

Metodología para mejorar la programación con Inteligencia Artificial

Methodology to Improve Programming with Artificial Intelligence

Elizabeth Patricia Pommier Gallo

Universidad Mayor de San Andrés, Postgrado en Informática. La Paz, Bolivia

<https://orcid.org/0009-0002-3316-6771>

epommierg@fcpn.edu.bo

Fecha de recepción: 13-03-2024

Fecha de aprobación: 29-04-2024

Fecha de publicación: 05-06-2024

Cómo citar este artículo/Citation: Pommier Gallo, E. P. (2024). Metodología para mejorar la programación con Inteligencia Artificial. *Actas Iberoamericanas En Ciencias Sociales*, 2(1), 86-97. <https://plagcis.com/journal/index.php/aicis/article/view/28>

Resumen:

Este artículo explora el potencial de las tecnologías de inteligencia artificial para mejorar y personalizar la enseñanza de la programación a nivel universitario. Se discuten conceptos como la personalización avanzada, las recomendaciones contextuales y las experiencias interactivas para involucrar más a los estudiantes. También se presentan algunas herramientas de IA que pueden usarse como instrumentos de programación para facilitar el aprendizaje. Una revisión de la literatura reciente indica un creciente interés en la aplicación de técnicas como el aprendizaje adaptativo, la realidad aumentada y el procesamiento de lenguaje natural en este campo.

Palabras clave: Inteligencia artificial, enseñanza de programación, personalización de aprendizaje, entornos interactivos

Copyright: © 2024 AICIS. Este es un artículo de acceso abierto distribuido bajo los términos de la licencia de uso y distribución Creative Commons Reconocimiento 4.0 Internacional (CC BY 4.0).



Abstract

This article explores the potential of artificial intelligence technologies to enhance and personalize programming education at the university level. Concepts such as advanced personalization, contextual recommendations, and interactive experiences are discussed to further engage students. Additionally, several AI tools that can be used as programming instruments to facilitate learning are presented. A review of recent literature indicates a growing interest in the application of techniques such as adaptive learning, augmented reality, and natural language processing in this field.

Keywords: Artificial intelligence, programming education, personalized learning, interactive environments

Copyright: © 2024 AICIS. This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0).

1. INTRODUCCIÓN

El uso de la IA y el aprendizaje automático en la educación en programación ya ha mostrado resultados prometedores. Al proporcionar recomendaciones contextuales, las plataformas pueden mejorar la experiencia y la relevancia del usuario. Por ejemplo, los estudiantes pueden recibir comentarios personalizados sobre sus tareas de codificación, y el sistema analiza su trabajo y ofrece sugerencias de mejora basadas en sus fortalezas y debilidades específicas (Bennedsen, y Caspersen, 2007; Watson, y Li 2014; González Ciriaco y Medina Marín, 2023; Román-Acosta, 2024; Román Acosta et al., 2023).

La personalización basada en IA también permite adaptar las ofertas a las necesidades de cada usuario, haciendo que el proceso de aprendizaje sea más eficiente y efectivo (Bornat et al., 2008). Estas recomendaciones pueden ayudar a los estudiantes a mantenerse motivados y comprometidos, ya que reciben apoyo y orientación personalizados durante su educación en programación.

Entre los factores que inciden en esta problemática se encuentran los diversos estilos y ritmos de aprendizaje entre los estudiantes, la curva inicial empinada en la comprensión de conceptos abstractos, y la falta de una retroalimentación personalizada (Andrist et al., 2014).

Las tecnologías de inteligencia artificial (IA) ofrecen potentes herramientas para abordar muchas de estas dificultades. Técnicas como el aprendizaje adaptativo (VanLehn, 2011), el procesamiento de lenguaje natural (NLP) (Mishra y Singh, 2021), la visión artificial y la realidad extendida (Bower et al., 2014; Singh et al., 2013) pueden reinventar la forma en que los estudiantes de pregrado aprenden a programar. Este

artículo presenta un panorama de estas innovaciones y su potencial transformador (Brown y Altadmri, 2017; Fernández Miranda et al., 2024).

2. METODOLOGÍA

La personalización en la programación se ha vuelto cada vez más importante en los últimos años, con el auge de la tecnología de inteligencia artificial (IA) y aprendizaje automático (ML) (Mishra y Singh, 2021). La IA y el ML permiten la creación de experiencias de usuario altamente personalizadas ajustando el contenido, esta personalización avanzada tiene muchas aplicaciones potenciales en la programación de pregrado, lo que permite a los educadores crear experiencias de aprendizaje personalizadas que satisfagan las necesidades y preferencias únicas de cada estudiante.

El uso de programación personalizada con IA y ML en la educación universitaria ya ha comenzado a revolucionar la forma en que los estudiantes aprenden. Las recomendaciones contextuales, las experiencias interactivas y la personalización avanzada del aprendizaje son posibles con el uso de análisis y automatización en la educación superior, la IA generativa tiene el potencial de crear experiencias educativas personalizadas mediante el desarrollo de simulaciones emocionales realistas, la implementación de estas tecnologías también presenta desafíos, como garantizar la privacidad y seguridad de los datos.

Programar IA requiere conocimientos de lenguajes de programación, algoritmos de comportamiento humano y matemáticas, los lenguajes de programación populares para desarrollar IA incluyen Python, que se utiliza para el aprendizaje automático y el procesamiento del lenguaje natural, al aprender estas habilidades, los programadores pueden contribuir al desarrollo de una personalización avanzada en la programación de pregrado, ayudando a crear experiencias de aprendizaje más efectivas y atractivas para los estudiantes. (Román-Acosta et al., 2024).

2.1. Recomendaciones contextuales en programación de IA

Es decir, las recomendaciones se adaptan a las necesidades y preferencias específicas del usuario en un momento determinado. Los algoritmos de aprendizaje automático se utilizan comúnmente para crear estas recomendaciones, analizando datos como el historial de navegación del usuario, las consultas de búsqueda y la ubicación, el uso de recomendaciones contextuales se ha vuelto cada vez más popular en diversas industrias, incluida la educación. En la programación de pregrado, se pueden utilizar recomendaciones contextuales para brindar a los estudiantes experiencias de aprendizaje personalizadas que se adapten a sus necesidades y preferencias individuales.



Al proporcionar recomendaciones contextuales, las plataformas pueden mejorar la experiencia y la relevancia del usuario. Por ejemplo, los estudiantes pueden recibir comentarios personalizados sobre sus tareas de codificación, y el sistema analiza su trabajo y ofrece sugerencias de mejora basadas en sus fortalezas y debilidades específicas. Estas recomendaciones pueden ayudar a los estudiantes a mantenerse motivados y comprometidos, ya que reciben apoyo y orientación personalizados durante su educación en programación.

A medida que la era de la inteligencia artificial continúa evolucionando, es crucial que las instituciones de educación superior exploren las oportunidades y desafíos que presenta la IA para la educación en programación (Bower et al., 2014).

2.2. Experiencias interactivas en programación de IA

Las experiencias pueden ser utilizadas en una variedad de campos, incluyendo la educación, y pueden ser especialmente útiles en la programación de pregrado. Al involucrar a los estudiantes en experiencias interactivas, se les brinda la oportunidad de aprender de manera más efectiva y personalizada. La realidad virtual y aumentada pueden proporcionar experiencias de aprendizaje más inmersivas e interactivas que pueden ayudar a los estudiantes a comprender mejor los conceptos difíciles. Al mismo tiempo, el aprendizaje automático puede ser utilizado para proporcionar recomendaciones personalizadas y sugerencias de contenido contextual (Gulwani, 2017). Al utilizar estas técnicas, los estudiantes pueden obtener una comprensión más profunda de los conceptos y mejorar su capacidad para aplicarlos en situaciones del mundo real (Park, 2003). Las experiencias interactivas pueden ser una herramienta valiosa para mejorar el aprendizaje en la programación de pregrado y preparar a los estudiantes para el éxito en el mundo laboral (Pu y Narasimhan, 2022; Leeds et al., 2020; Mishra et al., 2022).

2.3. IA Desarrollo de la metodología de programación con IA

El modelo delinea cuatro fases: (1) preparación, (2) desarrollo e integración, (3) implementación controlada, y (4) despliegue a gran escala y mejora continua. También explora áreas específicas como la personalización avanzada, las recomendaciones contextuales, las experiencias interactivas y los instrumentos de IA para programación. Se espera que este modelo metodológico informe futuras iniciativas para modernizar la enseñanza de programación con tecnologías de vanguardia.

2.4. Fases

Las fases de la metodología para implementar IA en la enseñanza programación a nivel de pregrado:

Fase 1: Preparación

- ❖ Evaluación de necesidades
- ❖ Realizar encuestas y entrevistas con estudiantes y docentes para identificar áreas problemáticas y oportunidades de mejora
- ❖ Analizar rendimiento académico actual a partir de evaluaciones y tasas de reprobación
- ❖ Detectar brechas de conocimiento, motivación y compromiso estudiantil

- ❖ Realizar sondeos tecnológicos para determinar disponibilidad de infraestructura
- ❖ Definición de objetivos
- ❖ Delinear objetivos y métricas concretas de mejora: notes, reprobación, motivación, adquisición de competencias técnicas y blandas, etc.
- ❖ Esbozar casos de uso para implementaciones de IA
- ❖ Definir alcance y limitaciones iniciales
- ❖ Crear marco de evaluación para pilotos y despliegue a gran escala
- ❖ Análisis del panorama tecnológico:
- ❖ Mapear soluciones comerciales y de investigación existentes
- ❖ Estudiar literatura sobre implementaciones de entornos virtuales y aumentados para programación
- ❖ Evaluar opciones open source para personalización adaptable
- ❖ Analizar herramientas de IA para autocompletado de código, traducción lenguaje natural a código y generación/optimización de software

Fase 2: Desarrollo e integración

En cada área se describen actividades como diseño conceptual y técnico, desarrollo de prototipos, evaluación interna, refinamiento iterativo, integración con IDEs de programación o sistemas de gestión de aprendizaje existentes, creación de materiales pedagógicos asociados, capacitación a docentes

Fase 3: Implementación controlada

Se dan detalles sobre actividades de lanzamiento de pilotos con grupos pequeños de estudiantes, protocolos de implementación, guías para docentes, conjunto de herramientas para estudiantes sobre nuevas soluciones tecnológicas, canales de retroalimentación y sugerencias, evaluación en aula con muestras limitadas, análisis de rendimiento, identificación de ajustes y mejoras.

Fase 4: Mejora continua

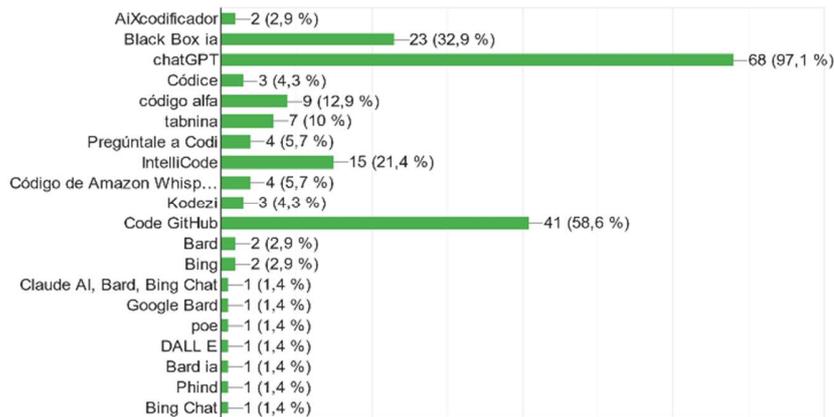
En esta etapa se incluye recopilación de analíticas de uso y rendimiento académico, identificación de nuevas mejoras y actualizaciones, incorporación de nuevas tendencias e innovaciones tecnológicas recientes.

2.5. Implementación

Se realizó la implementación de la metodología y se encuestó a 210 estudiantes respondieron al cuestionario online con 25 preguntas sobre el uso de las IAs que se implementó con las fases propuestas en diferentes materias de programación de primer hasta tercer semestre para minimizar las dificultades, se muestran algunas preguntas que dieron los siguientes resultados:

Figura 1

Cuales IAs a utilizado en las clases, según la materia que cursa, puede elegir varias



3. RESULTADOS

Se puede evidenciar que los estudiantes de las diferentes materias han utilizado en varios laboratorios el uso de IAs con supervisión, dando como mejor resultado el chatGPT, como el que lleva más usabilidad con la preferencia.

Figura 2.

¿Qué estrategias consideras más efectivas para maximizar la utilidad de IAs de programación en el ámbito de la programación?



El 37.1% que para el uso de IAs se debe proporcionar instrucciones claras y detalladas, para una mejor comprensión, además de asegurar un contexto preciso para las consultas, con el uso de formatos de preguntas específicos y directos que den una mejor respuesta.

Figura 3.

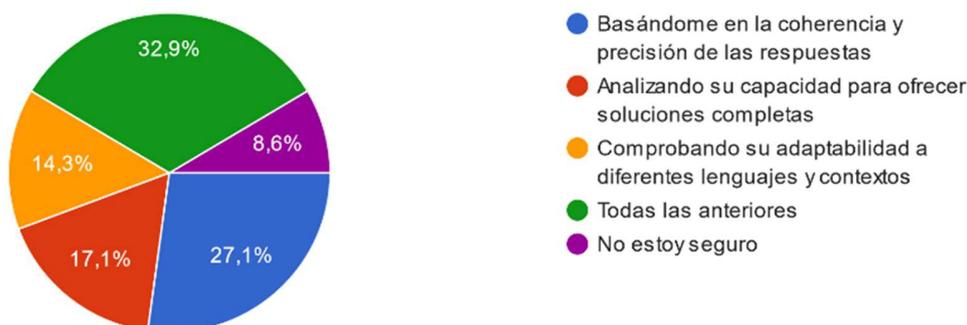
¿Cuáles son las mejores prácticas que consideras importantes al utilizar IAs en el desarrollo de proyectos?



Se evidencia que el 50% está consciente de utilizar las IAs de programación como un complemento, y no como un sustituto de habilidades propias, es decir lo común de copiar y pegar, así se va fortaleciendo la para ética en el uso.

Figura 4.

¿Cómo evaluarías la efectividad IAs de programación en entender y responder preguntas específicas sobre bibliotecas, frameworks o herramientas de programación?



También se realizó preguntas acerca del funcionamiento y cuál es la perspectiva de los estudiantes respecto a la efectividad de las IAs, viendo la eficiencia y las respuestas que han proporcionado según el uso que se les ha planteado y viendo la adaptabilidad con diferentes lenguajes.

Figura 5.

¿En qué medida LAs de programación puede ayudar a identificar y solucionar errores comunes de programación?



Desde la perspectiva de la identificación de errores en la programación, un 38.6% las utiliza para esto, el 15.7% para dar soluciones específicas de código, al igual que la depuración para mejorar el código.

Figura 6.

¿Cuál de las siguientes áreas consideras que LAs podría asistir mejor?



Las áreas más utilizadas son la revisión de código sin embargo igual consideran que tanto la revisión y generación código son áreas donde se utilizan más a las IAs. Al identificar las fortalezas y debilidades de cada estudiante, la IA puede adaptar las experiencias de aprendizaje para satisfacer sus necesidades específicas, lo que resulta en mayores niveles de compromiso y motivación. esta personalización también puede ayudar a los estudiantes a aprender de manera más eficiente, ya que pueden concentrarse en las áreas que requieren más atención.

La IA aumento la participación de los estudiantes al proporcionar experiencias interactivas, los estudiantes reciben comentarios en tiempo real sobre su progreso, lo que ayudo a identificar áreas en las que necesitan mejorar. Esta retroalimentación ayudo a identificar qué temas requieren mayor explicación.

Al utilizar la IA se creó experiencias interactivas, para que estén más motivados durante todo el proceso de aprendizaje, lo que condujo a mejores resultados de aprendizaje.

Esta mayor eficiencia condujo a una experiencia educativa más eficaz y optimizada tanto para los estudiantes. Implementar IA para proporcionar las recomendaciones contextuales para materiales y ejercicios complementarios basados en el tema actual que se está estudiando., ayudo a mejorar el desempeño en clases. Las IAs crearon experiencias de aprendizaje interactivas que se adaptaron al progreso y desempeño de los estudiantes.

Utilizar IA para facilitar experiencias de aprendizaje colaborativo ayudo a unir estudiantes con niveles de habilidades similares o fortalezas complementarias. La IA ayudo a recomendar proyectos grupales o actividades de programación en parejas, fomentando un entorno colaborativo donde los estudiantes aprendieron unos de otros y desarrollar habilidades esenciales de trabajo en equipo, que son cruciales en la industria del desarrollo de software.

4. DISCUSIÓN

La integración de la IA en la programación universitaria plantea varios desafíos, incluido cómo integrar eficazmente esta tecnología con los métodos de enseñanza tradicionales. Si bien la IA tiene el potencial de transformar la educación y personalizar las experiencias de aprendizaje, es esencial garantizar que complemente y mejore las prácticas pedagógicas existentes. El juicio humano y la comprensión contextual son cruciales en la toma de decisiones éticas, áreas en las que la IA puede quedarse corta a pesar de sus capacidades avanzadas.

Los modelos y algoritmos de IA requieren una importante potencia computacional y capacidades de almacenamiento de datos, que pueden estar más allá de los recursos de algunas instituciones educativas. Además, el desarrollo y mantenimiento de sistemas de IA requieren conocimientos técnicos especializados, que pueden no estar fácilmente disponibles en todos los entornos educativos. Los educadores deben considerar estos desafíos técnicos y sopesar los costos y beneficios de incorporar la IA en su programación.

Personalización avanzada de las IA en la programación universitaria pueden proporcionar experiencias de aprendizaje personalizadas, adaptadas al ritmo, estilo y nivel de comprensión de cada estudiante, ofrecer recomendaciones específicas del contexto a los estudiantes, sugiriendo recursos o ejercicios adicionales basados en su progreso y desafíos de aprendizaje actuales, las IA pueden crear simulaciones o ejercicios interactivos, mejorando la participación de los estudiantes y la comprensión de conceptos de programación complejos. Las IA en la programación de pregrado ofrecen personalización avanzada, recomendaciones contextuales y experiencias interactivas, lo que mejora el proceso de aprendizaje.

5. CONCLUSIONES

La integración de la IA en la programación universitaria ha abierto nuevas oportunidades para la personalización avanzada, recomendaciones contextuales y experiencias interactivas. Estos beneficios han dado lugar a mejores resultados de aprendizaje, una mayor participación de los estudiantes y una enseñanza y un aprendizaje más eficientes.

Sin embargo, también hay desafíos a considerar, como la integración con los métodos de enseñanza tradicionales, consideraciones éticas y desafíos técnicos. A medida que la tecnología de IA continúa avanzando, existe la posibilidad de experiencias de aprendizaje aún más mejoradas en el futuro. Es importante que los educadores se mantengan informados y se adapten a estos cambios para brindar las mejores oportunidades de aprendizaje a sus estudiantes.

6. REFERENCIAS

- Andrist, S., Tan, X. Z., Gleicher, M., Mutlu, B. (2014). Conversational gaze aversion for humanlike robots. In Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction (pp. 25–32).
- Bennedsen, J., y Caspersen, M. E. (2007). Failure rates in introductory programming. *ACM SIGCse Bulletin*, 39(2), 32-36.
- Bornat, R., Dehnadi, S., y Simon. (2008). Mental models, consistency and programming aptitude. In Proceedings of the tenth conference on Australasian computing education-Volume 78 (pp. 53-61).
- Bower, M., Howe, C., McCredie, N., Robinson, A., y Grover, D. (2014). Augmented Reality in education—cases, places and potentials. *Educational Media International*, 51(1), 1-15.
- Brown, N.C., Altadmri, A. (2017). Investigating novice programming mistakes: educator beliefs vs. student data. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (pp. 201–206).
- Fernández Miranda, M., Román Acosta, D., Jurado Rosas, A. A., Limón Domínguez, D., & Torres Fernandez, C. (2024). Artificial Intelligence in Latin American Universities: Emerging Challenges. *Computación y Sistemas*, 28(2). <https://doi.org/10.13053/CyS-28-2-4822>

- González Ciriaco, L. A. ., & Medina Marín, A. J. (2023). Avances y desafíos éticos en la integración de la IA en la producción científica. *Journal of Scientific Metrics and Evaluation*, 1(1), 48-67. <https://doi.org/10.69821/JoSME.v1i1.2>
- Gulwani, S. (2017). Automating programming education: A step towards meeting society's tech challenges. *Communications of the ACM*, 61(8), 28-30.
- Leeds, D., West, J., Lee, L. (2020). Object Oriented Learning Assistant: Using NLP to Help Novices While They Code. arXiv preprint arXiv:2010.12697.
- Mishra, A., Huang, H., Znati, T., Diana, C., Mooty, M., Gandhe, A., ... y Singh, V. (2022). NL2Code: Generating Code
- Mishra, A., y Singh, V. (2021). Natural language programming. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (pp. 1650-1654).
- Park, O. C., y Lee, J. (2003). Adaptive instructional systems. *Educational Technology Research and Development*, 51(4), 63-84.
- Pu, X., Narasimhan, K. (2022). Explanations can improve predictive models of student errors in programming. In *Proceedings of the Seventh ACM Conference on Learning@ Scale* (pp. 353–358).
- Román Acosta, D. D., Alarcón Osorio, D., y Rodríguez Torres, E. (2023). Implementación de ChatGPT: aspectos éticos, de edición y formación para estudiantes de posgrado. *Revista Senderos Pedagógicos*, 15(1), 15–31. <https://doi.org/10.53995/rsp.v15i1.1592>
- Román-Acosta, D. D., Rodríguez Torres, E., Baquedano Montoya, M. B., López Zavala & L. C., Pérez Gamboa, A. J., (2024). ChatGPT and its use to improve academic writing in postgraduate students. *PRA*, 24(36), 53–75. <https://doi.org/10.26620/uniminuto.praxis.24.36.2024.53-75>
- Román-Acosta, D. (2024). Exploración filosófica de la epistemología de la inteligencia artificial: Una revisión sistemática. *Uniandes Episteme*, 11(1), 101–122. <https://doi.org/10.61154/rue.v11i1.3388>
- Singh G., Gulwani S., Solar-Lezama A. (2013). Automated feedback generation for introductory programming assignments. In *Proceedings of the 34th ACM SIGPLAN conference on Programming language design and implementation* (pp. 15–26).

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4), 197-221.

Watson, C., y Li, F. W. (2014). Failure rates in introductory programming revisited. In *Proceedings of the 2014 conference on Innovation & technology in computer science education* (pp. 39-44).

SOBRE LA AUTORA

Elizabeth Patricia Pommier Gallo

Licenciada en Informática por la Universidad Mayor de San Andrés y con un Máster en Formación Docente e Innovación Educativa de la Universidad de Barcelona, nuestra experta es miembro activo de la Asociación Mundial de Tutores Virtuales (ASOMTV). Ha realizado diversas especializaciones en FATLA, incluyendo Educación Virtual, Tecnología Educativa, Entornos Educativos 3D, Administración Web, Comercio Electrónico, Periodismo Digital, Smart Working, Plataformas e-Learning y Aulas Virtuales. Su formación también incluye un Diplomado en Especialización en el Diseño y Desarrollo de Cursos Virtuales de IT Madrid IT Business y la Universidad Católica Boliviana San Pablo (UCB), un Diplomado en Didáctica y Evaluación de las Ciencias Exactas por la Universidad de Harvard LASPAU y UCB, y un Diplomado en Machine Learning y Ciencia de Datos de la Universidad Privada Boliviana (UPB). Actualmente, se desempeña como Directora de Experto en Educación Virtual en FATLA y Coordinadora de Formación en el Centro de Ciberseguridad de Bolivia. Su trayectoria docente se extiende a universidades como la Universidad Mayor de San Andrés (UMSA), la Universidad Católica Boliviana (UCB), la Universidad Privada Boliviana (UPB) y la Universidad Privada Domingo Savio (UPDS), donde ha sido tutora, relatora y miembro de tribunales en defensas de pregrado y posgrado. Además, ha impartido clases en varias maestrías y diplomados en Posgrado en Informática en UMSA, UCB y la Escuela Militar de Ingeniería (EMI).

Conflicto de interés

No existe ningún conflicto de interés

Financiamiento: