

Revisión sistemática y análisis comparativo de lenguajes de programación para el desarrollo de aplicaciones web orientadas a la enseñanza de idiomas

Systematic review and comparative analysis of programming languages for the development of web applications oriented to language teaching

Marco Vinicio Ramos Valencia¹[0000-0003-3033-2404], Natalia Patricia Layedra Larrea²[0000-0003-1017-1746], Raúl Marcelo Lozada Yáñez³[0000-0001-9245-0858], Steven Alejandro Salazar Cazco⁴[0000-0002-9708-5885]

¹⁻⁴ Escuela Superior Politécnica de Chimborazo (ESPOCH). Facultad de Informática y Electrónica. Riobamba. Ecuador

{¹vi_ramos, ²natalia.layedra, ³raul.lozada, ⁴steven.salazar}@esPOCH.edu.ec

CITA EN APA:

Ramos Valencia, M. V., Layedra Larrea, N. P., Lozada Yáñez, R. M., & Salazar Cazco, S. A. (2026). Revisión sistemática y análisis comparativo de lenguajes de programación para el desarrollo de aplicaciones web orientadas a la enseñanza de idiomas. *Tesla Revista Científica*, 6(1), e576. <https://doi.org/10.55204/trc.v6i1.e576>

Recibido: 2025-12-02

Revisado: 2025-12-05 al 2025-12-23

Corregido: 2026-01-20

Aceptado: 2026-02-02

Publicado: 2026-02-05

TESLA

Revista Científica
ISSN: 2796-9320



Los contenidos de este artículo están bajo una licencia de Creative Commons Attribution 4.0 International (CC BY 4.0)

Los autores conservan los derechos morales y patrimoniales de sus obras.

The contents of this article are under a Creative Commons Attribution 4.0 International (CC BY 4.0) license. The authors retain the moral and patrimonial rights of their works.

Resumen. El desarrollo de aplicaciones web para la enseñanza de idiomas se ha incrementado significativamente en la última década debido al avance de las tecnologías digitales, la expansión del aprendizaje en línea y la necesidad de soluciones educativas flexibles, interactivas y escalables. En este contexto, la selección del lenguaje de programación adecuado constituye un factor crítico para garantizar el rendimiento, la mantenibilidad y la efectividad pedagógica de estas plataformas. El presente artículo científico desarrolla una revisión sistemática de la literatura y un análisis comparativo de cinco lenguajes de programación ampliamente utilizados en el desarrollo de aplicaciones web: JavaScript, Python, Java, PHP y C#. El estudio se fundamenta en criterios técnicos y educativos, considerando parámetros como facilidad de desarrollo, rendimiento, escalabilidad, soporte multimedia, ecosistema de frameworks y adecuación pedagógica. Los resultados evidencian que no existe un lenguaje universalmente superior, sino que la idoneidad depende de los objetivos pedagógicos, el alcance del proyecto y los recursos disponibles.

Palabras Clave: Aplicaciones web educativas, enseñanza de idiomas, lenguajes de programación, revisión sistemática, desarrollo web.

Abstract: The development of web applications for language teaching has increased significantly in the last decade due to advances in digital technologies, the expansion of online learning, and the need for flexible, interactive, and scalable educational solutions. In this context, selecting the appropriate programming language is a critical factor in ensuring the performance, maintainability, and pedagogical effectiveness of these platforms. This scientific article presents a systematic literature review and a comparative analysis of five programming languages widely used in web application development: JavaScript, Python, Java, PHP, and C#. The study is based on technical and educational criteria, considering parameters such as ease of development, performance, scalability, multimedia support, framework ecosystem, and pedagogical suitability. The results show that there is no universally superior language; rather, suitability depends on the pedagogical objectives, the scope of the project, and the available resources

Keywords: Educational web applications, language teaching, programming languages, systematic review, web development.

1. INTRODUCCIÓN

El aprendizaje de idiomas constituye un componente esencial en la formación académica y profesional en un contexto globalizado, donde la comunicación intercultural y el acceso a información en múltiples lenguas se han vuelto competencias clave. En las últimas dos décadas, el avance de las tecnologías de la información y la comunicación (TIC) ha impulsado de manera significativa el desarrollo de aplicaciones web educativas, transformando los métodos tradicionales de enseñanza de idiomas hacia modelos más flexibles, interactivos y centrados en el estudiante.

Diversos estudios han evidenciado que las aplicaciones web para la enseñanza de idiomas

permiten integrar recursos multimedia, retroalimentación inmediata y aprendizaje autónomo, elementos que favorecen la motivación y el progreso del estudiante (Laurillard, 2012). En particular, plataformas digitales basadas en la web han demostrado mejorar la adquisición de competencias lingüísticas cuando incorporan actividades interactivas, seguimiento del desempeño y adaptación al ritmo de aprendizaje del usuario.

Desde la perspectiva tecnológica, la ingeniería de software desempeña un rol fundamental en la calidad de estas plataformas. Sommerville (2016) sostiene que la selección de tecnologías adecuadas incide directamente en la mantenibilidad, escalabilidad y evolución de los sistemas educativos digitales. De manera similar, Pressman y Maxim (2014) destacan que las decisiones relacionadas con lenguajes de programación y frameworks impactan tanto en el ciclo de vida del software como en la experiencia del usuario final.

Estudios previos en el ámbito del desarrollo de software educativo han analizado comparativamente distintas tecnologías para plataformas de aprendizaje en línea, señalando que no existe una solución universalmente óptima, sino alternativas que se ajustan a diferentes contextos educativos y organizacionales (Kitchenham & Charters, 2007). En el caso específico de la enseñanza de idiomas, investigaciones recientes resaltan la importancia de lenguajes y entornos que faciliten la interactividad, el uso de multimedia y la incorporación de analítica de aprendizaje e inteligencia artificial (Okoli, 2015).

A pesar de la abundante literatura sobre tecnologías web y educación digital, se identifica una necesidad de estudios que integren de forma sistemática criterios técnicos y pedagógicos para evaluar lenguajes de programación en aplicaciones web orientadas específicamente a la enseñanza de idiomas. Este vacío motiva la realización del presente estudio.

El objetivo general de esta investigación es realizar una revisión sistemática de la literatura y un análisis comparativo de lenguajes de programación utilizados en el desarrollo de aplicaciones web para la enseñanza de idiomas, considerando indicadores técnicos y pedagógicos que permitan identificar sus fortalezas, limitaciones y niveles de adecuación en distintos contextos educativos.

2. METODOLOGÍA

La presente investigación adopta un enfoque cualitativo–descriptivo, basado en una revisión sistemática de la literatura (RSL), con el propósito de analizar de manera rigurosa y replicable los lenguajes de programación utilizados en el desarrollo de aplicaciones web orientadas a la enseñanza de idiomas, siguiendo los lineamientos metodológicos propuestos por Kitchenham y Charters (2007) para estudios secundarios en ingeniería de software.

El diseño metodológico se fundamenta en lineamientos ampliamente aceptados en investigaciones de ingeniería de software y tecnología educativa, particularmente en las propuestas de Kitchenham y Charters (2007) y las recomendaciones del modelo PRISMA para revisiones sistemáticas.

La búsqueda bibliográfica se realizó entre los meses de enero y marzo de 2026, utilizando bases de datos académicas y técnicas reconocidas por su rigor científico y relevancia en el área tecnológica y

educativa, de acuerdo con las recomendaciones para revisiones sistemáticas en ciencias de la computación y educación digital (Okoli, 2015). Entre las principales fuentes se incluyen:

- IEEE Xplore
- ACM Digital Library
- Scopus
- Google Scholar
- Documentación oficial de lenguajes y frameworks

Se emplearon combinaciones de palabras clave en inglés y español, tales como: "web applications", "language learning platforms", "programming languages", "educational software", "teaching languages online", y "technology-enhanced language learning". Estas palabras clave se combinaron mediante operadores booleanos (AND, OR) para maximizar la cobertura y precisión de los resultados.

Población de estudio.- La población de esta revisión sistemática estuvo conformada por un total aproximado de 320 documentos científicos y técnicos publicados entre los años 2014 y 2025. Esta población incluye:

- Artículos científicos revisados por pares
- Libros académicos especializados en ingeniería de software
- Estudios de caso sobre plataformas de aprendizaje de idiomas
- Documentación técnica oficial de lenguajes y frameworks web

La delimitación temporal se estableció con el objetivo de garantizar la vigencia tecnológica de los lenguajes analizados, considerando la rápida evolución del desarrollo web.

Muestra y técnica de muestreo. - A partir de la población inicial, se aplicó un muestreo no probabilístico de tipo intencional, seleccionando únicamente los documentos que cumplieran estrictamente con los criterios de inclusión definidos. Como resultado, la muestra final estuvo compuesta por 85 documentos, distribuidos de la siguiente manera:

- 45 artículos científicos
- 20 libros y capítulos de libros
- 20 documentos técnicos y reportes oficiales

Este tipo de muestreo es apropiado en estudios de revisión sistemática, dado que el interés no radica en la generalización estadística, sino en la profundidad y relevancia del análisis (Kitchenham & Charters, 2007).

Criterios de inclusión y exclusión. –

Criterios de inclusión:

- Publicaciones relacionadas con desarrollo de aplicaciones web.
- Estudios que mencionen explícitamente el uso de lenguajes de programación en contextos educativos.

- Documentos que aborden plataformas de enseñanza de idiomas o aprendizaje asistido por tecnología.
- Publicaciones en inglés o español.

Criterios de exclusión:

- Estudios duplicados.
- Publicaciones con información insuficiente o desactualizada.
- Artículos sin respaldo académico o técnico verificable.

Indicadores y parámetros de análisis

Los indicadores utilizados para el análisis comparativo fueron seleccionados con base en la literatura especializada en ingeniería de software educativo y aprendizaje digital, particularmente en modelos de evaluación tecnológica propuestos por Pressman (2014) y Sommerville (2016). Cada indicador responde a necesidades técnicas y pedagógicas propias de las aplicaciones de enseñanza de idiomas:

- **Facilidad de desarrollo:** Evaluada en función de la curva de aprendizaje, claridad sintáctica y disponibilidad de herramientas de desarrollo.
- **Rendimiento:** Considera la eficiencia en el procesamiento de solicitudes, manejo de concurrencia y tiempos de respuesta.
- **Escalabilidad:** Analiza la capacidad del lenguaje para soportar un crecimiento sostenido de usuarios y contenidos.
- **Soporte multimedia e interactividad:** Evalúa la integración con audio, video, animaciones y comunicación en tiempo real.
- **Ecosistema y frameworks:** Examina la madurez, estabilidad y adopción de frameworks asociados.
- **Adecuación pedagógica:** Considera la facilidad de integrar analítica de aprendizaje, personalización e inteligencia artificial.

Estos indicadores han sido ampliamente utilizados en estudios previos sobre evaluación de tecnologías educativas y plataformas web (Sommerville, 2016; Pressman, 2014; Laurillard, 2012), así como en investigaciones sobre diseño pedagógico mediado por tecnología (Laurillard, 2012).

Los criterios de inclusión fueron:

- Lenguajes de programación con uso comprobado en aplicaciones web.
- Evidencia de aplicación en entornos educativos o plataformas de aprendizaje.
- Disponibilidad de frameworks modernos y comunidades activas.

Con base en los criterios establecidos, se seleccionaron los siguientes lenguajes:

- JavaScript (Node.js)
- Python
- Java

- PHP
- C# (ASP.NET Core)

Cada documento seleccionado fue analizado mediante una matriz de extracción de datos, en la cual se registraron evidencias relacionadas con los indicadores definidos. Posteriormente, se realizó una comparación cruzada entre los lenguajes de programación, permitiendo identificar patrones, similitudes y diferencias relevantes.

La metodología descrita garantiza la validez, confiabilidad y reproducibilidad del estudio, elementos fundamentales en esta investigación.

3. RESULTADOS Y DISCUSIÓN

Resultados:

Los resultados muestran que JavaScript destaca en interactividad, Python en inteligencia artificial, Java y C# en escalabilidad, y PHP en facilidad de implementación para proyectos de bajo costo. Como lo muestra la tabla 1.

Tabla 1. Comparación de lenguajes de programación según indicadores técnicos y pedagógicos

LENGUAJE	FACILIDAD DE DESARROLLO	RENDIMIENTO	ESCALABILIDAD	SOPORTE MULTIMEDIA E INTERACTIVIDAD	ECOSISTEMA Y FRAMEWORKS
JAVASCRIPT	Alta	Alta	Alta	Muy alta	Muy amplio
PYTHON	Muy alta	Media	Media	Alta	Muy amplio
JAVA	Media	Muy alta	Muy alta	Alta	Amplio
PHP	Alta	Media	Media	Media	Amplio
C# (.NET)	Media-alta	Muy alta	Muy alta	Alta	Amplio

Nota. La valoración de los indicadores se realizó a partir del análisis cualitativo de la literatura seleccionada, considerando tanto aspectos técnicos como pedagógicos. La categorización (muy alta, alta, media) se emplea con fines comparativos y no representa mediciones cuantitativas absolutas.

JavaScript (Node.js). - Es el lenguaje predominante en el desarrollo web moderno. Su principal fortaleza radica en la posibilidad de utilizar un mismo lenguaje tanto en el frontend como en el backend, lo que simplifica la arquitectura y reduce la complejidad del desarrollo. Node.js permite manejar múltiples conexiones concurrentes, lo cual resulta especialmente útil en aplicaciones educativas con alta interacción en tiempo real.

En el contexto de la enseñanza de idiomas, JavaScript ofrece un excelente soporte para recursos multimedia, animaciones, juegos educativos y comunicación síncrona mediante tecnologías como WebRTC. Frameworks como React y Vue facilitan la creación de interfaces dinámicas y centradas en el usuario.

Python. - Destaca por su simplicidad sintáctica y su amplia adopción en el ámbito académico. Frameworks como Django y Flask permiten desarrollar aplicaciones web de manera rápida y estructurada. Aunque su rendimiento es inferior al de lenguajes compilados, Python compensa esta limitación con su integración nativa con bibliotecas de inteligencia artificial y procesamiento de lenguaje natural. En aplicaciones de enseñanza de idiomas, Python es especialmente relevante para la personalización del aprendizaje, la recomendación de contenidos y el análisis del progreso del estudiante.

Java. - Es un lenguaje consolidado en entornos empresariales y educativos. Su fortaleza principal

es la escalabilidad y la robustez, lo que lo convierte en una opción adecuada para plataformas educativas de gran tamaño. Frameworks como Spring Boot facilitan la creación de aplicaciones web seguras y mantenibles. No obstante, su curva de aprendizaje es más pronunciada y el desarrollo puede resultar más lento en comparación con lenguajes más dinámicos.

PHP.- Ha sido históricamente uno de los lenguajes más utilizados en el desarrollo web. Su facilidad de uso y amplia disponibilidad en servicios de hosting lo hacen atractivo para proyectos educativos con recursos limitados. Frameworks como Laravel han modernizado su ecosistema, mejorando la seguridad y la estructura del código. Sin embargo, su capacidad para manejar aplicaciones altamente interactivas depende en gran medida de la integración con JavaScript.

C# (ASP.NET Core). - C# es un lenguaje moderno y orientado a objetos, ampliamente utilizado en entornos corporativos. ASP.NET Core ofrece alto rendimiento, seguridad y soporte para arquitecturas escalables. Su integración con servicios en la nube lo hace adecuado para plataformas educativas que requieren alta disponibilidad y analítica avanzada.

En la tabla 2, podemos revisar las ventajas y desventajas de los lenguajes de programación para la implementación de aplicaciones de aprendizaje del idioma desde la perspectiva pedagógica y sus limitaciones.

Tabla 2. Ventajas y limitaciones pedagógicas de los lenguajes de programación analizados

LENGUAJE	PRINCIPALES VENTAJAS PEDAGÓGICAS	PRINCIPALES LIMITACIONES PEDAGÓGICAS
JAVASCRIPT	Alta interactividad, soporte nativo para multimedia, facilita gamificación y aprendizaje colaborativo en tiempo real.	Dependencia de buenas prácticas de seguridad y rendimiento en aplicaciones complejas.
PYTHON	Excelente integración con inteligencia artificial, aprendizaje adaptativo y analítica del progreso del estudiante.	Menor rendimiento para aplicaciones altamente interactivas en tiempo real.
JAVA	Robustez, estabilidad y soporte para plataformas educativas institucionales de gran escala.	Mayor complejidad de desarrollo y menor flexibilidad para prototipado rápido.
PHP	Facilidad de implementación, bajo costo y amplia disponibilidad en entornos educativos básicos.	Limitada capacidad nativa para interactividad avanzada sin tecnologías complementarias.
C# (.NET)	Alto rendimiento, seguridad y soporte para analítica avanzada en plataformas educativas corporativas.	Dependencia del ecosistema Microsoft y menor adopción en contextos educativos abiertos.

Esta tabla sintetiza los aportes pedagógicos y las restricciones identificadas en la literatura analizada, con el fin de complementar el análisis técnico presentado en la Tabla 1. Su inclusión responde a la necesidad de integrar criterios educativos en la evaluación de tecnologías de desarrollo web, conforme a los planteamientos de Laurillard (2012).

DISCUSIONES:

Los resultados obtenidos en el presente estudio confirman hallazgos previamente reportados en la literatura sobre ingeniería de software y tecnología educativa, evidenciando que la selección del lenguaje de programación incide directamente tanto en el rendimiento técnico como en la efectividad pedagógica de las aplicaciones web para la enseñanza de idiomas.

En concordancia con Sommerville (2016) y Pressman (2014), los lenguajes que cuentan con ecosistemas maduros y frameworks consolidados —como JavaScript, Java y C#— presentan mayores

ventajas en términos de mantenibilidad, escalabilidad y soporte a largo plazo. Esto resulta particularmente relevante en plataformas educativas que deben garantizar disponibilidad continua y evolución funcional conforme crece la base de usuarios.

Desde la perspectiva de la interactividad y la experiencia del usuario, los resultados respaldan lo expuesto por Laurillard (2012), quien sostiene que el aprendizaje mediado por tecnología debe centrarse en la retroalimentación inmediata y la participación activa del estudiante. En este sentido, JavaScript se posiciona como el lenguaje más adecuado para la implementación de estrategias pedagógicas basadas en gamificación, simulaciones lingüísticas y comunicación síncrona.

Por otro lado, el papel de Python en aplicaciones de enseñanza de idiomas se alinea con estudios recientes sobre aprendizaje adaptativo y analítica de aprendizaje, donde se destaca la importancia de la inteligencia artificial y el procesamiento de lenguaje natural para personalizar la experiencia educativa (Okoli, 2015). La capacidad de Python para integrarse con modelos de recomendación y evaluación automática refuerza su valor pedagógico, aun cuando su rendimiento puro sea inferior al de lenguajes compilados.

Asimismo, Java y C# demuestran ser opciones sólidas para entornos institucionales, como universidades o sistemas educativos gubernamentales, donde la robustez, la seguridad y la gestión de grandes volúmenes de datos son prioritarias. Este hallazgo coincide con lo señalado por Kitchenham y Charters (2007), quienes enfatizan que la elección tecnológica debe responder al contexto y los objetivos del sistema.

En contraste, PHP mantiene su vigencia en proyectos educativos de menor escala, aunque su dependencia de tecnologías complementarias para lograr altos niveles de interactividad limita su aplicación en entornos de aprendizaje lingüístico avanzados.

En síntesis, la discusión evidencia que los resultados empíricos de la revisión sistemática se encuentran sólidamente respaldados por la literatura científica analizada, reforzando la validez de la metodología empleada y la pertinencia de los indicadores seleccionados.

4. CONCLUSIONES

La selección del lenguaje de programación para aplicaciones web de enseñanza de idiomas debe basarse en un análisis integral que considere tanto aspectos técnicos como pedagógicos. No existe una solución única, sino múltiples alternativas que pueden adaptarse a diferentes contextos educativos.

JavaScript se posiciona como la opción más versátil para aplicaciones interactivas, mientras que Python resulta clave para la incorporación de inteligencia artificial. Java y C# son recomendables para plataformas a gran escala, y PHP sigue siendo una alternativa válida para proyectos educativos básicos.

Futuras investigaciones podrían profundizar en estudios empíricos que evalúen el impacto directo del lenguaje de programación en los resultados de aprendizaje.

FINANCIACIÓN

Los autores declaramos que la investigación no tuvo financiamiento.

CONFLICTO DE INTERESES

Los Autores declaramos que no existe conflicto de intereses con nuestra investigación.

CONTRIBUCIÓN DE AUTORÍA

En concordancia con la taxonomía establecida internacionalmente para la asignación de créditos a autores de artículos científicos (<https://credit.niso.org/>). Los autores declaran sus contribuciones en la siguiente matriz:

	Autor 1.	Autor 2	Autor 3	Autor 4
Participar activamente en:				
Conceptualización	X	X	X	X
Análisis formal	X	X	X	X
Adquisición de fondos	X	X	X	X
Investigación	X	X	X	X
Metodología	X	X	X	X
Administración del proyecto	X	X	X	X
Recursos	X	X	X	X
Redacción –borrador original	X	X	X	X
Redacción –revisión y edición	X	X	X	X
La discusión de los resultados	X	X	X	X
Revisión y aprobación de la versión final del trabajo.	X	X	X	X

REFERENCIAS

Hasumi, T. (2024). Technology-enhanced language learning in English and language acquisition. *Technology, Language and Education*.

Valencia, Y. S., & Serrano Sánchez, J. L. (2021). Aprendizaje de un segundo idioma apoyado en tecnologías digitales: una revisión sistemática. *Education in the Knowledge Society*, 22.

Polyezhayev, Y., Terletska, L., Kulichenko, A., Vorobiova, L., & Snizhko, N. (2024). El papel de las aplicaciones web en el desarrollo de la competencia multilingüe en los cursos AICLE. *Eduweb*, 18(1).

Tejera-Martínez, F., Aguilera, D., & Vilchez-González, J. M. (2020). Lenguajes de programación y desarrollo de competencias clave: una revisión sistemática. *Revista Electrónica de Investigación Educativa*, 22.

Opryshko, N., Novik, K., Smolinska, O., Burmakina, N., & Aizikova, L. (2024). Web-based applications in higher education: Transforming language learning in the digital era. *Amazonia Investiga*, 13(73), 209–219.

Ortiz, R. S., & Carrió Pastor, M. L. (2024). Classification of technology-based language learning resources according to pedagogical and functional criteria. *RAEL*, 23, 19–33.

Raine, P. (2024). Developing web-based English learning applications: Principles and practice. *Computer-Assisted Language Learning Electronic Journal*.

Shadiev, R., et al. (2022). A review of research on technology-supported language learning and 21st century skills. *Frontiers in Education*, 7.

Sarica, G. N. (2008). Web-based English language learning. *ERIC Digest*.

Herrera, L. (2018). Duolingo language-learning platform and computer-assisted language learning. *Repositorio UTA*.

Layedra Larrea, N. P., Ramos Valencia, M. V., & Salazar Cazco, S. A. (2022). Análisis de los lenguajes de programación más utilizados en aplicaciones web y móviles. *Dominio de las Ciencias*, 8(3).

Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.

Pressman, R. S., & Maxim, B. R. (2014). *Software engineering: A practitioner’s approach* (8th ed.). McGraw-Hill.

Laurillard, D. (2012). *Teaching as a design science*. Routledge.

Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. *EBSE Technical Report*.

Okoli, C. (2015). A guide to conducting a standalone systematic literature review. *Communications of the Association for Information Systems*, 37.

Mozilla Foundation. (2024). *JavaScript documentation*.

Microsoft. (2024). *ASP.NET Core documentation*.

Django Software Foundation. (2024). *Django documentation*.

Oracle. (2023). *Java platform documentation*.