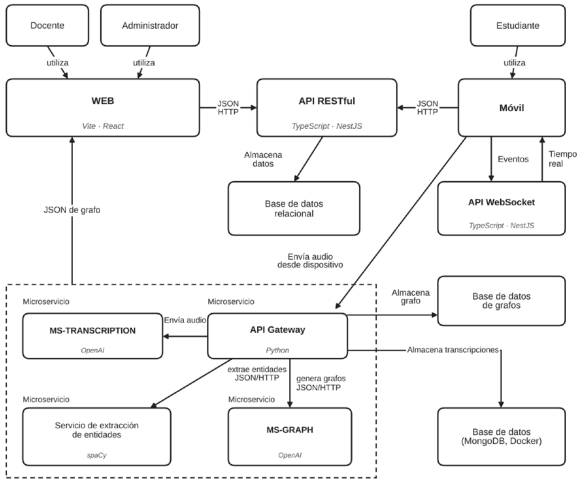
Nota. Fuente: Guamán, et al. (2025), a partir del diagrama de los autores, redibujado con rótulos corregidos.

El nivel 1 (figura 1) representó la interacción de los usuarios con el sistema. Cada usuario accedió con un rol específico que le otorgó permisos diferenciados. La aplicación web permitió a los usuarios con rol de administrador y de docente gestionar usuarios, salas y tópicos para los debates, y visualizar los grafos. Los usuarios con rol de estudiante interactuaron con la aplicación móvil, que capturó el audio mediante el micrófono del dispositivo y lo envió al servicio de transcripción y generación de grafos, que el nivel siguiente desglosa en microservicios. Este servicio integró procesamiento de lenguaje natural para transcribir el audio, reconocer entidades y extraer los datos relevantes de los debates, y generó los grafos de conocimiento. Finalmente, la aplicación web recibió los

DOI: https://doi.org/10.29394/Scientific.issn.2542-2987.2026.11.39.11.150-165
OAI-PMH: http://www.indteca.com/ojs/index.php/Revista_Scientific/oai

grafos generados y los puso a disposición de los usuarios autorizados, con los roles de docente y administrador. Estos pudieron visualizar la estructura del grafo, con sus nodos y conexiones derivados de los debates, y utilizarla en la evaluación que realizó el docente.

Figura 2. Nivel 2 del modelo C4 de CTEM.



Nota. Fuente: Guamán, et al. (2025), a partir del diagrama de los autores, redibujado con rótulos corregidos.

Para el nivel 2 (figura 2) se detalló la comunicación entre los contenedores y las bases de datos. Las aplicaciones web y móvil se comunicaron con una API de tipo REST que hizo de capa intermedia para acceder a la lógica de negocio y a la base de datos relacional principal, y que expuso puntos de acceso para crear, leer, actualizar y eliminar la información de usuarios, salas y tópicos. La aplicación móvil se comunicó, además, con una API WebSocket para el intercambio de eventos en tiempo real. Del lado del servidor, una arquitectura de microservicios, orquestada por un API Gateway, procesó las solicitudes de audio.

El Gateway dirigió las solicitudes al servicio de transcripción (MS-TRANSCRIPTION), al servicio de extracción de entidades basado en spaCy y al servicio de creación de grafos (MS-GRAPH); los servicios de transcripción y de grafos recurrieron a las capacidades de OpenAI para transcribir el audio, analizar el contenido, identificar las relaciones entre conceptos y construir los grafos de conocimiento, mientras que la extracción de entidades se aisló en un servicio propio. Para la persistencia de datos, los microservicios emplearon un modelo de almacenamiento basado en bases de datos NoSQL: la figura 2 identificó MongoDB como el motor de las transcripciones y una base de datos distinta para los grafos. Esta separación respecto de la base de datos relacional principal buscó que la capa de persistencia de los microservicios fuese independiente y escalable, con baja latencia para el almacenamiento de transcripciones, entidades y grafos de conocimiento, condiciones que el equipo consideró pertinentes para un sistema distribuido con procesamiento de datos complejos.

2.2. Desarrollo del Software y Aseguramiento de la Calidad (QA)

La construcción del software siguió una metodología ágil iterativa, con planificación por iteraciones, reuniones de seguimiento y entrega incremental de funcionalidades, sustentada en integración continua, lo que permitió ajustar la arquitectura y las funcionalidades según la retroalimentación del equipo. El diseño de la arquitectura determinó, con base en los requisitos del proyecto, las tecnologías, los patrones de diseño, las herramientas y los *frameworks* adecuados, así como la estructuración de los componentes para favorecer la escalabilidad y el mantenimiento, en línea con los principios y patrones de diseño que la literatura reúne para las arquitecturas de microservicios (Velepucha & Flores, 2023). Se desplegaron y probaron una aplicación web, una aplicación móvil y una API de tipo REST en distintos entornos, con configuración de servidores, integración de servicios y pruebas funcionales y de rendimiento.

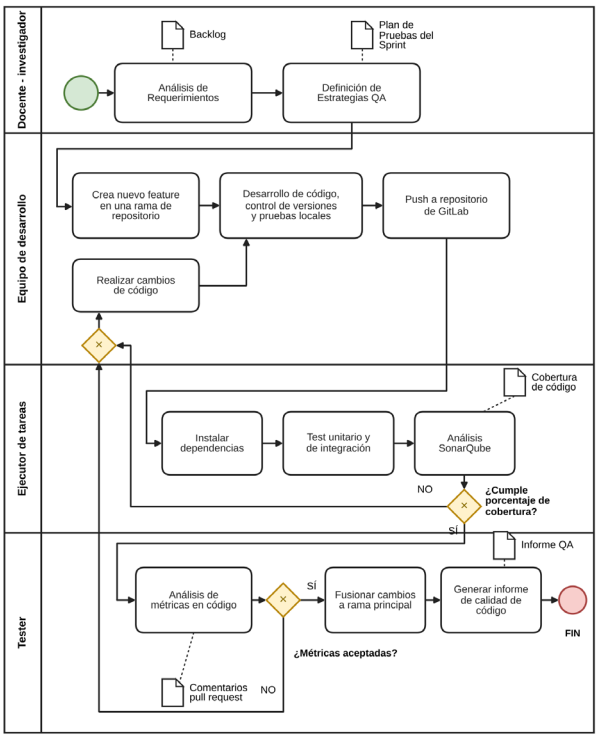
El aseguramiento de la calidad del código incluyó estándares de programación, análisis estático y automatización de pruebas, con el fin de identificar y corregir errores, mejorar la legibilidad y sostener un código mantenible. Se implementó un proceso estructurado en cuatro fases, desde la definición de objetivos y criterios de calidad hasta la revisión final del código, que integró pruebas automatizadas, análisis de

Revista Scientific - Artículo de Investigación Empírica - Arbitrado - Volumen 11, Número 39 - Febrero-Abril 2026 - pp. 150-165
pp. BA2016000002 - e-ISSN: 2542-2987 - ISNI: 0000 0004 6045 0361

DOI: https://doi.org/10.29394/Scientific.issn.2542-2987.2026.11.39.11.150-165
OAI-PMH: http://www.indteca.com/ojs/index.php/Revista_Scientific/oai

cobertura y revisión entre pares en cada iteración del desarrollo, en línea con las prácticas de integración continua mencionadas.

Figura 3. Proceso de aseguramiento de la calidad (QA) de CTEM.



Nota. Fuente: Guamán, et al. (2025), a partir del diagrama de los autores, redibujado con rótulos corregidos.

La figura 3 ilustró el proceso mediante un diagrama de carriles que distinguió cuatro roles: el docente-investigador, el equipo de desarrollo, el ejecutor de tareas y el *tester*. El flujo fue iterativo y no lineal: si el análisis estático de cobertura (SonarQube) no cumplía el porcentaje definido, o si el *tester* no aceptaba las métricas del código, el proceso regresaba a la etapa de realizar cambios de código antes de continuar, en lugar de avanzar hacia la fusión con la rama principal. El diagrama señaló, además, los artefactos que generaba cada etapa, que se detallan en los Resultados. Las cuatro fases del proceso se describieron a continuación.

2.2.1. Fase de Análisis

Al iniciar la primera fase se definieron los objetivos y el alcance del proyecto, así como los criterios de calidad que regirían todo el proceso de desarrollo. El equipo identificó y documentó los requisitos funcionales y no funcionales, y estableció los criterios de aceptación de cada uno. A partir de este análisis se elaboró una estrategia de QA que detalló los tipos de pruebas unitarias y de integración y las herramientas que se emplearían. El resultado de la fase fue una lista de características, o *backlog*, que sirvió de guía para las etapas siguientes y formalizó los lineamientos de calidad.

2.2.2. Fase de Desarrollo Local

Durante la segunda fase, los programadores trabajaron en nuevas funcionalidades o en correcciones de errores dentro de un entorno controlado, por lo general en ramas específicas del repositorio. En este punto se realizaron pruebas iniciales para comprobar que los cambios no introdujeran defectos graves. El proceso incluyó buenas prácticas de versionado y la configuración de un entorno de desarrollo que reflejara, en la medida de lo posible, el entorno productivo. Al finalizar, el código se envió al repositorio central de GitLab.

2.2.3. Fase de Pipeline de Automatización

Una vez que el código se integraba en el repositorio central, se activaba el *pipeline* de automatización. Este ejecutaba de forma secuencial la instalación de dependencias, las pruebas unitarias, las pruebas de integración y el análisis de calidad de código con SonarQube. El objetivo principal fue detectar problemas de manera temprana y garantizar la consistencia del producto en cada iteración. Si alguna de las pruebas o validaciones fallaba, se notificaba al equipo para que realizara las correcciones pertinentes antes de continuar.

2.2.4. Fase de Métricas y Revisión

Tras la ejecución del *pipeline* se recopilaron las métricas de cobertura de código y de calidad, y se compararon con los umbrales definidos en la estrategia de QA para determinar si los cambios cumplían los estándares establecidos. Después se realizó una

DOI: https://doi.org/10.29394/Scientific.issn.2542-2987.2026.11.39.11.150-165
OAI-PMH: http://www.indteca.com/ojs/index.php/Revista_Scientific/oai

revisión de código en la que los miembros del equipo evaluaron la implementación y verificaron la legibilidad y la adherencia a las convenciones de estilo. Si todo estaba conforme, se autorizaba la fusión con la rama principal; en caso contrario, se solicitaban ajustes y se repetía el ciclo hasta cumplir los criterios de calidad. Al final se generó un informe que documentó el estado final del proceso y las recomendaciones para futuras iteraciones.

2.3. Auditoría de Seguridad

La auditoría de seguridad se orientó a identificar vulnerabilidades y a mitigar riesgos potenciales, mediante análisis de código y auditorías de acceso que protegieran los datos y la integridad de la aplicación. El procedimiento se basó en una revisión manual, con inspección detallada de funciones críticas como la autenticación, el control de acceso, la gestión de sesiones y los validadores de entrada, y se apoyó en STRIDE como técnica de modelado de amenazas (Kwarteng et al., 2025; Palanivel & Selvadurai, 2014).

Se utilizó un modelado de amenazas estructurado en el que cada componente del sistema se analizó en función de los riesgos que podía presentar. Los componentes evaluados fueron la aplicación web, la API REST, la API WebSocket, la aplicación móvil, el servicio de transcripción y el servidor Ubuntu de despliegue. Los resultados se clasificaron conforme a los estándares de OWASP y al modelo STRIDE, con sus seis categorías de amenazas. Para asignar el nivel de riesgo (crítico, alto, medio, bajo, muy bajo y nulo) se consideraron factores como la cantidad de usuarios, los datos tratados, el alcance de uso y la exposición del sistema, de manera que una misma amenaza no recibió necesariamente la misma severidad en todos los sistemas, y los niveles asignados correspondieron al contexto del sistema evaluado. El API Gateway, el servicio de creación de grafos y el servicio de extracción de entidades no se evaluaron como componentes independientes.

2.4. Procesamiento y Análisis de los Datos

Las amenazas se registraron por componente, por nivel de riesgo y por categoría STRIDE. Con esas frecuencias se calcularon los totales, los porcentajes sobre el total de amenazas (sobre el total clasificado, en el caso de STRIDE) y la proporción de amenazas críticas y altas en cada componente, con un redondeo a un decimal. Estos cálculos fueron de elaboración propia y se realizaron a partir de las frecuencias registradas; los porcentajes pueden no sumar exactamente 100% por efecto del redondeo. Los datos se organizaron en tablas y figuras, y la interpretación se reservó para la Discusión. No se calcularon pruebas de inferencia, porque el estudio describe una única auditoría sobre un sistema y no compara grupos ni estima parámetros de una población.

2.5. Alcance y Consideraciones Éticas

La fase evaluada no involucró a estudiantes ni a docentes como sujetos de investigación, de modo que no se recolectaron datos personales con fines de análisis. La evaluación pedagógica con usuarios se reservó para una fase posterior, que requerirá la aprobación de un comité de ética y el consentimiento informado de los participantes.

3. Resultados

Los resultados se organizaron según los tres objetivos específicos y corresponden a la primera fase del proyecto CTEM, que abarcó la gestión de usuarios y roles, la grabación de audio, la visualización de grafos, una auditoría de seguridad y las pruebas de calidad del software previas a su implementación en estudios experimentales. Primero se presentó el flujo principal verificado, asociado a la arquitectura descrita en la metodología (primer objetivo específico), y el proceso de aseguramiento de la calidad allí documentado (segundo objetivo específico); después se caracterizaron las amenazas identificadas en la auditoría (tercer objetivo específico).

3.1. Estado Funcional y Proceso de Aseguramiento de la Calidad

En este apartado se presentó el flujo principal

DOI: https://doi.org/10.29394/Scientific.issn.2542-2987.2026.11.39.11.150-165
OAI-PMH: http://www.indteca.com/ojs/index.php/Revista_Scientific/oai

verificado y el proceso de aseguramiento de la calidad documentado.

3.1.1. Generación de Grafos: Flujo Principal Verificado

Al cierre de la fase, el flujo principal del sistema, que comprendió la gestión de usuarios y roles, la grabación de audio y la generación de grafos de conocimiento, se completó de extremo a extremo en la verificación funcional realizada por el equipo de desarrollo. El recorrido descrito en la figura 2 tuvo seis etapas: captura del audio en la aplicación móvil, recepción en el API Gateway, transcripción, extracción de entidades y generación del grafo, almacenamiento de las transcripciones y del grafo, y entrega del grafo a la aplicación web, donde los usuarios con rol de docente y de administrador pudieron visualizarlo. El grafo, con sus nodos y conexiones derivados del debate, fue el producto final del flujo y la base prevista para visualizar las relaciones entre conceptos matemáticos.

La evidencia disponible para este objetivo fue de tipo dicotómico (flujo completado o no completado) y procedió de la verificación del propio equipo de desarrollo. No se reportaron casos de prueba, tiempos de respuesta, tasas de error de la transcripción ni medidas de la calidad de los grafos generados, por lo que el resultado acredita la integración de los componentes y no su robustez ni su exactitud.

3.1.2. Proceso de QA Documentado

Se documentó un proceso de aseguramiento de la calidad en cuatro fases (Análisis, Desarrollo Local, *Pipeline* de Automatización, y Métricas y Revisión) que, según la figura 3, produjo artefactos verificables en varios tramos: el *backlog*, derivado del análisis de requerimientos, y el Plan de Pruebas del Sprint, derivado de la definición de estrategias de QA; el reporte de cobertura de código, derivado del análisis con SonarQube; y los comentarios del *pull request* y el Informe QA, derivados del análisis de métricas y de la revisión. El diagrama incluyó, además, dos puntos de decisión: el cumplimiento del porcentaje de cobertura y la aceptación de las métricas, que devolvían el trabajo a la etapa de cambios de código.

Los informes del proceso de aseguramiento de la calidad no se incluyeron en este reporte, por lo que no se presentaron el porcentaje de cobertura de código, el umbral de cobertura exigido ni el número de pruebas ejecutadas por tipo. Los resultados de este objetivo describieron, en consecuencia, el proceso documentado y no su efecto sobre los defectos del software.

3.2. Auditoría de Seguridad

Seis componentes del sistema fueron analizados en la auditoría, y las amenazas registradas se clasificaron por nivel de riesgo y, en su mayoría, por categoría STRIDE. Se presentaron primero los niveles de riesgo y después las categorías STRIDE, con tablas y figuras que permitieron comparar los componentes entre sí.

3.2.1. Niveles de Riesgo de las Amenazas

Para describir cómo se distribuyó el riesgo entre los componentes evaluados, se elaboró la tabla 1, que reunió las frecuencias de amenazas por componente y por nivel de riesgo, junto con el total de cada fila y la proporción de amenazas críticas y altas. Las figuras que la siguieron ilustraron esa distribución de manera global y por componente.

Tabla 1. Amenazas identificadas por componente y nivel de riesgo.

Componente	Crítico	Alto	Medio	Bajo	Muy bajo	Nulo	Total	Crítico + alto (%)
Aplicación web	2	0	1	0	1	2	6	33,3
API REST	12	13	7	4	0	0	36	69,4
API WebSocket	6	6	2	0	0	0	14	85,7
Aplicación móvil	0	5	3	0	0	0	8	62,5
Servicio de transcripción	4	11	5	0	0	0	20	75,0
Servidor Ubuntu	5	4	2	2	0	0	13	69,2
Total	**29**	**39**	**20**	**6**	**1**	**2**	**97**	**70,1**
% del total	29,9	40,2	20,6	6,2	1,0	2,1	100,0	

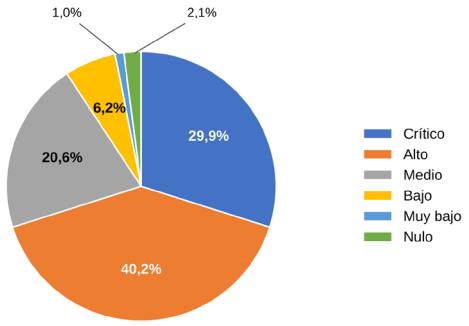
Nota. Fuente: Guamán, et al. (2025), a partir de las frecuencias registradas en la auditoría de seguridad de CTEM. Los porcentajes se calcularon sobre el total de 97 amenazas. La columna *Crítico + alto (%)* indica la proporción de amenazas críticas y altas dentro de cada componente.

Se registraron 97 amenazas en seis componentes (tabla 1). Las críticas sumaron 29 (29,9%) y las altas 39 (40,2%), con lo cual siete de cada diez amenazas (70,1%) se ubicaron en los dos niveles

DOI: https://doi.org/10.29394/Scientific.issn.2542-2987.2026.11.39.11.150-165
OAI-PMH: http://www.indteca.com/ojs/index.php/Revista_Scientific/oai

superiores. La API REST concentró el mayor número, con 36 amenazas, equivalentes a 37,1% del total, seguida por el servicio de transcripción (20) y la API WebSocket (14). La aplicación web, con seis, fue el componente con menos amenazas. Los niveles medio, bajo, muy bajo y nulo reunieron en conjunto 29 amenazas, es decir, 29,9% del total.

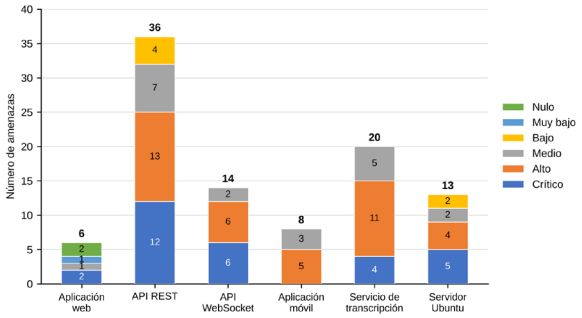
Figura 4. Distribución porcentual de las amenazas por nivel de riesgo.



Nota. Fuente: Guamán, et al. (2025), con base en los totales de la tabla 1 (N = 97).

Construida con los totales de la tabla 1, la figura 4 mostró una distribución inclinada hacia los niveles superiores. El nivel alto fue el sector más amplio, seguido del crítico y del medio; los niveles bajo, muy bajo y nulo constituyeron sectores pequeños, el del nivel muy bajo apenas visible. Sumadas, las categorías crítica y alta ocuparon más de dos tercios del círculo, mientras que el conjunto de los niveles inferiores al medio ocupó una fracción reducida del gráfico.

Figura 5. Niveles de riesgo de las amenazas por componente del sistema.



Nota. Fuente: Guamán, et al. (2025), con base en las frecuencias de la tabla 1 (N = 97).

Al desagregar los niveles de riesgo por componente, la figura 5 permitió observar perfiles distintos. En la API REST se combinaron el mayor número de amenazas y una concentración en los niveles crítico y alto. El servicio de transcripción presentó más de la mitad de sus amenazas en el nivel alto. La API WebSocket registró tantas amenazas críticas como altas, y la aplicación móvil no tuvo amenazas críticas, aunque la mayor parte se ubicó en el nivel alto. El servidor Ubuntu reunió amenazas desde el nivel crítico hasta el bajo, y la aplicación web mostró el perfil más disperso, con amenazas críticas, medias, muy bajas y nulas.

3.2.2. Categorías STRIDE de las Amenazas

Con el fin de caracterizar la naturaleza de las amenazas, se elaboró la tabla 2, que reunió las frecuencias por componente y por categoría STRIDE, con el total de cada fila y el porcentaje que representa cada categoría sobre el total clasificado. La figura 6 mostró la misma información de forma gráfica.

Tabla 2. Amenazas identificadas por componente y categoría STRIDE.

Componente	S	T	R	I	D	E	Total
Aplicación web	0	1	0	4	1	0	6
API REST	2	9	2	9	4	7	33
API WebSocket	2	3	2	3	3	0	13
Aplicación móvil	1	1	0	6	0	0	8
Servicio de transcripción	1	8	1	1	6	1	18
Servidor Ubuntu	2	1	0	2	3	5	13
Total	**8**	**23**	**5**	**25**	**17**	**13**	**91**
% del total	8,8	25,3	5,5	27,5	18,7	14,3	100,0

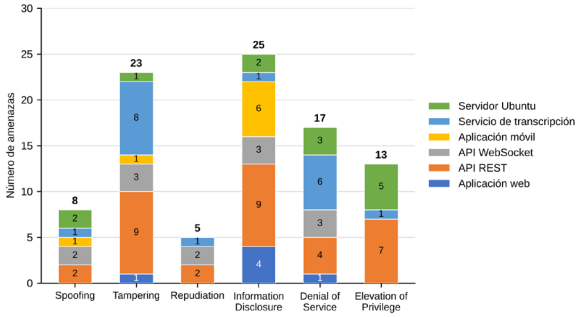
Nota. Fuente: Guamán, et al. (2025), a partir de las frecuencias registradas en la auditoría de seguridad de CTEM. S = *Spoofing*; T = *Tampering*; R = *Repudiation*; I = *Information Disclosure*; D = *Denial of Service*; E = *Elevation of Privilege*. Los porcentajes se calcularon sobre el total de 91 amenazas clasificadas y pueden no sumar 100,0% por redondeo.

Según la tabla 2, 91 amenazas quedaron clasificadas en alguna categoría STRIDE, seis menos que las 97 de la tabla 1. La diferencia se repartió entre la API REST (33 frente a 36), el servicio de transcripción (18 frente a 20) y la API WebSocket (13 frente a 14), mientras que los otros tres componentes coincidieron en ambas tablas. *Information Disclosure* (27,5%) y *Tampering* (25,3%) sumaron más de la mitad de las amenazas clasificadas (52,7%), y les siguieron *Denial of Service* (18,7%), *Elevation of Privilege* (14,3%), *Spoofing*

DOI: https://doi.org/10.29394/Scientific.issn.2542-2987.2026.11.39.11.150-165
OAI-PMH: http://www.indteca.com/ojs/index.php/Revista_Scientific/oai

(8,8%) y *Repudiation* (5,5%). En la API REST, *Tampering* e *Information Disclosure* aportaron nueve amenazas cada una, y *Elevation of Privilege*, siete.

Figura 6. Amenazas por categoría STRIDE y componente del sistema.



Nota. Fuente: Guamán, et al. (2025), con base en las frecuencias de la tabla 2 (N = 91).

Los componentes que dominaron cada categoría se aprecian en la figura 6. En *Information Disclosure*, la API REST aportó la porción más grande, seguida de la aplicación móvil; en *Tampering*, la API REST y el servicio de transcripción reunieron la mayor parte de las amenazas. *Denial of Service* se repartió sobre todo entre el servicio de transcripción y la API REST, y *Elevation of Privilege* se concentró en la API REST y en el servidor Ubuntu. *Spoofing* y *Repudiation* fueron las categorías menos frecuentes.

4. Discusión

Esta fase respalda la dimensión técnica de CTEM como sistema de generación de grafos para el aprendizaje matemático: el sistema integra la captura del audio, la transcripción, la extracción de entidades y la construcción del grafo en un flujo que llega a su visualización. La dimensión de aprendizaje, en cambio, queda por demostrar, porque el estudio no midió si los grafos representan con fidelidad los conceptos matemáticos del debate ni si ayudan a comprenderlos. Tampoco se evaluaron su pertinencia para el contexto de países de ingresos bajos y medios que motiva el estudio ni la medida en que su diseño recoge la perspectiva conectivista que inspira su nombre, cuestiones que corresponden a la fase pedagógica.

Que el flujo principal se completara de extremo a extremo indica que los componentes de la arquitectura (aplicaciones web y móvil, API REST, API Gateway y microservicios de transcripción, extracción de entidades y creación de grafos) pueden integrarse, pero la evidencia es de viabilidad técnica y no de robustez, pues no se reportaron los criterios de aceptación aplicados a esa verificación. La separación en microservicios descrita en la arquitectura del sistema es compatible con esa integración, aunque la escalabilidad, la latencia y la tolerancia a fallos no se reportaron en esta fase, y por ello no pueden darse por demostradas.

Conforme a Soares et al. (2022), cuya revisión de 101 estudios empíricos muestra que la integración continua aporta, en su mayoría, efectos positivos al desarrollo de software, aunque también puede traer desafíos técnicos, el proceso de aseguramiento de la calidad diseñado para CTEM es congruente con esa evidencia. En el ámbito del software científico, de acuerdo con Schubert y Argent (2024), la transición a la integración y la entrega continuas se vincula con la calidad del producto.

La estructura de CTEM en cuatro fases incorpora la integración continua descrita en la metodología. Esta coincidencia con la literatura, sin embargo, es de diseño y no de desempeño, y por tanto no permite contrastar el efecto del proceso sobre los defectos.

En la ejecución de pruebas, un estudio comparativo sobre priorización en integración continua muestra que el desempeño de los modelos de aprendizaje automático depende del presupuesto de tiempo y del historial usado para el entrenamiento (Marijan, 2023). CTEM aplica umbrales predefinidos de cobertura y calidad y no prioriza las pruebas; explorar esa priorización queda como línea futura, sin que este estudio permita valorar su beneficio.

En cuanto a la auditoría de seguridad, el hallazgo de que la API REST concentra el mayor número de amenazas es compatible con el lugar que ocupa en la arquitectura: es el punto de acceso común de las aplicaciones web y móvil a la lógica de negocio y a la

Revista Scientific - Artículo de Investigación Empírica - Arbitrado - Volumen 11, Número 39 - Febrero-Abril 2026 - pp. 150-165
pp. BA2016000002 - e-ISSN: 2542-2987 - ISNI: 0000 0004 6045 0361

DOI: https://doi.org/10.29394/Scientific.issn.2542-2987.2026.11.39.11.150-165
OAI-PMH: http://www.indteca.com/ojs/index.php/Revista_Scientific/oai

base de datos relacional principal. A modo de antecedente, la revisión de Berardi et al. (2022), que reunió 290 publicaciones, muestra que la literatura sobre seguridad en microservicios es amplia y dispersa; esa dispersión implica que no existe un marco único con el cual contrastar los conteos de CTEM, por lo que la concentración en la API REST no debe leerse como una propiedad general de los microservicios.

Los conteos absolutos, además, matizan la lectura. Aunque la API REST reúne el mayor número de amenazas, la API WebSocket presenta la proporción más alta de amenazas críticas y altas (12 de 14), seguida por el servicio de transcripción (15 de 20), y la API REST y el servidor Ubuntu se ubican en torno a siete de cada diez. En consecuencia, la remediación no debería limitarse al componente con más amenazas, sino atender también a otros componentes con menos amenazas pero con mayor proporción de críticas y altas.

La concentración en *Information Disclosure* y *Tampering* podría relacionarse con el uso no controlado de identificadores para modificar o crear datos, lo que habilitaría la obtención de roles superiores o el acceso a recursos sin autorización; el reporte no verificó esa hipótesis.

Por su parte, Van Landuyt y Joosen (2022) analizaron 640 supuestos formulados en 96 modelos de amenazas STRIDE de un mismo sistema y hallaron que los supuestos se emplean sobre todo para excluir amenazas del análisis posterior. Ese hallazgo recuerda que los conteos dependen de las decisiones de alcance del equipo auditor, por lo cual las cifras de este estudio describen la auditoría realizada y no un inventario exhaustivo de las amenazas posibles.

La orientación por riesgo de la auditoría guarda relación con la literatura. De forma afín, Palanivel y Selvadurai (2014) emplean el análisis de riesgos con STRIDE para seleccionar casos de prueba de seguridad y reducir el conjunto de pruebas; los conteos de las tablas 1 y 2 podrían alimentar en CTEM un procedimiento semejante, que ordene las pruebas según el riesgo de cada componente. A su vez, Kwarteng et al. (2025) encuestaron a 81 participantes de fabricantes de dispositivos médicos, encontraron que STRIDE es la metodología más utilizada y desarrollaron un asistente basado en un modelo de lenguaje para apoyar el modelado de amenazas. Su contexto difiere del de CTEM, pero el apoyo de modelos de lenguaje es una línea que podría explorarse en auditorías futuras, siempre con la validación que cualquier asistente automático exige.

Un aspecto relevante escapa a las cifras: el tratamiento de la voz de los estudiantes. CTEM envía el audio a servicios de transcripción y de generación de grafos que recurren a OpenAI, por lo que, además de las amenazas de seguridad, la fase pedagógica debe considerar los asuntos éticos identificados al revisar 21 estudios empíricos sobre analíticas de aprendizaje (Cerratto & McGrath, 2021). Asimismo, el API Gateway, el servicio de grafos y el servicio de extracción de entidades no se evaluaron como componentes separados; por consiguiente, los servicios que dependen de proveedores externos, en particular el de grafos, requieren una evaluación específica.

Cuando CTEM pase a la fase pedagógica, la visualización de grafos podría funcionar como una analítica del proceso y no solo del resultado. Una revisión sistemática de los tableros de analíticas de aprendizaje dirigidos al estudiante identifica una trayectoria emergente hacia diseños informados por marcos teóricos y orientados al aprendizaje (Paulsen & Lindsay, 2024).

En este sentido, Choi y Nam (2025) derivan 15 principios de diseño para el aprendizaje colaborativo en línea basado en analíticas, que pueden tomarse como punto de partida para planificar esa fase. De igual modo, la asociación negativa, de magnitud pequeña, entre el uso problemático del teléfono inteligente y el rendimiento académico (Paterna et al., 2024) invita a definir criterios de uso académico del dispositivo en el diseño de las sesiones, puesto que los estudiantes participan desde dispositivos móviles.

Una herramienta técnicamente funcional no

DOI: https://doi.org/10.29394/Scientific.issn.2542-2987.2026.11.39.11.150-165
OAI-PMH: http://www.indteca.com/ojs/index.php/Revista_Scientific/oai

garantiza, por sí sola, un efecto pedagógico. Por ello, el éxito técnico de la fase evaluada es una condición necesaria, pero no suficiente: la adopción de CTEM por parte de docentes y estudiantes y su efecto sobre la comprensión de los conceptos deberán documentarse con evidencia propia.

Dado que CTEM se apoya en debates académicos entre estudiantes, su pertinencia pedagógica podrá juzgarse frente a la literatura sobre aprendizaje activo ya revisada (Kurepa et al., 2019; Vale & Barbosa, 2023). Los grafos generados a partir de los debates podrían servir al docente para identificar qué conexiones entre conceptos construyen los estudiantes, aunque esa utilidad es una hipótesis que la fase pedagógica debe comprobar.

Cabe señalar las limitaciones de este estudio. La evaluación corresponde únicamente a la primera fase de desarrollo, centrada en la verificación técnica del flujo funcional y en la auditoría de seguridad; no se han realizado pruebas con usuarios reales ni con estudiantes, por lo que no existe evidencia todavía sobre el impacto pedagógico de CTEM. Además, la verificación funcional la realizó el propio equipo de desarrollo.

Los informes de QA quedaron fuera de este reporte, como se señaló en los resultados. Además, el reporte de la auditoría no permite explicar por qué seis amenazas del total no aparecen clasificadas según STRIDE ni informa el número de auditores, las fechas, la versión de OWASP empleada ni el acuerdo entre auditores. Las líneas futuras que se derivan de estas limitaciones se recogen en las Conclusiones.

5. Conclusiones

La primera fase de CTEM, como sistema de generación de grafos de conocimiento para el aprendizaje matemático, logra un flujo principal funcional: la gestión de usuarios y roles, la grabación de audio en la aplicación móvil y la generación de grafos de conocimiento visualizables en la aplicación web se completan de extremo a extremo en la verificación del equipo de desarrollo. La arquitectura de microservicios, descrita en dos niveles del modelo C4, resulta así viable para ese flujo, aunque no se reportaron mediciones de escalabilidad, de latencia ni de calidad de los grafos generados, por lo que esos atributos siguen sin ponerse a prueba. Como aspecto novedoso, el estudio integra en una misma evaluación técnica el flujo de generación de grafos a partir del audio de debates, el proceso de QA y el perfil de amenazas por componente, como paso previo a la evaluación pedagógica con estudiantes.

El procedimiento de QA en cuatro fases (análisis, desarrollo local, *pipeline* de automatización, y métricas y revisión) queda documentado, con integración continua, análisis estático con SonarQube, revisión entre pares y artefactos verificables en varias etapas. Su efecto sobre los defectos del software no puede afirmarse, porque el estudio no incluye los informes de cobertura ni de pruebas. Para mantener ese procedimiento conviene sostener la ejecución automatizada de pruebas y el análisis de cobertura en cada iteración, someter cada cambio a revisión entre pares y documentar todo el proceso para conservar la trazabilidad.

En los componentes evaluados en la auditoría de seguridad se observan amenazas, con predominio de los niveles crítico y alto. La API REST concentra el mayor número de amenazas, mientras que la API WebSocket y el servicio de transcripción presentan las proporciones más altas de severidad. Entre las amenazas clasificadas con STRIDE predominan *Information Disclosure* y *Tampering*. Estos conteos describen la auditoría realizada y no un inventario exhaustivo de riesgos: la arquitectura no garantiza por sí sola un sistema seguro, y la capa de API y el servicio de transcripción deben remediarse antes de un despliegue a mayor escala.

Los servicios de OpenAI que CTEM emplea para transcribir el audio y relacionar conceptos automatizan en esta fase la transformación de las participaciones orales en datos estructurados; sin embargo, este estudio no evalúa la exactitud de las transcripciones ni de los grafos. Entre sus limitaciones, la evaluación se restringe a la primera fase de desarrollo, se apoya en la verificación del propio equipo de desarrollo y no incluye

DOI: https://doi.org/10.29394/Scientific.issn.2542-2987.2026.11.39.11.150-165
OAI-PMH: http://www.indteca.com/ojs/index.php/Revista_Scientific/oai

pruebas con estudiantes ni docentes, por lo que el efecto pedagógico de CTEM permanece sin demostrar.

Se recomienda que futuras investigaciones incorporen estudios piloto con estudiantes y docentes para medir el impacto pedagógico de CTEM, con aprobación ética y consentimiento informado; que se remedien y se vuelvan a auditar las amenazas críticas y altas, con prioridad en la API REST, la API WebSocket y el servicio de transcripción, antes de ampliar su despliegue; que se publiquen los informes de cobertura y de pruebas del proceso de QA; que se evalúen de forma específica los servicios de inteligencia artificial y la protección de los datos de voz; y que se exploren técnicas de priorización automatizada de pruebas (Marijan, 2023) para valorar su utilidad cuando crezca el volumen de pruebas del sistema.

6. Referencias

Abu-Salih, B., & Alotaibi, S. (2024). A systematic literature review of knowledge graph construction and application in education. *Heliyon, 10*(3), e25383. https://doi.org/10.1016/j.heliyon.2024.e25383

Alshammari, S. H. (2023). Critical factors affecting students' intention to adopt technology-enhanced learning. *International Journal of Education in Mathematics, Science and Technology, 11*(2), 542–553. https://doi.org/10.46328/ijemst.3206

Bell, F. (2011). Connectivism: Its place in theory-informed research and innovation in technology-enabled learning. *The International Review of Research in Open and Distributed Learning, 12*(3), 98–118. https://doi.org/10.19173/irrodl.v12i3.902

Berardi, D., Giallorenzo, S., Mauro, J., Melis, A., Montesi, F., & Prandini, M. (2022). Microservice security: A systematic literature review. *PeerJ Computer Science, 8*, e779. https://doi.org/10.7717/peerj-cs.779

Bergdahl, N., Bond, M., Sjöberg, J., Dougherty, M., & Oxley, E. (2024). Unpacking student engagement in higher education learning analytics: A systematic review. *International Journal of Educational Technology in Higher Education, 21*(1), 63. https://doi.org/10.1186/s41239-024-00493-y

Cerratto Pargman, T., & McGrath, C. (2021). Mapping the ethics of learning analytics in higher education: A systematic literature review of empirical research. *Journal of Learning Analytics, 8*(2), 123–139. https://doi.org/10.18608/jla.2021.1

Chai, X., Yang, J., & Liu, Y. (2024). Influential factors for medical students' classroom concentration: Evaluation with speech recognition and face recognition technology. *BMC Medical Education, 24*(1), 1236. https://doi.org/10.1186/s12909-024-06204-5

Choi, J., & Nam, C. (2025). Development of design principles based on learning analytics for online collaborative learning in higher education. *Korean Journal of Educational Research, 63*(3), 1–30. https://doi.org/10.30916/kera.63.3.1

Cohen, L., Manion, L., & Morrison, K. (2018). *Research methods in education* (8th ed.). Routledge.

Creswell, J. W., & Creswell, J. D. (2018). *Research design: Qualitative, quantitative, and mixed methods approaches* (5th ed.). SAGE Publications.

Engelbrecht, J., & Borba, M. C. (2024). Recent developments in using digital technology in mathematics education. *ZDM – Mathematics Education, 56*(2), 281–292. https://doi.org/10.1007/s11858-023-01530-2

Gqoli, N. (2024). Digital technologies for mathematics learning in rural higher education: Students' perspectives. *Research in Social Sciences and Technology, 9*(1), 265–278. https://doi.org/10.46303/ressat.2024.15

Hernández-Sampieri, R., & Mendoza, C. (2018). *Metodología de la investigación: Las rutas cuantitativa, cualitativa y mixta*. McGraw-Hill Education.

Ifenthaler, D., & Yau, J. Y.-K. (2020). Utilising learning analytics to support study success in higher education: A systematic review. *Educational Technology Research and Development, 68*(4), 1961–1990. https://doi.org/10.1007/s11423-020-09788-z

Kurepa, A., Roop, J. P., & Edoh, K. (2019). Changing students' perception of mathematics through active learning. *International Journal of Education, 11*(1), 29–39. https://doi.org/10.5296/ije.v11i1.13983

Kwarteng, E., Cebe, M., Kwarteng, J., & Bansal, N. (2025). A comprehensive user study on medical device threat modeling methodologies and CyberLlama2 - MEDICALHARM threat modeling assistant. *International Journal of Information Security, 24*(4), 180. https://doi.org/10.1007/s10207-025-01094-0

Liang, X., Si, G., Li, J., Tian, P., An, Z., & Zhou, F. (2024). A survey of inductive knowledge graph completion. *Neural Computing and Applications, 36*(8), 3837–3858. https://doi.org/10.1007/s00521-023-09286-2

Marijan, D. (2023). Comparative study of machine learning test case prioritization for continuous integration testing. *Software Quality Journal, 31*(4), 1415–1438. https://doi.org/10.1007/s11219-023-09646-0

Nautsch, A., Jasserand, C., Kindt, E., Todisco, M., Trancoso, I., & Evans, N. (2019). The GDPR & speech data: Reflections of legal and technology communities, first steps towards a common understanding. In *Proceedings of Interspeech*

DOI: https://doi.org/10.29394/Scientific.issn.2542-2987.2026.11.39.11.150-165
OAI-PMH: http://www.indteca.com/ojs/index.php/Revista_Scientific/oai

2019 (pp. 3695–3699). ISCA. https://doi.org/10.21437/Interspeech.2019-2647

OECD. (2018). *PISA for development: Results in focus* (PISA in Focus). OECD Publishing.

OECD. (2019). *PISA 2018 results (Volume I): What students know and can do*. OECD Publishing.

Palanivel, M., & Selvadurai, K. (2014). Risk-driven security testing using risk analysis with threat modeling approach. *SpringerPlus, 3*(1), 754. https://doi.org/10.1186/2193-1801-3-754

Paterna, A., Alcaraz-Ibáñez, M., Aguilar-Parra, J. M., Salavera, C., Demetrovics, Z., & Griffiths, M. D. (2024). Problematic smartphone use and academic achievement: A systematic review and meta-analysis. *Journal of Behavioral Addictions, 13*(2), 313–326. https://doi.org/10.1556/2006.2024.00014

Paulsen, L., & Lindsay, E. (2024). Learning analytics dashboards are increasingly becoming about learning and not just analytics: A systematic review. *Education and Information Technologies, 29*(11), 14279–14308. https://doi.org/10.1007/s10639-023-12401-4

Schubert, A., & Argent, R. (2024). Promoting scientific software quality through transition to continuous integration and continuous delivery. *Socio-Environmental Systems Modelling, 6*, 18779. https://doi.org/10.18174/sesmo.18779

Soares, E., Sizilio, G., Santos, J., da Costa, D. A., & Kulesza, U. (2022). The effects of continuous integration on software development: A systematic literature review. *Empirical Software Engineering, 27*(3), 78. https://doi.org/10.1007/s10664-021-10114-1

Vale, I., & Barbosa, A. (2023). Active learning strategies for an effective mathematics teaching and learning. *European Journal of Science and Mathematics Education, 11*(3), 573–588. https://doi.org/10.30935/scimath/13135

Van Landuyt, D., & Joosen, W. (2022). A descriptive study of assumptions in STRIDE security threat modeling. *Software and Systems Modeling, 21*(6), 2311–2328. https://doi.org/10.1007/s10270-021-00941-7

Velepucha, V., & Flores, P. (2023). A survey on microservices architecture: Principles, patterns and migration challenges. *IEEE Access, 11*, 88339–88358. https://doi.org/10.1109/ACCESS.2023.3305687

Wong, S. L., & Wong, S. L. (2019). Relationship between interest and mathematics performance in a technology-enhanced learning context in Malaysia. *Research and Practice in Technology Enhanced Learning, 14*(1), 21. https://doi.org/10.1186/s41039-019-0114-3

Zhou, Y., Wang, W., Qiao, Z., Xiao, M., & Du, Y. (2020). A survey on the construction methods and applications of sci-tech big data knowledge graph. *SCIENTIA SINICA Informationis, 50*(7), 957–987. https://doi.org/10.1360/ssi-2019-0271

Zhu, B., Wang, R., Wang, J., Shao, F., & Wang, K. (2024). A survey: Knowledge graph entity alignment research based on graph embedding. *Artificial Intelligence Review, 57*(9), 229. https://doi.org/10.1007/s10462-024-10866-4

Reseña Biográfica de los autores

Marco Aurelio Guamán Buestán | maguaman2@sudamericano.edu.ec

Es ingeniero informático y docente del Instituto Tecnológico Superior Sudamericano (ITS), donde imparte las asignaturas de Programación Web y Despliegue de Aplicaciones. Ha participado en investigaciones aplicadas sobre tecnologías de asistencia para personas con discapacidad visual. Su trabajo combina el desarrollo de soluciones tecnológicas innovadoras con la pedagogía orientada al aprendizaje significativo en carreras técnicas.

Nancy Maribel Eras Eras | nmeras@sudamericano.edu.ec

Es ingeniera electrónica y en telecomunicaciones, con formación en la Universidad de Cuenca (UCUENCA), donde desarrolló su tesis de grado vinculada al diseño de sistemas electrónicos aplicados a la educación y las telecomunicaciones. Es docente e investigadora del Instituto Tecnológico Superior Sudamericano (ITS), Cuenca, Ecuador.

Juan Marcelo Pérez Pérez | jperez@sudamericano.edu.ec

Es ingeniero en Sistemas Informáticos y magíster en Educación, con más de siete años como docente titular del Instituto Tecnológico Superior Sudamericano (ITS), Cuenca. Lidera proyectos como la aplicación de la metodología

Revista Scientific - Artículo de Investigación Empírica - Arbitrado - Volumen 11, Número 39 - Febrero-Abril 2026 - pp. 150-165
pp. BA2016000002 - e-ISSN: 2542-2987 - ISNI: 0000 0004 6045 0361

DOI: https://doi.org/10.29394/Scientific.issn.2542-2987.2026.11.39.11.150-165
OAI-PMH: http://www.indteca.com/ojs/index.php/Revista_Scientific/oai

STEAM para el *Sumak Kawsay* en sectores vulnerables y una solución gamificada para adultos mayores con artrosis. Ha sido conferencista en el Congreso de Innovación Tecnológica en dos ocasiones y ha publicado investigaciones en revistas indexadas sobre aplicativos web y gamificación en educación especial.

Jonnathan Danilo Vallejo Sacoto | jdvallejo@sudamericano.edu.ec

Es tecnólogo superior en Desarrollo de Software y docente del Instituto Tecnológico Superior Sudamericano (ITS), Cuenca, Ecuador, donde ejerce funciones académicas y de acompañamiento de proyectos. Ha participado como tutor de trabajos técnicos y como ponente invitado en eventos institucionales de innovación tecnológica.

Declaraciones de los autores

Contribución de los autores (Taxonomía CRediT). Marco Aurelio Guamán Buestán: Conceptualización, Metodología, Software, Escritura - borrador original. Nancy Maribel Eras Eras: Conceptualización, Investigación, Escritura - revisión y edición. Juan Marcelo Pérez Pérez: Administración del proyecto, Supervisión, Validación, Escritura - revisión y edición. Jonnathan Danilo Vallejo Sacoto: Software, Validación, Recursos.

Financiamiento. Esta investigación no recibió financiamiento externo.

Conflicto de intereses. Los autores declaran no tener conflicto de intereses.

Declaración de disponibilidad de datos. El código fuente, la documentación técnica y los artefactos de evaluación (QA y auditoría de seguridad) generados en esta investigación están disponibles a solicitud razonable dirigida al autor de correspondencia (maguaman2@sudamericano.edu.ec).

Declaración de uso de Inteligencia Artificial. Se utilizó un asistente de inteligencia artificial generativa (Claude, Anthropic) como apoyo en tareas de edición: la adecuación del formato, las citas y las referencias a las normas APA 7.ª edición, y la corrección ortográfica y de estilo. Todo el contenido fue revisado, verificado y validado por los autores, quienes asumen la responsabilidad final sobre los datos, los resultados y las conclusiones del estudio.

Aprobación ética y consentimiento informado. Este estudio no requirió aprobación de un comité de ética, dado que esta etapa de la investigación consistió en el desarrollo y la evaluación técnica del software (arquitectura, aseguramiento de la calidad y auditoría de seguridad) y no involucró la participación de seres humanos como sujetos de investigación. Los futuros estudios piloto con estudiantes y docentes mencionados en este artículo sí requerirán la aprobación de un comité de ética y el consentimiento informado correspondiente.

El contenido de este manuscrito se difunde bajo una Licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional