



Ciencia Latina Revista Científica Multidisciplinar, Ciudad de México, México.
ISSN 2707-2207 / ISSN 2707-2215 (en línea), mayo-junio 2025,
Volumen 9, Número 3.

https://doi.org/10.37811/cl_rcm.v9i1

**PERCEPCIÓN DEL USO DE PYCHARM EN ESTUDIANTES
DE PRIMER Y SEGUNDO SEMESTRE EN LA CARRERA DE
TECNOLOGÍAS DE LA INFORMACIÓN**

PERCEPTION OF PYCHARM USAGE AMONG FIRST AND
SECOND-SEMESTER STUDENTS IN THE INFORMATION
TECHNOLOGY

Walter Rodrigo Nuñez Zamora

Universidad Estatal Amazónica

Hirám Hernández Ramos

Universidad Estatal Amazónica

Eberto Tuniesky Gutiérrez de León

Universidad Estatal Amazónica

Beivy Amarilis Rivera Vergara

Universidad Estatal Amazónica

DOI: https://doi.org/10.37811/cl_rem.v9i3.18282

Percepción del uso de PyCharm en Estudiantes de Primer y segundo semestre en la carrera de Tecnologías de la Información

Walter Rodrigo Nuñez Zamora¹walternza@gmail.com<https://orcid.org/0009-0007-7289-5914>

Universidad Estatal Amazónica

Hirám Hernández Ramoshiram03021@gmail.com<https://orcid.org/0009-0000-6722-869X>

Universidad Estatal Amazónica

Eberto Tuniesky Gutiérrez de Leóntunieskygutierrez@gmail.com<https://orcid.org/0000-0001-7682-9657>

Universidad Estatal Amazónica

Beivy Amarilis Rivera Vergarariverabeivy@gmail.com<https://orcid.org/0009-0002-4806-6247>

Universidad Estatal Amazónica

RESUMEN

El presente documento analizó la percepción del uso de PyCharm en estudiantes de primer y segundo semestre de la carrera de Tecnologías de la Información en la Universidad Estatal Amazónica. El objetivo fue evaluar cómo el uso de PyCharm impacta en el aprendizaje de conceptos de programación, la productividad y el rendimiento académico de los estudiantes. Se realizó un estudio cuantitativo de tipo descriptivo y comparativo, mediante una encuesta estructurada aplicada a 246 estudiantes. Los resultados mostraron que los estudiantes de segundo semestre percibieron significativamente más positiva la herramienta que los de primer semestre, particularmente en cuanto a la facilidad de uso y la familiaridad con las funciones de PyCharm. No se encontraron diferencias significativas en las percepciones según el género. En general, los estudiantes reconocieron que el uso de PyCharm les ayudó a mejorar su comprensión de los conceptos de programación, destacando características como la depuración y el autocompletado de código. Se concluye que PyCharm es una herramienta valiosa en la enseñanza de la programación y su uso prolongado potencia la adquisición de competencias técnicas necesarias para el ámbito profesional.

Palabras clave: PyCharm, percepción de estudiantes, enseñanza de programación, comprensión de la programación, tecnologías de la información

¹ Autor principal

Correspondencia: walternza@gmail.com

Perception of PyCharm Usage Among First and Second-Semester Students in the Information Technology

ABSTRACT

This paper analyzed the perception of PyCharm usage among first and second-semester students of the Information Technology degree at the Universidad Estatal Amazónica. The objective was to assess how PyCharm impacts the learning of programming concepts, productivity, and academic performance. A quantitative, descriptive, and comparative study was conducted through a structured survey applied to 246 students. The results showed that second-semester students had a significantly more positive perception of the tool than first-semester students, particularly regarding ease of use and familiarity with PyCharm's features. No significant differences in perceptions were found based on gender. Overall, students acknowledged that using PyCharm helped improve their understanding of programming concepts, highlighting features such as debugging and code completion. It is concluded that PyCharm is a valuable tool in programming education, and its prolonged use enhances the acquisition of technical skills necessary for the professional field.

Keywords: PyCharm, student perception, programming teaching, understanding programming, information technologies

Artículo recibido 12 abril 2025
Aceptado para publicación: 17 mayo 2025



INTRODUCCIÓN

Los entornos de desarrollo integrados (IDEs) han desempeñado un papel fundamental en la evolución de la programación y la enseñanza de las tecnologías de la información. Los primeros intentos por crear herramientas que facilitasen la codificación datan de las décadas de 1960 y 1970, cuando se empezaron a desarrollar compiladores y depuradores rudimentarios que ayudaban a los programadores a manejar lenguajes de programación emergentes como COBOL y FORTRAN (Salinas et al., 2023). A medida que los lenguajes de programación se volvieron más complejos, las herramientas que los soportaban también evolucionaron. En las décadas de 1980 y 1990, surgieron las primeras interfaces gráficas de usuario, lo que permitió a los desarrolladores interactuar más fácilmente con el código y gestionar proyectos más grandes. IDEs como Turbo Pascal, lanzado en 1983, y Microsoft Visual Studio en los años 90, marcaron un hito al integrar el editor de texto, el compilador y el depurador en un solo entorno, lo que transformó la experiencia de programación y la hizo más accesible para nuevos usuarios (Lukkarinen et al., 2021).

En la actualidad, los IDEs han evolucionado hacia plataformas aún más sofisticadas, diseñadas para soportar lenguajes de programación modernos y ofrecer una amplia gama de herramientas integradas que mejoran la productividad. IDEs como PyCharm han ganado prominencia en la última década debido a su enfoque en lenguajes específicos como Python, un lenguaje clave en el desarrollo de software, análisis de datos, e inteligencia artificial (Buckner et al., 2023). PyCharm, desarrollado por JetBrains, no solo facilita la escritura de código con características como autocompletado y resaltado de sintaxis, sino que también permite la integración con sistemas de control de versiones, entornos virtuales y herramientas de testing, lo que lo convierte en una opción ideal tanto para principiantes como para programadores avanzados. Además, la capacidad de PyCharm de integrarse con diversas tecnologías emergentes ha potenciado su uso en el ámbito académico, particularmente en la enseñanza de programación, donde el uso de herramientas eficaces puede marcar una diferencia significativa en la adquisición de competencias tecnológicas (Xu et al., 2021).

Los entornos de desarrollo integrados (IDEs) juegan un papel crucial en la enseñanza de la programación, especialmente en los primeros niveles de formación. Los IDEs proporcionan un entorno controlado y estructurado en el que los estudiantes pueden escribir, probar y depurar su código de



manera eficiente. Al ofrecer herramientas visuales como el resaltado de sintaxis, la autocompletación de código y las alertas en tiempo real de errores, los IDEs ayudan a los estudiantes a identificar problemas en su código de manera inmediata, lo que facilita el proceso de aprendizaje (Salinas et al., 2023). En lugar de enfrentar las frustraciones de encontrar errores manualmente, los estudiantes pueden concentrarse en comprender los principios fundamentales de la programación. Estas herramientas no solo reducen la carga cognitiva asociada con los errores sintácticos, sino que también promueven la experimentación y el autoaprendizaje, aspectos vitales en la educación tecnológica.

Además, los IDEs permiten a los estudiantes familiarizarse con las prácticas de desarrollo profesional desde las primeras etapas de su formación. Con funcionalidades que incluyen la integración de sistemas de control de versiones, testing automatizado y soporte para múltiples lenguajes de programación, los estudiantes adquieren competencias que serán esenciales en su futuro profesional (Noor et al., 2023a). PyCharm, por ejemplo, es un IDE ampliamente utilizado en la enseñanza de lenguajes como Python, y su capacidad para ofrecer un entorno de desarrollo completo lo convierte en una herramienta ideal en contextos educativos. El uso de PyCharm en la enseñanza de la programación no solo mejora la comprensión de los conceptos básicos, sino que también expone a los estudiantes a flujos de trabajo profesionales, lo que incrementa su preparación para el mercado laboral. Por tanto, la adopción de IDEs como PyCharm en el aula no es solo una cuestión de facilitar el aprendizaje, sino de preparar a los futuros profesionales con las herramientas que se usan en la industria.

PyCharm es un entorno de desarrollo integrado (IDE) ampliamente reconocido por su enfoque en el lenguaje de programación Python, ofreciendo una serie de características que optimizan el flujo de trabajo de los programadores (Hliš et al., 2023). Una de las principales características de PyCharm es su sistema de autocompletado inteligente, que no solo sugiere posibles variables y funciones a medida que el programador escribe, sino que también es capaz de predecir estructuras complejas de código, lo que acelera significativamente el proceso de desarrollo. Además, PyCharm cuenta con un sistema de inspección de código en tiempo real que alerta al programador de errores sintácticos y lógicos mientras el código se escribe, lo que reduce el tiempo necesario para detectar y corregir fallos. Estas herramientas de asistencia no solo facilitan el desarrollo, sino que también ayudan a los estudiantes y profesionales a mejorar sus habilidades al ofrecerles retroalimentación inmediata sobre su código (Elvina et al., 2018).



Otra de las características clave de PyCharm es su capacidad para integrarse con entornos virtuales y sistemas de control de versiones, lo que permite a los usuarios gestionar proyectos de manera más eficiente y profesional. PyCharm ofrece soporte nativo para Git, lo que permite a los programadores realizar tareas de versionado, como commits, merges y revisión de código, directamente desde el IDE, sin necesidad de recurrir a herramientas externas. Esta integración es particularmente útil en el ámbito académico y profesional, ya que permite a los estudiantes y desarrolladores trabajar en proyectos colaborativos de manera organizada. Además, PyCharm facilita la creación y gestión de entornos virtuales de Python, lo que asegura que las dependencias de cada proyecto se mantengan aisladas y que no haya conflictos entre diferentes bibliotecas o versiones de Python. Este tipo de funcionalidad es crucial tanto en el desarrollo profesional como en la enseñanza, ya que introduce a los usuarios en buenas prácticas de programación desde las primeras etapas de su formación (Skorohodov & Kurochkina, 2021).

El uso de PyCharm en la educación de programación presenta varias ventajas que facilitan el aprendizaje y mejoran la experiencia del estudiante. Una de las principales ventajas de PyCharm es su interfaz intuitiva, que permite a los estudiantes concentrarse más en el desarrollo del código que en la configuración del entorno de trabajo. PyCharm proporciona herramientas como el autocompletado de código y el resaltado de errores en tiempo real, lo que reduce las frustraciones comunes que los principiantes suelen enfrentar al programar (Nikandrov, 2023). Estas características permiten a los estudiantes identificar errores rápidamente y corregirlos de inmediato, lo que les ayuda a comprender mejor los conceptos básicos de programación sin perder tiempo en depurar problemas simples. Además, la organización clara de proyectos y archivos facilita la gestión de tareas y proyectos más complejos, lo que es especialmente útil en el entorno académico donde los estudiantes trabajan en diferentes asignaciones simultáneamente.

Otra ventaja importante de PyCharm es su amplia gama de herramientas integradas que permiten a los estudiantes aprender buenas prácticas desde el principio de su formación. PyCharm no solo facilita la escritura de código, sino que también incluye herramientas para la depuración, pruebas automatizadas y control de versiones, lo que expone a los estudiantes a procesos que se utilizan comúnmente en el desarrollo profesional de software.



Al utilizar PyCharm, los estudiantes pueden aprender a depurar su código de manera estructurada, a ejecutar pruebas unitarias para asegurar la calidad del software y a colaborar con otros desarrolladores mediante sistemas de control de versiones como Git (Hu et al., 2021). Además, PyCharm permite la integración con otros servicios y tecnologías, como bases de datos y plataformas de desarrollo web, lo que amplía las oportunidades de aprendizaje y prepara a los estudiantes para enfrentar desafíos reales en el mundo laboral. Esto hace que PyCharm sea una herramienta invaluable en la educación de programación, tanto para principiantes como para estudiantes avanzados que buscan desarrollar habilidades aplicables en la industria tecnológica (Patil et al., 2023).

PyCharm se destaca frente a otros entornos de desarrollo integrados (IDEs) como Visual Studio Code y Sublime Text por su enfoque especializado en Python y su conjunto de herramientas integradas que abordan todas las etapas del ciclo de desarrollo. A diferencia de Visual Studio Code, que es un IDE altamente personalizable y puede ser configurado para soportar una gran variedad de lenguajes y entornos mediante extensiones, PyCharm está diseñado específicamente para mejorar la productividad en Python desde el inicio, sin necesidad de una configuración adicional (Xu et al., 2021). Esto lo convierte en una opción preferida para aquellos que buscan una herramienta lista para usar, que ofrezca características como autocompletado avanzado de código, inspección de código y depuración específica de Python (Buckner et al., 2023). Si bien Visual Studio Code puede adaptarse para igualar muchas de las características de PyCharm, el tiempo y esfuerzo requeridos para personalizarlo completamente pueden ser desventajas para los usuarios que buscan un entorno de desarrollo más especializado y dedicado (Zhang et al., 2023).

Las estrategias pedagógicas para enseñar programación han evolucionado a lo largo del tiempo, adaptándose a los cambios tecnológicos y a las necesidades de los estudiantes. Una de las estrategias más comunes es el aprendizaje basado en problemas, en el que los estudiantes aprenden a programar resolviendo problemas reales o simulados (Aires et al., 2023a). Esta técnica promueve un aprendizaje activo, donde los estudiantes aplican los conceptos teóricos a situaciones prácticas, lo que fortalece su comprensión y retención del material. Al enfocarse en resolver problemas, los estudiantes no solo adquieren habilidades técnicas, sino que también desarrollan habilidades de pensamiento crítico y resolución de problemas, que son esenciales en el ámbito de la programación (Omeh et al., 2022).



Además, esta estrategia fomenta la colaboración entre estudiantes, ya que muchos problemas requieren la cooperación de equipos, lo que simula ambientes laborales reales.

Además de los IDEs, otras herramientas tecnológicas como los simuladores y plataformas en línea han ganado popularidad en los primeros cursos de programación. Herramientas como repl.it y CodePen permiten a los estudiantes escribir y ejecutar código directamente en un navegador web, lo que elimina la necesidad de instalar software adicional y reduce las barreras técnicas para empezar a programar. Estas plataformas también facilitan la enseñanza a distancia y el trabajo colaborativo, ya que los estudiantes pueden compartir sus proyectos y recibir retroalimentación en tiempo real de sus compañeros y profesores (Rahman & Morgan, 2021). Estas herramientas tecnológicas permiten que los estudiantes de programación tengan acceso a un entorno de aprendizaje más dinámico y flexible, lo que resulta especialmente útil en los primeros semestres, donde es crucial mantener el interés y fomentar una comprensión sólida de los conceptos básicos (Macedo et al., 2022).

Los entornos de desarrollo integrados (IDEs) juegan un papel esencial en la adquisición de competencias de programación, ya que proporcionan a los estudiantes las herramientas necesarias para aprender de manera estructurada y eficiente. Al ofrecer características como el autocompletado de código, la depuración en tiempo real y la gestión automatizada de proyectos, los IDEs reducen la carga cognitiva que los estudiantes enfrentan cuando comienzan a programar. En este sentido, entornos como PyCharm destacan por ser una plataforma robusta que, desde el primer contacto, facilita la creación, edición y corrección de código en un entorno visual intuitivo, lo que acelera el proceso de aprendizaje y fortalece la adquisición de competencias básicas (Xu et al., 2021).

Además, los IDEs ayudan a los estudiantes a desarrollar competencias avanzadas que van más allá de escribir código correctamente. Herramientas como el control de versiones, la integración de pruebas automatizadas y la gestión de entornos virtuales que PyCharm proporciona son fundamentales para preparar a los estudiantes para desafíos del mundo real (Xu et al., 2021). Estas competencias no solo son esenciales en el contexto profesional, sino que también promueven una mayor comprensión de cómo funcionan los sistemas de software a gran escala (Hackett et al., 2023). La habilidad de colaborar en proyectos de equipo, manejar diferentes versiones del código y depurar errores complejos son competencias que los estudiantes deben adquirir para ser programadores competentes (Buckner et al.,



2023). Por lo tanto, los entornos de desarrollo no solo facilitan el aprendizaje inicial, sino que también permiten a los estudiantes avanzar hacia niveles más complejos y profesionalmente relevantes, integrando herramientas que se utilizan en la industria y preparando a los futuros programadores para una carrera exitosa.

Los estudios previos sobre el impacto de PyCharm en el aprendizaje de estudiantes universitarios han demostrado que esta herramienta ofrece beneficios significativos en la enseñanza de la programación, especialmente en cursos centrados en Python. Investigaciones realizadas en diversas universidades han mostrado que PyCharm, al ser un entorno de desarrollo especializado en Python, permite a los estudiantes concentrarse más en el contenido y la lógica de la programación en lugar de en los aspectos técnicos de la configuración del entorno. Al ofrecer autocompletado avanzado de código, sugerencias contextuales y verificación en tiempo real, PyCharm reduce el número de errores sintácticos y ayuda a los estudiantes a centrarse en el desarrollo de soluciones a problemas complejos. Estas características, documentadas en varios estudios, contribuyen a una mejora general del rendimiento académico, especialmente en estudiantes que tienen poca experiencia previa en programación (García & Revano, 2021a).

Otro aspecto importante que han revelado los estudios sobre PyCharm es su capacidad para fomentar el autoaprendizaje y la independencia del estudiante. Al ofrecer un entorno que proporciona retroalimentación inmediata y visualización clara de errores, los estudiantes pueden corregir sus propios problemas sin depender tanto del docente, lo que potencia el desarrollo de habilidades autodidactas (Xu et al., 2021). Investigaciones también han sugerido que los estudiantes que utilizan PyCharm desarrollan una mejor comprensión de la estructura del código y de las mejores prácticas en programación, en comparación con aquellos que utilizan editores de texto simples o menos integrados. Esto se debe a que PyCharm no solo permite escribir y ejecutar código de forma eficiente, sino que también expone a los estudiantes a flujos de trabajo profesionales desde una etapa temprana, como el uso de control de versiones, testing y la gestión de proyectos más complejos. Estos estudios concluyen que el uso de herramientas como PyCharm no solo mejora el rendimiento a corto plazo, sino que también prepara mejor a los estudiantes para desafíos técnicos más avanzados (Smirnov et al., 2021).



Las percepciones de los estudiantes sobre PyCharm como herramienta educativa son en su mayoría positivas, especialmente entre aquellos que buscan desarrollar habilidades avanzadas en programación. Los estudiantes suelen valorar la interfaz intuitiva y las herramientas integradas que PyCharm ofrece, como el autocompletado de código y la verificación en tiempo real de errores (Zhou et al., 2021). Estas características permiten a los principiantes cometer menos errores durante la escritura del código y enfocarse en la lógica del problema en lugar de en los detalles sintácticos. Además, muchos estudiantes consideran que PyCharm, al ser un entorno profesional utilizado ampliamente en la industria, les da una ventaja al familiarizarse con herramientas que encontrarán en el mundo laboral. Aunque algunos estudiantes al principio pueden sentirse abrumados por la cantidad de funciones disponibles, la mayoría coincide en que, una vez superada la curva de aprendizaje inicial, PyCharm se convierte en una herramienta poderosa y eficiente que mejora su capacidad para resolver problemas de programación (Dirks-Naylor, 2023).

Desde la perspectiva de los docentes, PyCharm también es percibido como una herramienta educativa efectiva que facilita tanto la enseñanza como la evaluación del progreso de los estudiantes. Los docentes aprecian la capacidad de PyCharm para ofrecer retroalimentación inmediata, lo que permite a los estudiantes aprender de manera autónoma y resolver problemas por sí mismos sin depender constantemente de la intervención del instructor (Gu et al., 2022). Sin embargo, algunos profesores han señalado que, debido a la complejidad de la herramienta, es necesario dedicar tiempo adicional para capacitar a los estudiantes en su uso efectivo. A pesar de esto, la mayoría de los docentes concuerda en que PyCharm no solo mejora el proceso de enseñanza, sino que también prepara a los estudiantes de manera más completa para los desafíos técnicos que enfrentarán en el ámbito profesional.

Los beneficios documentados del uso de PyCharm en términos de rendimiento académico han sido ampliamente destacados en estudios recientes, especialmente en el contexto de la enseñanza de la programación en Python. Uno de los principales beneficios que se ha observado es la reducción significativa de errores sintácticos y lógicos en los estudiantes que utilizan PyCharm en comparación con aquellos que usan editores de texto más simples. PyCharm proporciona una retroalimentación inmediata mediante su sistema de autocompletado y la detección en tiempo real de errores, lo que permite a los estudiantes corregir fallos rápidamente (Zhou et al., 2021). Esta característica no solo



mejora la precisión del código, sino que también fomenta la confianza en los estudiantes al ayudarlos a identificar y solucionar problemas de manera autónoma (Kurniawan et al., 2023). Los estudios han mostrado que los estudiantes que utilizan PyCharm tienden a cometer menos errores en sus exámenes y proyectos finales, lo que resulta en una mejora general en su rendimiento académico.

Otro beneficio documentado es el aumento en la comprensión de conceptos avanzados de programación debido al uso de herramientas integradas como depuradores, pruebas automatizadas y entornos virtuales que PyCharm ofrece. Estas herramientas permiten a los estudiantes interactuar de manera más profunda con su código y entender mejor cómo funcionan los programas en diferentes niveles. Por ejemplo, el uso del depurador integrado facilita la identificación de errores lógicos que no son evidentes a simple vista, lo que incrementa la comprensión de estructuras complejas como bucles, condicionales y recursividad. Además, PyCharm expone a los estudiantes a flujos de trabajo profesionales, como el uso de control de versiones con Git, lo que mejora su preparación para el mundo laboral (Guerrero-Higueras et al., 2020).

A pesar de los numerosos estudios sobre el impacto de PyCharm en el aprendizaje de la programación, existe una notable falta de investigación específica que se centre en los estudiantes de primer y segundo semestre de carreras relacionadas con Tecnologías de la Información. La mayoría de las investigaciones actuales se han enfocado en estudiantes avanzados o en contextos más amplios, lo que ha dejado un vacío en la comprensión de cómo PyCharm afecta a aquellos que están en las primeras etapas de su formación académica. Estos primeros semestres son cruciales porque es cuando los estudiantes desarrollan las bases de la lógica de programación y comienzan a familiarizarse con el entorno técnico (Salinas et al., 2023).

La falta de estudios enfocados en estos estudiantes iniciales también impide entender cómo el uso de PyCharm influye en su rendimiento académico y en la adquisición de competencias básicas de programación. Si bien se han reconocido los beneficios generales de PyCharm en términos de autocompletado, depuración y organización de proyectos, no se ha examinado detalladamente si estas características son percibidas como útiles o abrumadoras por los estudiantes de primer nivel. Es necesario llevar a cabo investigaciones que exploren cómo PyCharm impacta en la motivación de estos estudiantes, en su capacidad para resolver problemas simples, y en su percepción de la programación



como una disciplina accesible (Thorat & Kshirsagar, 2021). Este tipo de estudios no solo contribuiría a mejorar la enseñanza de la programación, sino que también podría proporcionar a los docentes información valiosa para ajustar sus metodologías y apoyar mejor a los estudiantes en los primeros semestres, quienes a menudo necesitan más orientación y apoyo en la fase inicial de su aprendizaje (Hulls & Rennick, 2020).

La literatura académica sobre la percepción de los estudiantes que utilizan PyCharm en comparación con otros entornos de desarrollo integrados (IDEs) es sorprendentemente escasa, a pesar de la creciente popularidad de estas herramientas en la enseñanza de la programación. Existen numerosos estudios que documentan el uso de IDEs en general, pero muy pocos profundizan en cómo los estudiantes perciben las diferencias entre PyCharm y otras plataformas como Visual Studio Code, Eclipse o Sublime Text. La falta de investigaciones comparativas deja varias preguntas sin respuesta, como cuál es el IDE más efectivo para mejorar el aprendizaje de conceptos de programación, o si la complejidad de herramientas como PyCharm puede generar una curva de aprendizaje demasiado pronunciada para los estudiantes novatos. Aún no se ha explorado de manera suficiente si la especialización de PyCharm en Python lo convierte en una herramienta preferida para los estudiantes o si su enfoque avanzado puede ser contraproducente para quienes recién comienzan (Q. Nguyen, 2019).

La escasa investigación comparativa también afecta la comprensión de cómo los estudiantes valoran las funcionalidades específicas de PyCharm frente a las de otros IDEs. Mientras que herramientas como Visual Studio Code destacan por su flexibilidad y extensibilidad, y Sublime Text por su ligereza y velocidad, PyCharm ofrece una experiencia más robusta y orientada a proyectos grandes y complejos. Sin embargo, no se ha investigado en profundidad si los estudiantes, especialmente en niveles introductorios, encuentran estas funcionalidades avanzadas de PyCharm útiles o si prefieren entornos más simplificados. Tampoco hay suficiente literatura que explore cómo los estudiantes perciben la facilidad de uso, la capacidad de personalización o la integración de herramientas adicionales en estos diferentes IDEs (Xu et al., 2021). Este vacío en la literatura genera una necesidad urgente de estudios que comparen la experiencia del usuario y el rendimiento académico de los estudiantes que utilizan diferentes IDEs, lo que permitiría a los docentes tomar decisiones más informadas al seleccionar herramientas de enseñanza para sus cursos.



A pesar de la creciente adopción de PyCharm en entornos educativos, existe un notable desconocimiento sobre la relación directa entre su uso y la comprensión de los conceptos fundamentales en Tecnologías de la Información y Comunicación (TIC). Si bien es ampliamente reconocido que PyCharm facilita el proceso de programación mediante funciones como el autocompletado y la depuración en tiempo real, no se ha investigado lo suficiente cómo estas características impactan directamente en la asimilación de conceptos clave, como la lógica de programación, las estructuras de control o la gestión de datos. Muchos estudios tienden a enfocarse en la efectividad del IDE en términos de productividad y eficiencia en la escritura de código, pero pocos han examinado cómo estas herramientas ayudan, o en algunos casos, obstaculizan el entendimiento profundo de los fundamentos de la programación. Este vacío en la investigación impide comprender si los estudiantes realmente logran interiorizar los principios subyacentes a la programación, o si dependen demasiado de las funciones automatizadas que ofrece el entorno (Xu et al., 2021).

Además, no se ha explorado suficientemente si el uso intensivo de PyCharm podría llevar a una excesiva dependencia de sus características en lugar de fomentar una comprensión más autónoma de los conceptos fundamentales. Sin embargo, no se ha documentado si PyCharm contribuye a fortalecer esta base o si, por el contrario, podría estar desviando la atención de los estudiantes hacia la resolución rápida de problemas sin una comprensión sólida (Xu et al., 2021). Esto plantea la necesidad urgente de estudios empíricos que evalúen cómo PyCharm, y sus características avanzadas, influyen en la adquisición de habilidades de programación a nivel conceptual, para determinar si esta herramienta es ideal para los niveles introductorios o si se debe utilizar con más cautela en etapas iniciales.

METODOLOGÍA

Este estudio se llevó a cabo con un enfoque cuantitativo, utilizando un diseño descriptivo y comparativo, con el objetivo de analizar la percepción del uso de la herramienta PyCharm entre estudiantes de primer y segundo semestre de la carrera de Tecnologías de la Información en la Universidad Estatal Amazónica. Se buscó evaluar la familiaridad, facilidad de uso, y el impacto de PyCharm en la comprensión de los conceptos de programación, así como su influencia en el rendimiento académico de los estudiantes.



El diseño de la investigación es no experimental y transversal, ya que se recolectaron datos en un único punto en el tiempo sin manipular las variables. Este enfoque permitió obtener un panorama amplio sobre cómo los estudiantes perciben PyCharm en diferentes etapas de su formación académica. La comparación entre estudiantes de primer y segundo semestre facilita identificar diferencias basadas en la experiencia con la herramienta.

La muestra estuvo compuesta por 246 estudiantes matriculados en los primeros dos semestres de la carrera de Tecnologías de la Información. De estos, 96 estudiantes pertenecían al primer semestre (Nivel I) y 150 al segundo semestre (Nivel II). La selección de los participantes se realizó a través de un muestreo no probabilístico por conveniencia, ya que se trabajó con los estudiantes disponibles durante el periodo de estudio.

Se recogieron datos demográficos básicos de los estudiantes, como el género y el nivel académico. La muestra estuvo compuesta por 102 estudiantes femeninos y 144 estudiantes masculinos. Estos datos se incluyeron para explorar posibles diferencias en la percepción de PyCharm según el género y el nivel académico.

Para la recolección de datos, se utilizó una encuesta estructurada diseñada específicamente para este estudio. La encuesta constaba de 11 ítems organizados en tres secciones:

Competencias Técnicas y Familiaridad: Evaluaba el grado de familiaridad de los estudiantes con PyCharm, así como sus competencias técnicas básicas en el uso de esta herramienta.

Facilidad de Uso y Funcionalidades: Exploraba la percepción de los estudiantes sobre la facilidad de uso de PyCharm, la intuitividad de su interfaz, y la utilidad de funcionalidades específicas como el autocompletado de código.

Impacto en la Comprensión de la Programación: Medía la percepción de los estudiantes sobre el impacto de PyCharm en su comprensión de los conceptos fundamentales de programación y en su rendimiento académico.

Para asegurar la fiabilidad del instrumento de medición, se calculó el coeficiente Alfa de Cronbach, obteniendo un valor de 0.966. Este valor indica una excelente consistencia interna entre los ítems de la encuesta, lo que confirma que el instrumento utilizado es adecuado para medir la percepción de los estudiantes sobre PyCharm.



La recolección de datos se llevó a cabo en el transcurso de una semana, durante las clases de programación en las que se utiliza PyCharm como entorno de desarrollo principal. Los estudiantes completaron la encuesta de manera anónima y voluntaria, en formato físico. Se explicó a los participantes el propósito del estudio y se garantizó la confidencialidad de sus respuestas. No hubo incentivos para la participación, con el fin de evitar sesgos en las respuestas.

Los datos recolectados se analizaron utilizando el software estadístico SPSS, versión 25. Se realizaron análisis descriptivos para obtener medidas de tendencia central y dispersión (media, desviación estándar) para cada ítem y para las diferentes subescalas de la encuesta.

Prueba de fiabilidad: Se utilizó el Alfa de Cronbach para evaluar la consistencia interna de los ítems de la encuesta.

Análisis descriptivo: Se calcularon las medias y desviaciones estándar de las respuestas, tanto para el total de la muestra como por subgrupos (Nivel I, Nivel II, género).

Prueba t de muestras independientes: Para comparar las percepciones entre los estudiantes de Nivel I y Nivel II, se realizó una prueba t de muestras independientes, que permitió identificar si existían diferencias estadísticamente significativas entre ambos grupos. Antes de la prueba t, se realizó la prueba de Levene para evaluar la igualdad de varianzas entre los grupos.

Prueba t para comparación de género: Para evaluar si existían diferencias significativas en la percepción de PyCharm entre hombres y mujeres, se realizó una prueba t de muestras independientes similar.

Análisis ANOVA: Se llevó a cabo un análisis de varianza (ANOVA) para explorar posibles diferencias en la percepción de PyCharm entre las diferentes regiones (Amazónica, Sierra y Costa).

RESULTADOS Y DISCUSIÓN

El presente estudio evaluó la percepción de los estudiantes de primer y segundo semestre de la carrera de Tecnologías de la Información sobre el uso de PyCharm, con un enfoque en su familiaridad con la herramienta, su facilidad de uso y el impacto en la comprensión de los conceptos de programación. Se llevaron a cabo análisis estadísticos para comparar las percepciones entre los diferentes niveles académicos y géneros, utilizando pruebas t de muestras independientes y ANOVA. A continuación, se presentan los hallazgos más relevantes, que incluyen diferencias significativas en las percepciones de



los estudiantes según su nivel académico, así como un análisis detallado de la fiabilidad de los instrumentos de medición empleados en el estudio. (Tabla 1).

Tabla 1 Estadísticas de fiabilidad

Alfa de Cronbach	Alfa de Cronbach basada en elementos estandarizados	N de elementos
0,966	0,967	11

Nota: elaborado por los autores

El instrumento utilizado en el estudio muestra un Alfa de Cronbach de 0.966, lo que indica una excelente consistencia interna de los ítems. Este valor sugiere que las preguntas incluidas en el cuestionario miden de manera coherente la percepción sobre el uso de PyCharm. El Alfa de Cronbach basado en los elementos estandarizados es de 0.967, lo cual refuerza la fiabilidad del instrumento. Con un total de 11 ítems, estos resultados permiten asegurar que las respuestas proporcionadas por los estudiantes tienen un alto grado de confiabilidad, lo que avala la validez de los análisis subsecuentes.

Tabla 2 | Estadísticas de elemento de resumen

	Media	Mínimo	Máximo	Rango	Máximo / Mínimo	Varianza	N de elementos
Medias de elemento	3,721	3,585	3,846	,260	1,073	,009	11
Varianzas de elemento	1,216	1,013	1,378	,365	1,361	,012	11

Nota: elaborado por los autores

La (Tabla 2) proporciona la información sobre las características descriptivas de los ítems incluidos en el cuestionario:

- La media de los ítems es de 3.721, lo que sugiere una tendencia positiva en la percepción general de los estudiantes hacia el uso de PyCharm.
- El mínimo valor registrado en las medias de los ítems es 3.585, y el máximo es 3.846, con un rango de 0.260, lo que indica una variación relativamente baja entre los ítems.
- El cociente entre el valor máximo y mínimo es de 1.073, lo que refleja una proporción equilibrada en las respuestas de los estudiantes.
- La varianza de los ítems es de 1.216 en promedio, con un mínimo de 1.013 y un máximo de 1.378, lo que indica diferencias moderadas en la dispersión de las respuestas entre los ítems.



- El cuestionario constaba de 11 ítems, y los resultados sugieren que la dispersión en las respuestas no fue excesiva, asegurando una medición estable de las percepciones.

Tabla 3 | Prueba de Muestras Independientes.

		Prueba de Levene de calidad de varianzas		prueba t para la igualdad de medias						
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar	95% de intervalo de confianza de la diferencia	
									Inferior	Superior
Promedio	Se asumen varianzas iguales	8,599	,004	-4,709	244	,000	-.56356	,11968	-.79930	-.32782
	No se asumen varianzas iguales			-4,460	166,898	,000	-.56356	,12635	-.81301	-.31410

Nota: elaborado por los autores

La prueba de muestras independientes (Tabla 3) se realizó para comparar la percepción sobre PyCharm entre los estudiantes de primer y segundo semestre (Nivel I y Nivel II). Los resultados clave son los siguientes:

Prueba de Levene para la igualdad de varianzas: El valor de $F = 8.599$ con una significancia (p) de 0.004 indica que no se pueden asumir varianzas iguales entre los grupos, ya que la significancia es menor a 0.05 . Esto sugiere que la dispersión de las respuestas difiere significativamente entre los estudiantes de Nivel I y Nivel II.

Prueba t (asumiendo varianzas iguales):

El valor de $t = -4.709$ con $gl = 244$ y una significancia (p) bilateral de 0.000 indica que hay una diferencia estadísticamente significativa entre las medias de los dos niveles.

La diferencia de medias es de -0.56356 , lo que significa que los estudiantes de Nivel II tienen una percepción más alta de PyCharm en comparación con los de Nivel I.

El intervalo de confianza del 95% para la diferencia de medias va desde -0.79930 hasta -0.32782 , confirmando que esta diferencia es significativa.

Prueba t (sin asumir varianzas iguales):

Aunque las varianzas no son iguales, la prueba t sigue siendo significativa ($t = -4.460$, $gl = 166.898$, $p = 0.000$), lo que refuerza la conclusión de que existe una diferencia significativa en la percepción entre los niveles.

La diferencia de error estándar es ligeramente mayor cuando no se asumen varianzas iguales (0.12635), pero la diferencia de medias sigue siendo la misma (-0.56356), con un intervalo de confianza de -0.81301 a -0.31410.

En resumen, los resultados indican que los estudiantes de Nivel II tienen una percepción significativamente más positiva de PyCharm en comparación con los estudiantes de Nivel I.

Tabla 4 | Prueba de Muestras Independientes.

		Prueba de Levene de calidad de varianzas		prueba t para la igualdad de medias						
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar	95% de intervalo de confianza de la diferencia	
									Inferior	Superior
Promedio	Se asumen varianzas iguales	,001	,974	-1,165	244	,245	-,14382	,12343	-,38693	,09930
	No se asumen varianzas iguales			-1,160	214,264	,247	-,14382	,12396	-,38814	,10051

Nota: elaborado por los autores

La prueba de muestras independientes (Tabla 4), también se aplicó para comparar las percepciones entre hombres y mujeres respecto al uso de PyCharm. Los resultados fueron los siguientes:

Prueba de Levene para la igualdad de varianzas: El valor de $F = 0.001$ y la significancia (p) = 0.974 indican que no hay una diferencia significativa en las varianzas entre los grupos de hombres y mujeres, por lo que se pueden asumir varianzas iguales entre los dos grupos.

Prueba t (asumiendo varianzas iguales):

El valor de $t = -1.165$ con $gl = 244$ y una significancia (p) bilateral = 0.245 muestra que no hay una diferencia estadísticamente significativa entre las percepciones de hombres y mujeres, ya que el valor de p es mayor a 0.05.

La diferencia de medias es de -0.14382, lo que sugiere que las mujeres tienen una percepción ligeramente inferior de PyCharm en comparación con los hombres, pero esta diferencia no es significativa.

El intervalo de confianza del 95% para la diferencia de medias oscila entre -0.38693 y 0.09930, lo que incluye el valor cero, reforzando la falta de significancia.

Prueba t (sin asumir varianzas iguales):



Cuando no se asumen varianzas iguales, los resultados son similares: $t = -1.160$, $gl = 214.264$, $p = 0.247$, con una diferencia de medias de -0.14382 .

En conclusión, los resultados indican que no existen diferencias estadísticamente significativas en la percepción de PyCharm entre hombres y mujeres, ya que ambas pruebas t no revelan diferencias significativas entre los grupos.

Tabla 5 | Prueba Anova

	Suma de cuadrados	Gl	Media cuadrática	F	Sig.
Entre grupos	5,230	2	2,615	2,916	,056
Dentro de grupos	217,943	243	,897		
Total	223,173	245			

Nota: elaborado por los autores

El análisis ANOVA (Tabla 5) se utilizó para examinar si existen diferencias significativas en la percepción de PyCharm entre las diferentes regiones (Amazonia, Sierra y Costa).

Estos resultados sugieren que, aunque existe cierta variabilidad en las percepciones entre las regiones, no es lo suficientemente significativa como para rechazar la hipótesis nula al nivel de significancia convencional de 0.05. El valor de $p = 0.056$ indica que la diferencia entre los grupos está cerca del umbral de significancia, pero no lo alcanza. En resumen, el ANOVA muestra que no hay diferencias estadísticamente significativas en la percepción de PyCharm entre los grupos evaluados

El hecho de que los estudiantes de segundo semestre tengan una percepción significativamente más positiva sobre PyCharm en comparación con los de primer semestre coincide con estudios que sugieren que el aumento en la exposición y práctica mejora la competencia técnica y la familiaridad con los entornos de desarrollo integrados (IDEs) (Aires et al., 2023b)(Garcia & Revano, 2021b). Los IDEs como PyCharm, que ofrecen herramientas como el autocompletado de código y la depuración en tiempo real, reducen la carga cognitiva de los estudiantes, permitiéndoles concentrarse más en los conceptos de programación que en los detalles técnicos (Buckner et al., 2023).

El hecho de que no se encontraran diferencias significativas en la percepción de PyCharm entre hombres y mujeres refuerza la neutralidad de esta herramienta tecnológica, como se ha documentado en estudios similares que evalúan el uso de IDEs en educación (Rahman & Morgan, 2021). Esto sugiere que, cuando



se trata de la programación, el acceso a herramientas avanzadas como PyCharm puede generar una experiencia de aprendizaje equitativa para todos los estudiantes, independientemente de su género (Rahman & Morgan, 2021).

El análisis ANOVA no reveló diferencias estadísticamente significativas entre los estudiantes de diferentes regiones. Este hallazgo coincide con investigaciones que señalan que el uso de tecnologías educativas como PyCharm tiende a homogeneizar la experiencia de aprendizaje, ofreciendo las mismas oportunidades de acceso y desarrollo de habilidades independientemente del contexto geográfico (Rahman & Morgan, 2021).

El alto coeficiente de fiabilidad del Alfa de Cronbach (0.966) refuerza la validez del instrumento utilizado en este estudio, lo que coincide con otras investigaciones que han empleado encuestas similares para evaluar percepciones sobre herramientas tecnológicas en la educación (Aires et al., 2023). Esto garantiza que los resultados obtenidos son consistentes y reflejan adecuadamente las percepciones de los estudiantes respecto al uso de PyCharm (García & Revano, 2021b).

CONCLUSIONES

El presente estudio ha revelado importantes hallazgos sobre la percepción del uso de PyCharm entre los estudiantes de primer y segundo semestre de la carrera de Tecnologías de la Información. A continuación, se destacan las principales conclusiones:

Tanto los estudiantes de primer como de segundo semestre perciben de manera favorable el uso de PyCharm en su proceso de aprendizaje. Las características como el autocompletado de código, la facilidad de depuración y la interfaz intuitiva fueron aspectos valorados positivamente, especialmente entre los estudiantes con más experiencia.

Se identificaron diferencias significativas entre los estudiantes de primer y segundo semestre, con los estudiantes del segundo semestre reportando una mayor familiaridad y competencia técnica en el uso de PyCharm. Este hallazgo es consistente con investigaciones previas que indican que la exposición prolongada a herramientas de desarrollo mejora la percepción de su utilidad y facilidad de uso.

Ausencia de diferencias de percepción por género: No se encontraron diferencias significativas en la percepción de PyCharm entre hombres y mujeres, lo que sugiere que esta herramienta es percibida de



manera similar entre ambos géneros. Esto refuerza la idea de que PyCharm, al ser una herramienta altamente accesible, facilita un entorno de aprendizaje equitativo.

REFERENCIAS BIBLIOGRAFICAS

- Aires, J. P., Aires, S. B. K., Pereira, M. J. V., & Alves, L. M. (2023a). Using the Methodology Problem-Based Learning to Teaching Programming to Freshman Students. *International Journal of Information and Education Technology*. <https://api.semanticscholar.org/CorpusID:257443671>
- Aires, J. P., Aires, S. B. K., Pereira, M. J. V., & Alves, L. M. (2023b). Using the Methodology Problem-Based Learning to Teaching Programming to Freshman Students. *International Journal of Information and Education Technology*. <https://api.semanticscholar.org/CorpusID:257443671>
- Allison, M., Turner, S., & Kwasny, M. (2021). Assessing Cognitive Load for Junior Software Engineers: A Mixed-Method Study. *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, 883–888.
<https://api.semanticscholar.org/CorpusID:249929699>
- Buckner, J., Liu, X., Chakravorty, A., Wu, Y., Cervantes, L. F., Lai, T. T., & Brooks, C. L. I. I. I. (2023). pyCHARMM: Embedding CHARMM Functionality in a Python Framework. *Journal of Chemical Theory and Computation*, *19*(12), 3752–3762.
<https://doi.org/10.1021/acs.jctc.3c00364>
- Cheng, G. (2021). The Effects of Self-regulated Learning Strategies on Computer Programming Achievement in Teacher Education. *Frontiers in Education Technology*.
<https://api.semanticscholar.org/CorpusID:244553839>
- Dirks-Naylor, A. J. (2023). Perceived value and use of assessment software exam summary reports by student pharmacists. *Physiology*. <https://api.semanticscholar.org/CorpusID:258981598>
- Elvina, E., Karnalim, O., Ayub, M., & Wijanto, M. C. (2018). Combining program visualization with programming workspace to assist students for completing programming laboratory task. *Journal of Technology and Science Education*.
<https://api.semanticscholar.org/CorpusID:56394504>



Garcia, M. B., & Revano, T. F. (2021a). Assessing the Role of Python Programming Gamified Course on Students' Knowledge, Skills Performance, Attitude, and Self-Efficacy. *2021 IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, 1–5.

<https://api.semanticscholar.org/CorpusID:244919688>

Garcia, M. B., & Revano, T. F. (2021b). Assessing the Role of Python Programming Gamified Course on Students' Knowledge, Skills Performance, Attitude, and Self-Efficacy. *2021 IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, 1–5.

<https://api.semanticscholar.org/CorpusID:244919688>

Guerrero-Higueras, Á. M., Llamas, C. F., González, L. S., Fernández, A. G., Costales, G. E., & Gonzalez, M. A. C. (2020). Academic Success Assessment through Version Control Systems. *Applied Sciences*. <https://api.semanticscholar.org/CorpusID:214121518>

Hackett, P., Wermelinger, M., Kear, K., & Douce, C. (2023). Using a Virtual Computing Lab to Teach Programming at a Distance. *Proceedings of the 7th Conference on Computing Education Practice*. <https://api.semanticscholar.org/CorpusID:254975541>

Hliš, T., Četina, L., Beranič, T., & Pavlič, L. (2023). Evaluating the Usability and Functionality of Intelligent Source Code Completion Assistants: A Comprehensive Review. *Applied Sciences (Switzerland)*, 13(24). <https://doi.org/10.3390/app132413061>

Hu, M., Assadi, T., & Mahrooian, H. (2021). Teaching Visualization-first for Novices to Understand Programming. *2021 IEEE International Conference on Engineering, Technology & Education (TALE)*, 654–660. <https://api.semanticscholar.org/CorpusID:246291765>

Hulls, C. C. W., & Rennick, C. J. (2020). Use of a Cornerstone Project to Teach Ill-Structured Software Design in First Year. *IEEE Transactions on Education*, 63, 98–107.

<https://api.semanticscholar.org/CorpusID:214256561>

Khatser, G., & Khatser, M. (2022). Online Learning Through LMSs: Comparative Assessment of Canvas and Moodle. *Int. J. Emerg. Technol. Learn.*, 17, 184–200.

<https://api.semanticscholar.org/CorpusID:249938054>



- Ling, H.-C., Hsiao, K.-L., & Hsu, W.-C. (2021). Can Students' Computer Programming Learning Motivation and Effectiveness Be Enhanced by Learning Python Language? A Multi-Group Analysis. *Frontiers in Psychology, 11*. <https://api.semanticscholar.org/CorpusID:231654367>
- Lukkarinen, A., Malmi, L., & Haaranen, L. (2021). Event-driven Programming in Programming Education. *ACM Transactions on Computing Education (TOCE), 21*, 1–31. <https://api.semanticscholar.org/CorpusID:237250643>
- Macedo, F., Yamin, J. L., Silva, W. J., de Holanda, M. T., & Oliveira, R. B. (2022). AsPIn: A Moodle Plugin for Introductory Programming Teaching. *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*, 1–6. <https://api.semanticscholar.org/CorpusID:250567544>
- Nguyen, B.-A., Ha, T.-V. T., & Chen, H.-M. (2023). ANALYZING GIT LOG IN AN CODE-QUALITY AWARE AUTOMATED PROGRAMMING ASSESSMENT SYSTEM: A CASE STUDY. *TRA VINH UNIVERSITY JOURNAL OF SCIENCE; ISSN: 2815-6072; E-ISSN: 2815-6099*. <https://api.semanticscholar.org/CorpusID:265335617>
- Nguyen, Q. (2019). *Hands-on application development with PyCharm: Accelerate your Python applications using practical coding techniques in PyCharm*. Packt Publishing Ltd.
- Nikandrov, A. A. (2023). Multifunctional and flexible online platforms for creating educational materials. *Informatics and Education*. <https://api.semanticscholar.org/CorpusID:256256581>
- Noor, N. M., Saad, A., Ibrahim, A. B., & Noor, N. M. (2023a). THE ACCEPTANCE OF AN EDUCATIONAL INTEGRATED DEVELOPMENT ENVIRONMENT TO LEARN PROGRAMMING FUNDAMENTALS. *Information Technologies and Learning Tools*. <https://api.semanticscholar.org/CorpusID:257274115>
- Noor, N. M., Saad, A., Ibrahim, A. B., & Noor, N. M. (2023b). THE ACCEPTANCE OF AN EDUCATIONAL INTEGRATED DEVELOPMENT ENVIRONMENT TO LEARN PROGRAMMING FUNDAMENTALS. *Information Technologies and Learning Tools*. <https://api.semanticscholar.org/CorpusID:257274115>
- Omeh, C. B., Olelewe, C. J., & Nwangwu, E. C. (2022). Impact of teaching computer programming using innovative pedagogy embedded with live online lectures and related tools: A randomized control trial. *Computer Applications in Engineering Education, 30*, 1390–1405.



<https://api.semanticscholar.org/CorpusID:248758905>

Protasevich, Yu. A., Zmeev, O. A., & Sokolov, D. A. (2021). Tools for organizing teachers-students interaction using version control systems. *Informatics and Education*.

<https://api.semanticscholar.org/CorpusID:243314307>

Skorohodov, D., & Kurochkina, M. A. (2021). On the Use of Artificial Intelligence for Student Distribution in Dormitories: MPhI University Campus Case. *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, 690–694.

<https://api.semanticscholar.org/CorpusID:233226075>

Smirnov, O., Lobanov, A., Golubev, Y., Tikhomirova, E., & Bryksin, T. (2021). Revizor: A Data-Driven Approach to Automate Frequent Code Changes Based on Graph Matching. *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 1242–1246.

<https://api.semanticscholar.org/CorpusID:237290097>

Tan, J., Chen, Y., & Jiao, S. (2023). Visual Studio Code in Introductory Computer Science Course: An Experience Report. *ArXiv, abs/2303.10174*.

<https://api.semanticscholar.org/CorpusID:257631579>

Thorat, S. A., & Kshirsagar, D. P. (2021). Developing Logic Building, Problem Solving, and Debugging Programming Skills Among Students. *Journal of Engineering Education Transformations*.

<https://api.semanticscholar.org/CorpusID:234093564>

Wang, P., & Zhao, Y. (2022). Research of online teaching management system based on artificial intelligence. *2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP)*, 1004–1008. <https://api.semanticscholar.org/CorpusID:249048989>

Xu, F. F., Vasilescu, B., & Neubig, G. (2021). In-IDE Code Generation from Natural Language: Promise and Challenges. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 31, 1–47. <https://api.semanticscholar.org/CorpusID:231718679>

Zhang, Z., Xing, Z., Xu, X., & Zhu, L. (2023). RIdiom: Automatically Refactoring Non-Idiomatic Python Code with Pythonic Idioms. *2023 IEEE/ACM 45th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 102–106.

<https://api.semanticscholar.org/CorpusID:259835438>



Zhou, W., Kim, S., Murali, V., & Aye, G. A. (2021). Improving Code Autocompletion with Transfer Learning. *2022 IEEE/ACM 44th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 161–162.

<https://api.semanticscholar.org/CorpusID:234482526>

