JETYPEKA
Revista Científica Multidisciplinaria
ISSN: 2617-7374. Julio-Diciembre 2026
Vol. 6 Nro.2. Pág: 19-35

ARTÍCULO ORIGINAL

ANÁLISIS COMPARATIVO: EFECTIVIDAD DE LA PROGRAMACIÓN EN PAREJA Y EN EQUIPO EN INGENIERÍA EN SISTEMAS

Isamar del Socorro Bonilla Martínez[1]

Recibido: 10/febrero/2026

Aprobado: 18/mayo/2026

Publicado:04/agosto/2026

[1]Centro Universitario Regional de Chontales "Cornelio Silva Argüello". Universidad Nacional Autónoma de Nicaragua, Managua. UNAN-Managua. Académica investigadora del Departamento de Ciencias Tecnológicas y Salud. Chontales-Nicaragua, https://orcid.org/0009-0002-0970-4605, isamar.bonilla@unan.edu.ni, ingsistunan@gmail.com

Análisis comparativo: efectividad de la programación en pareja y en equipo en ingeniería en sistemas.

Resumen:

La literatura consultada señala que el aprendizaje de la programación constituye un desafío central para los estudiantes de ciencias computacionales dado que requiere desarrollar competencias para dominar conceptos complejos, sintaxis de los lenguajes de programación y aplicar estos conocimientos para la resolución de problemas o corrección de errores complicados de programación que implica una elevada carga cognitiva en el proceso de enseñanza-aprendizaje de esta disciplina, se llevó a cabo este proceso investigativo con el objetivo: analizar la efectividad de los métodos de aprendizaje colaborativos en el desarrollo de competencias en programación de los estudiantes de Ingeniería en Sistemas de Información del CUR-Chontales, el estudio fue conducido bajo una metodología; con diseño no experimental, descriptivo y transversal que evalúa de manera integral la efectividad de los métodos colaborativos en pareja y en equipo en el aprendizaje de programación como competencia específica, en proyectos integradores, es una investigación de tipo aplicativo de campo. En términos comparativos, a partir del análisis de los resultados, puede evidenciarse que la programación en pareja mantiene una incidencia similar en las dificultades técnicas con la lógica, la sintaxis y el aprendizaje de nuevas tecnologías. En contraste, la programación en equipo induce una mayor complejidad en problemas de integración en la gestión del código compartido. Las dificultades y competencias más significativas de cada método fueron analizadas en los tres niveles: conceptual, procedimental y actitudinal. Los estudiantes perciben el método de aprendizaje colaborativo en grupo como más efectivo para potenciar la integración multidimensional y favorecer el desarrollo de las competencias en programación, aunque ambos métodos presentan dificultades y fortalezas específicas según el tipo de competencia. Este análisis permite a los docentes y coordinadores académicos tomar decisiones informadas sobre las estrategias pedagógicas más adecuadas que contribuyan al fortalecimiento de las habilidades técnicas y blandas en contextos académicos similares.

Palabras clave: analizar, efectividad, métodos colaborativos, aprendizaje, programación, competencias.

Los trabajos publicados en esta revista están bajo la licencia Creative Commons Atribución – No comercial

Ñehesa'ÿijo ñembojojáva: Mba'eapopyrâ oñondivegua ha atypegua ra'eve Mba'eapokuaahára papañemohendakuaaguiguápe

Ñemombyky:

Ñepyrûrâ. Mboyve guare: Kuatiañe'êita oñemoñe'êva'ekue ko tembiapo aporâ ahechauka pe mba'eapopyrâ jekuaave ha'eha peteî mba'e tekotevêtereíva ojejapova'erâ umi temimbo'ekuérape guarâ, umi oñemoarandúva Mba'eñemohendakuaa kuaapýpe (Ciencias Computacionales-pe), péa ko kuaapy oikotevêgui oguerojera katupyryrânguéra oikuaa porâ haguâ umi kuaapavê hekoetáva, umi ñe'êjoajungatu mba'eapopyrâ ñe'ênguéra ha ojepuru haguâ kuaaita oñemyesakäva apañuâi , térâ oñemoî porâ haguâ jejavyeta ijetu'úva mba'eapopyrâ rehegua, oikotevêva heta jekuaapy pe tembiapoguata tekombo'e ha jekuaaverâ ko mbo'erâme. Ko tembikuaareka ojejapokuri oñehesa'ÿijo haguâ ha'evépa umi jekuaaverâ ñopytyvôgua rapereko, ojeguerojerávo umi temimbo'ekuéra Mba'eapokuaahára papañemohendakuaaguiguápe Momarandugua CUR-CHONTALES-pegua (Ingeniería en Sistemas de Información del CUR-Chontales) katupyryrânguéra mba'eapopyrâme. Ko tembiapo oñesambyhýkuri peteî Taperekokuaaty orekóva haira'anga anduaporeko'ÿva, techapyrehaíva avei hasapáva, ohepyme'êmbaitéva ha ohechaukáva oikoiteha umi tapereko ojegueroguatáva oñondive ha atýpe pe mba'eapopyrâ jekuaave, ha'eramo guáicha peteî katupyryrâ iñemopeteîva, umi apopyrâ hyepyjapývape; péva ha'e peteî jetypeka jepururâ heseguápe. Tembiapokue: Ñe'ê ñembojojávape, ha ko ñehesa'ÿijo rire, oñemyesakâ pe mba'eapopyrâ oñondivegua ojojoguaha aporekópe umi pa' â oîvape kuaarape ndive, avei ñe'êjoajungatu ndive ha jekuaave mba'epyahupurukuaa ndive. Iñembohovaképe, mba'eapopyrâ oñondivegua ogueraha pa'â tuichavéva umi apanuâi hyepyjapývape, oñemohembiaporapévo pe ñe'êhai oñondivegua. Umi pa'â ha katupyryrâ imbaretevéva umi tapereko apytépe oñehesa'ÿijókuri mbohapy yvatekuépe: he'iséva rupi, ijejapo rupi ha imba'eapo rupive.

Ñe'ê tenondegua: jehesa'ÿijo, ha'eve, tapereko oñondivegua, jekuaave, mba'eapopyrâ, katuryrânguéra.

Los trabajos publicados en esta revista están bajo la licencia Creative Commons Atribución – No comercial

Comparative analysis: effectiveness of pair and team programming in systems engineering

Abstract:

The consulted literature indicates that learning programming constitutes a central challenge for computer science students, as it requires developing competencies to master complex concepts, programming language syntax, and applying this knowledge to solve problems or correct complicated programming errors, which entails a high cognitive load in the teaching-learning process. This research was conducted with the objective of analyzing the effectiveness of collaborative learning methods in the development of programming competencies among Information Systems Engineering students at CUR-Chontales. The study was carried out using a non-experimental, descriptive, and cross-sectional design, which comprehensively evaluates the effectiveness of pair and team-based collaborative methods in learning programming as a specific competency within integrative projects; it is an applied field study. In comparative terms, the analysis of the results shows that pair programming has a similar impact on technical difficulties related to logic, syntax, and learning new technologies, whereas team programming induces greater complexity in issues related to integration and shared code management. The most significant difficulties and competencies of each method were analyzed across three levels: conceptual, procedural, and attitudinal. Students perceive group-based collaborative learning as more effective for promoting multidimensional integration and fostering the development of programming competencies, although both methods present specific strengths and challenges depending on the type of competency. This analysis allows instructors and academic coordinators to make informed decisions about the most suitable pedagogical strategies to strengthen both technical and soft skills in similar academic contexts.

Keywords: analyze, effectiveness, collaborative methods, learning, programming, competencies

Los trabajos publicados en esta revista están bajo la licencia Creative Commons Atribución – No comercial

Introducción:

La literatura consultada, incluidas las publicaciones de la Association for Computing Machinery (ACM), la mayor asociación profesional del mundo en informática y ciencias de la computación, indica que la programación de computadoras es una de las habilidades más complejas y difíciles de desarrollar para los estudiantes de ingeniería en sistemas de información o computación. Esto se evidencia en investigaciones específicas: Gómez et al. (2008) señalan que "aprender a programar es complejo y requiere esfuerzo, perseverancia y un enfoque especial en lo que respecta a la forma de estudio y enseñanza." [traducción propia] (p. 2).

Según Herlambang et al. (2024), entre los principales problemas se destacan la elevada carga cognitiva implicada en el proceso de enseñanza-aprendizaje de este componente, la falta de competencias cognitivas propias para la resolución de errores y problemas por parte de los estudiantes y la dificultad de la sintaxis de los lenguajes de programación. Estos obstáculos son recurrentes en los cursos introductorios de programación, afectando tanto el desempeño como la retención de conceptos clave por parte de los estudiantes.

Asimismo, se encontró en la revisión de la literatura que los estudiantes presentan diversos desafíos. Según Qian & Lehman (2017), estas dificultades experimentadas por los estudiantes están relacionadas con muchos factores, incluyendo la falta de familiaridad con la sintaxis, el lenguaje natural, los conocimientos matemáticos, modelos mentales inexactos, falta de estrategias, entornos de programación y el conocimiento e instrucción de los docentes.

Estos hallazgos se complementan con lo que señala Insuasti (2016), citando a Soloway y Spohrer (1989), quienes sostienen que aprender a programar puede presentar grandes dificultades para los estudiantes:

La enseñanza de los fundamentos de programación es un punto clave en la formación de profesionales en ciencias computacionales. Pero existe evidencia que sugiere que aprender a programar no es una tarea fácil; así lo expresan Soloway y Spohrer en su libro. Estos autores afirman que la creación y el control de ambientes y soluciones computacionales a través de la programación son cosas que para un individuo pueden ser difíciles de realizar (Soloway & Spohrer, 1989, como se citó en Insuasti 2016, p. 235).

Los trabajos publicados en esta revista están bajo la licencia Creative Commons Atribución – No comercial

El aprendizaje de la programación representa un reto significativo para los estudiantes de Ingeniería en Sistemas de Información de la UNAN Managua-CUR Chontales. Aún no existe evidencia comparativa que determine cuál de estos métodos es más eficaz para el aprendizaje. La falta de información genera incertidumbre sobre la estrategia más adecuada, lo que afecta la posibilidad de fortalecer las competencias requeridas y de optimizar tanto el desempeño académico como la colaboración efectiva entre los estudiantes.

Para enfrentar esta dificultad, se han implementado en proyectos integradores ambos métodos colaborativos: la colaboración en pareja y el trabajo en equipo (con tres o más integrantes). Por ello, resulta necesario investigar la efectividad de ambos métodos en este contexto académico.

Este estudio comparativo de métodos colaborativos: programación en pareja vs. trabajo en equipo en el aprendizaje de programación en proyectos integradores en estudiantes de Ingeniería en Sistemas de Información tiene como objetivo: analizar la efectividad de los métodos de aprendizaje colaborativos en el desarrollo de competencias en programación de los estudiantes de Ingeniería en Sistemas de Información del CUR-Chontales.

El presente objetivo brinda las pautas para la siguiente interrogante: ¿Cuál es la efectividad de los métodos de aprendizaje colaborativos para el desarrollo de competencias en programación de los estudiantes?

La investigación es relevante porque el aprendizaje de la programación constituye un desafío central y una de las principales habilidades prácticas que los estudiantes de Ingeniería en Sistemas de Información deben desarrollar para resolver problemas computacionales. En el modelo educativo por competencias, se espera que cada participante del proceso de aprendizaje adquiera estas habilidades. Según el Documento Curricular de la Carrera de Ingeniería en Sistemas de Información, la primera competencia específica consiste en la “capacidad de desarrollar software de aplicación mediante el uso de nuevas tecnologías y las buenas prácticas para ofrecer soluciones integrales a la realidad socioeconómica del país” (Universidad Nacional Autónoma de Nicaragua, 2023, p. 19). En este sentido, el modelo curricular por competencias aflora logros en el trabajo integral de los profesores y muy especialmente en los estudiantes. (Aburto Jarquín, s.f.)

Los trabajos publicados en esta revista están bajo la licencia Creative Commons Atribución – No comercial

El análisis realizado permitirá identificar el método colaborativo más efectivo para el fortalecimiento de competencias técnicas como habilidades de colaboración. Asimismo, los resultados del estudio pueden servir como referencia para docentes y coordinadores académicos, aportando información valiosa que contribuye a mejorar la enseñanza de la programación y al desarrollo de competencias profesionales en proyectos integradores y contextos similares.

Metodología:

Se utilizó una metodología con un diseño que evalúa de manera integral la efectividad de los métodos colaborativos en pareja y en equipo en el aprendizaje de programación para el desarrollo de competencias en proyectos integradores de la carrera de Ingeniería en Sistemas de Información. En este apartado se detallan los procedimientos, herramientas y técnicas utilizados, incluyendo descripciones del contexto y de los participantes. Así como los instrumentos usados para la recolección de datos y el proceso seguido para su análisis.

El estudio es de tipo aplicativo de campo, con diseño no experimental, descriptivo y transversal. Se utilizó como instrumento para la recolección de datos un cuestionario con 40 preguntas cerradas, usando la escala Likert con opciones ordinales de 1-5, preguntas de opción múltiple con una sola respuesta y con múltiples respuestas. Este instrumento se organizó en seis dimensiones, que primero evalúan las dificultades conceptuales, procedimentales y actitudinales, así como las competencias conceptuales, procedimentales y actitudinales en cada uno de los métodos colaborativos, en correlación con una de las competencias específicas establecidas en el Documento Curricular de la carrera de Ingeniería en Sistemas de Información.

De igual forma, se realizó una prueba piloto utilizando la herramienta Google Forms, seleccionando aleatoriamente a 4 participantes del grupo objetivo para verificar la claridad de las preguntas y la comprensión de los ítems. Los datos de la prueba piloto no se incluyen en el análisis final. Los resultados de la prueba piloto permitieron realizar los ajustes necesarios para mejorar la comprensión del cuestionario optimizando la redacción de los ítems.

El tipo de muestreo empleado es no probabilístico por conveniencia, el cual “tiene su origen en consideraciones de tipo práctico, en las cuales se busca obtener la mejor información en el menor tiempo posible, de acuerdo con las circunstancias concretas que

Los trabajos publicados en esta revista están bajo la licencia Creative Commons Atribución – No comercial

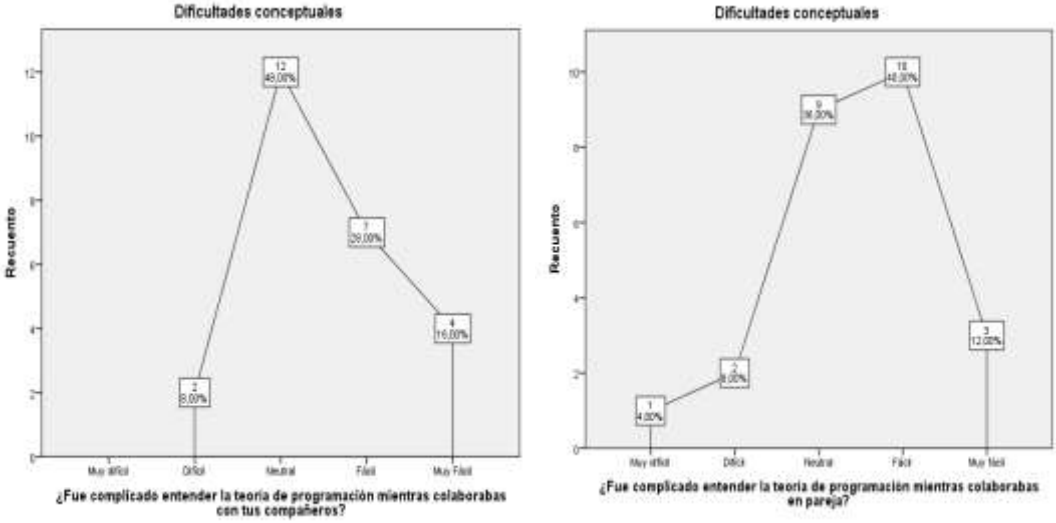
rodean tanto al investigador como a los sujetos o grupos investigados". (Quintana Peña, 2006, p. 13)

Una vez incorporados los ajustes que se identificaron en la prueba piloto, se procedió a compartir el cuestionario final por correo electrónico a 25 estudiantes que participaron en la investigación.

Resultados, discusión y análisis:

Ilustración 1

Comparativa de las dificultades conceptuales trabajo en equipo vs. trabajo en pareja



Fuente: Elaboración propia

A continuación, se presentan las dificultades conceptuales encontradas en los métodos colaborativos en equipo y en pareja.

Los gráficos detallan la distribución de las respuestas a la pregunta sobre la dificultad conceptual percibida al comprender la teoría de programación, comparando los métodos colaborativos en grupo y en pareja.

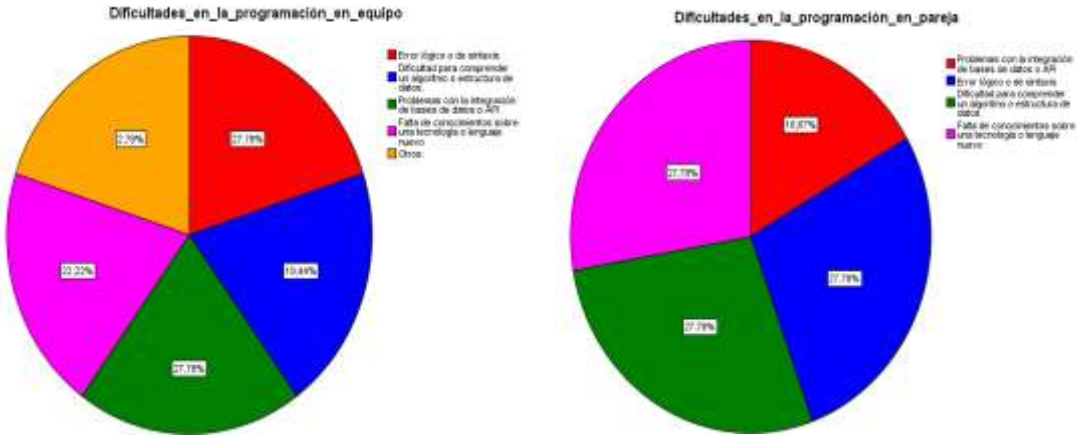
En la colaboración en grupo, más del 56% de los participantes considera que la teoría era difícil o muy difícil de entender, mientras que en el trabajo en pareja solo el 12% percibe la teoría como difícil o muy difícil de entender. De forma complementaria, la facilidad percibida en grupo es del 16% y en pareja del 52%. Por consiguiente, la colaboración en pareja reduce significativamente la percepción de dificultad, facilitando un aprendizaje más concentrado en contraste con el trabajo de grupo, donde se puede generar mayor confusión.

Los trabajos publicados en esta revista están bajo la licencia Creative Commons Atribución – No comercial

Seguidamente, se presenta el análisis de los gráficos sobre las principales dificultades procedimentales identificadas que enfrentan los estudiantes durante la programación en cada uno de los métodos colaborativos: en grupo y en pareja.

Ilustración 2

Comparativa de las dificultades procedimentales trabajo en equipo vs. trabajo en pareja



Fuente: Elaboración propia

En el trabajo en equipo se revelan diferencias importantes entre las principales dificultades encontradas: errores lógicos o de sintaxis y problemas con la integración de bases de datos o API, ambos con 27,78%. En segundo lugar, la falta de conocimientos sobre una tecnología o lenguaje con un 22.22%, asimismo, las dificultades para comprender un algoritmo o estructuras de datos con un 19.44% y por último otras dificultades con 2.18%.

Por el contrario, en la programación en pareja, los resultados muestran una distribución más equilibrada entre las dificultades, obteniendo tres respuestas con los mismos porcentajes: error lógico o de sintaxis, dificultad para comprender un algoritmo o estructuras de datos y falta de conocimientos sobre una tecnología o lenguaje nuevo, todas con un 27.78%, y en menor incidencia los problemas con la integración de bases de datos o API con un 16.67%.

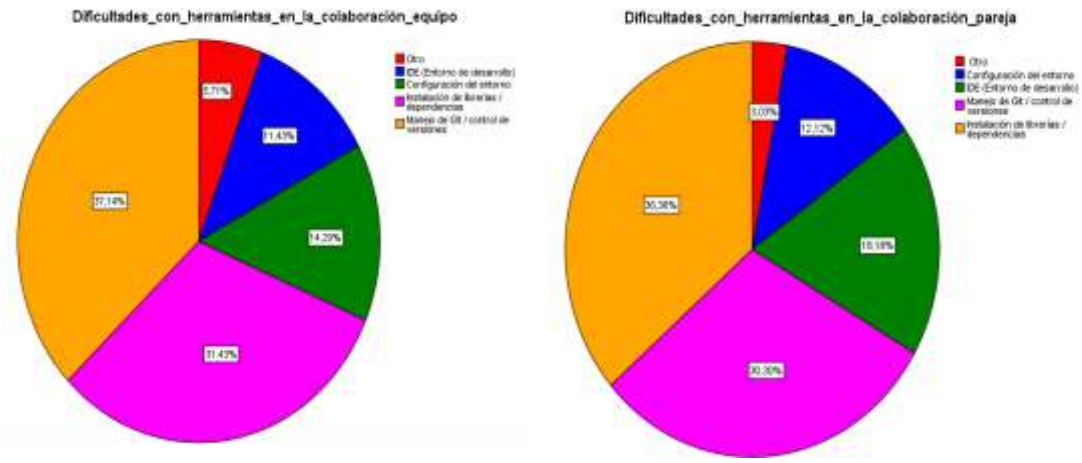
En términos comparativos, puede evidenciarse que la programación en pareja mantiene una incidencia similar en las dificultades técnicas con la lógica, la sintaxis y el aprendizaje de nuevas tecnologías. En contraste, la programación en equipo induce una mayor complejidad en problemas de integración en la gestión del código compartido, lo cual incrementa la probabilidad de errores vinculados a la sintaxis.

Los trabajos publicados en esta revista están bajo la licencia Creative Commons Atribución – No comercial

En la siguiente ilustración, se muestran las dificultades procedimentales al usar herramientas o entornos de programación en la colaboración en equipo y en pareja.

Ilustración 3

Comparativas dificultades procedimentales con herramientas del trabajo en equipo vs. en pareja



Fuente: Elaboración propia

En el caso de la colaboración en equipo, la principal dificultad identificada en relación con el uso de herramientas o entornos de programación es el manejo de Git o control de versiones con un 37.14%, en segundo lugar, la instalación de librerías o dependencias con un 31.43%, seguidamente la configuración del entorno con un 14.29% y los inconvenientes con el IDE con un 11.43%, y con menos frecuencia otros tipos de problemas con un 5.71%.

Por su parte, en la colaboración en pareja, la tendencia es similar, pero con algunas diferencias. La instalación de librerías o dependencias es la dificultad principal con un 36,36%; esto indica que los problemas técnicos relacionados con la configuración de los entornos de trabajo compartidos son los más comunes cuando se trabaja en pareja. Le siguen la dificultad de manejo de Git o control de versiones con un 30.30% y el entorno de desarrollo (IDE) con un 18.18%, con menos implicaciones, la configuración del entorno con un 12.12% y en menor escala otras dificultades con un 3.03%.

Por lo tanto, al comparar ambos casos, se puede observar que en la colaboración en pareja las dificultades están más relacionadas con la configuración inicial del entorno de trabajo específicamente la instalación de librerías o dependencias y en la colaboración en equipo las mayores dificultades se vinculan al uso de herramientas colaborativas como

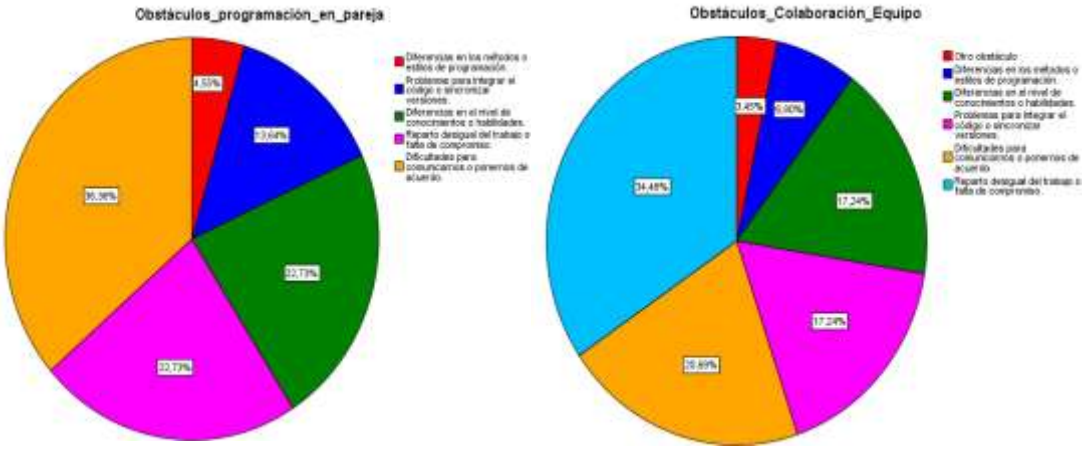
Los trabajos publicados en esta revista están bajo la licencia Creative Commons Atribución – No comercial

Git, evidenciando la complejidad intrínseca que conlleva la coordinación del trabajo de varios integrantes en un mismo repositorio representa desafíos técnicos y organizativos significativos.

Posteriormente, se analizan las dificultades actitudinales encontradas en ambos métodos colaborativos.

Ilustración 4:

Comparativa de las dificultades actitudinales trabajo en pareja vs. trabajo en equipo



Fuente: Elaboración propia

En el caso de la programación en pareja, los obstáculos más frecuentes son la dificultad para comunicarse o ponerse de acuerdo con un 36,36%, seguidos del reparto desigual del trabajo o falta de compromiso y las diferencias en el nivel de conocimiento o habilidades, ambos con un 22,76%. Los obstáculos con menos relevancia son los problemas para integrar el código o sincronizar versiones con un 13,64% y las diferencias en los métodos o estilos de programación con un 4,55%.

Por otro lado, la colaboración en equipo presenta como obstáculo más relevante en la programación el reparto desigual del trabajo o la falta de compromiso con un 34,48%, seguido de las dificultades para comunicarse o ponerse de acuerdo con un 20,69%. También se observan diferencias en problemas para integrar el código o sincronizar versiones y diferencias en el nivel de conocimiento o habilidades, ambas con un 17.24%, en menor nivel, las diferencias en los métodos o estilos de programación con un 6.90% y otro obstáculo representado con un 3.45%.

Se demuestra que en la programación en pareja predominan los conflictos interpersonales directos relacionados con la comunicación y la coordinación mutua para

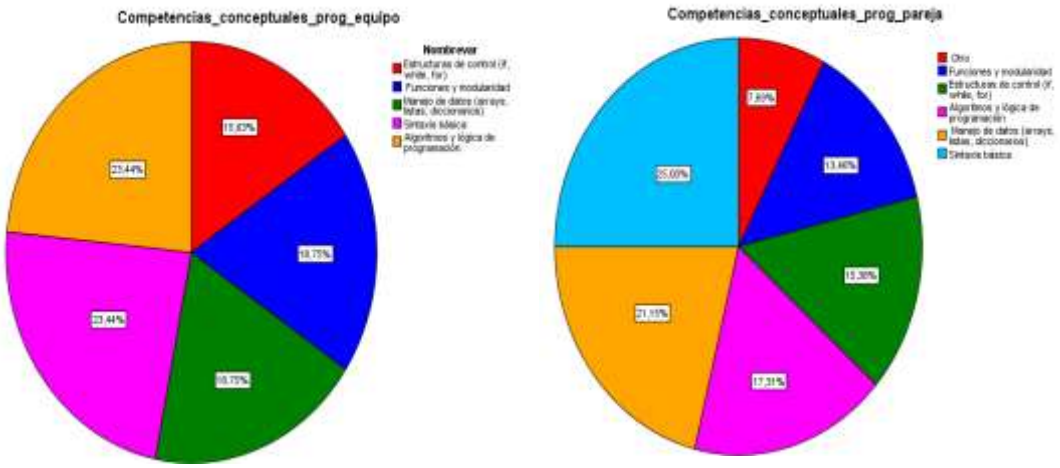
Los trabajos publicados en esta revista están bajo la licencia Creative Commons Atribución – No comercial

la aplicación de los aspectos técnicos, mientras que en la programación en equipo el reto se centra en la gestión del trabajo, específicamente en el reparto equitativo de tareas y en que cada uno de los integrantes del equipo esté individualmente comprometido con el trabajo colaborativo.

Posteriormente, se evidencian las competencias conceptuales identificadas en la colaboración en equipo y en pareja.

Ilustración 5

Comparativas competencias conceptuales en equipo vs. en pareja



Fuente: Elaboración propia

En el trabajo colaborativo en grupo, destacan las competencias fundamentales de programación relacionadas al dominio de sintaxis básicas y algoritmos de lógica de programación, ambas con 23.44%; de igual forma, comparten el mismo porcentaje las funciones y la modularidad con el manejo de datos (arreglos, listas y diccionarios) con un 18.75%; en menor incidencia, las estructuras de control (if, while y for) con un 15.63%.

En la colaboración en pareja, hay un incremento en las competencias conceptuales relacionadas con la sintaxis básica, con un 25%, que permiten expresar correctamente un algoritmo en un lenguaje de programación, y en el manejo de datos (arrays, listas y diccionarios), con un 21,15%. En contraste, se evidencian descensos en algoritmos y lógica de programación con 17.31%; las estructuras de control (if, while y for) se mantienen muy similares con 15. 38%, asimismo, funciones y modularidad con un decremento, indicando 13.46%, y en otros presenta un 7.69%.

Los resultados indican que el trabajo en grupo favorece competencias conceptuales más amplias y estructurales, como el pensamiento lógico colaborativo, así

Los trabajos publicados en esta revista están bajo la licencia Creative Commons Atribución – No comercial

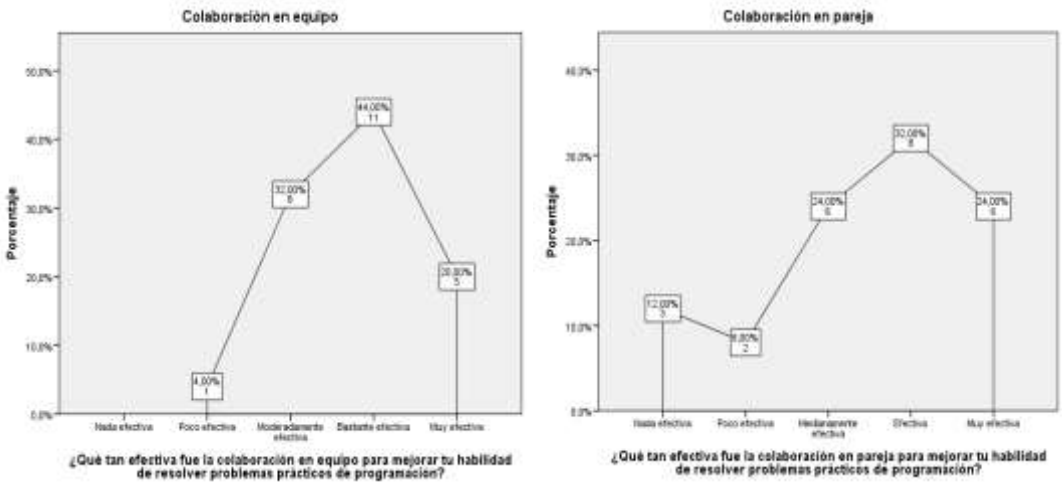
como el dominio de las reglas y estructuras básicas de la sintaxis en los lenguajes de programación, lo cual es coherente con la dinámica propia del trabajo en equipo.

En cambio, la programación en pareja muestra mayor énfasis en competencias básicas y estructura de código que les permiten comprender y aplicar razonamiento lógico en la resolución de problemas, especialmente la sintaxis y el manejo de datos. Estas competencias son más prácticas y de implementación directa, en un contexto donde ambos participantes programan activamente. Las estructuras de control se mantienen muy similares en ambos casos.

A continuación, se presentan las competencias procedimentales señaladas en la colaboración en equipo y en pareja.

Ilustración 6

Competencias procedimentales en equipo vs. en pareja



Fuente: Elaboración propia

En la colaboración en equipo, el 64% de los estudiantes considera que este método fue bastante y muy efectivo para mejorar la habilidad de resolver problemas prácticos de programación; además, un 32% lo califica como moderadamente efectivo, indicando que la mayoría reconoce beneficios notables y solo un 4% lo percibe como poco efectivo.

En cambio, en la colaboración en pareja la mayoría de los estudiantes lo considera como efectivo o muy efectivo con un 56%, un 24% la percibe como moderadamente efectivo, lo que indica que, aunque es útil, no siempre potencia al máximo las habilidades de resolución de problemas prácticos de programación para el desarrollo de las competencias procedimentales. Y solo un 20% lo califica como poco o nada efectivo.

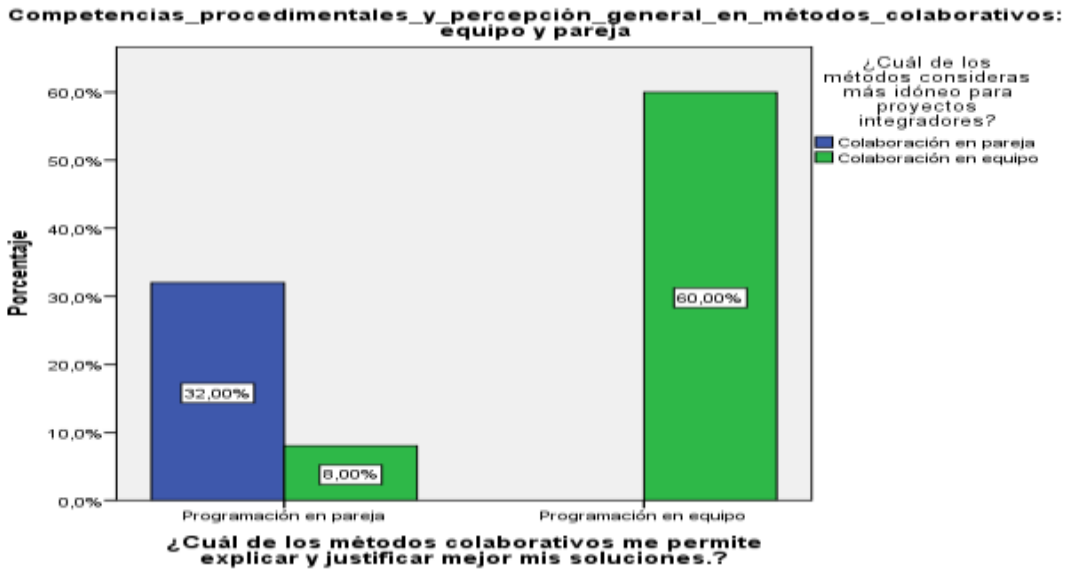
Los trabajos publicados en esta revista están bajo la licencia Creative Commons Atribución – No comercial

La interpretación de estos resultados indica que el trabajo en equipo se percibe como más efectivo en términos generales, dado que solo un 4% de los participantes lo considera poco efectivo. Por su parte, el trabajo en pareja también es bien valorado; sin embargo, podría limitar la diversidad de enfoques y la profundidad de discusión, como se refleja en el 20% que lo califica como nada y poco efectivo, lo que explica su porcentaje menor en las categorías más altas. No obstante, el 24% la percibe "muy efectiva", lo que sugiere que, cuando la colaboración funciona bien, puede ser más cercana y productiva.

Seguidamente, se muestran los resultados de los métodos colaborativos que permiten explicar y justificar mejor las soluciones de programación en el trabajo en equipo y en pareja.

Ilustración 7

Competencias procedimentales y percepción general de los métodos colaborativos



Fuente: Elaboración propia

En coherencia con estos hallazgos, comprender, explicar y justificar mejor las soluciones de programación es una de las competencias procedimentales que alcanza un valor del 60% para el método colaborativo en equipo. Por el contrario, la colaboración en pareja obtiene un 40%; se orienta más a la ejecución práctica que a la argumentación. Si bien la interacción entre dos estudiantes facilita la resolución de problemas de manera rápida, tiende a limitar la discusión conceptual y la construcción colectiva del conocimiento.

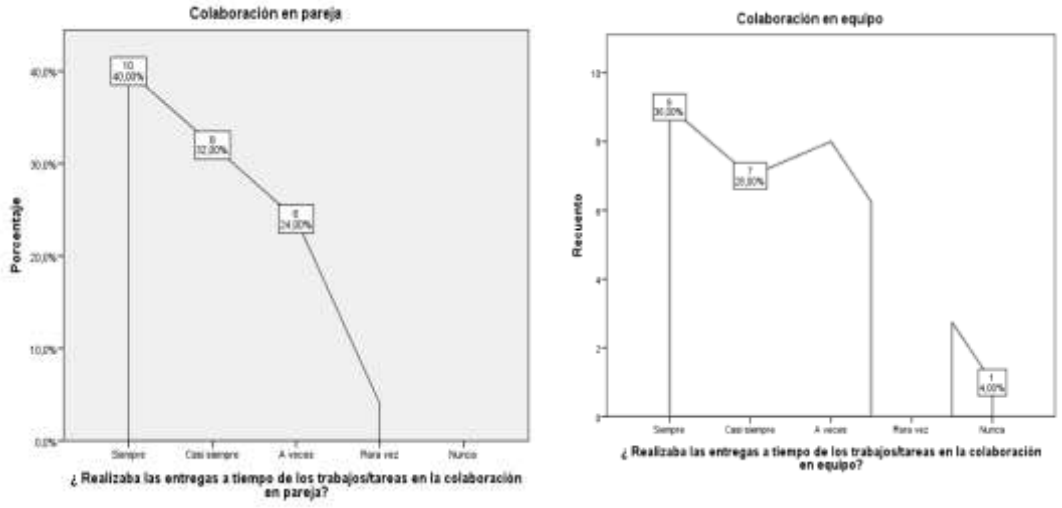
Los trabajos publicados en esta revista están bajo la licencia Creative Commons Atribución – No comercial

Lo anterior se complementa con la percepción general de los estudiantes respecto a la idoneidad de cada método colaborativo, donde el trabajo en equipo se consolida como el enfoque más adecuado, con un 68% que lo considera más efectivo para el desarrollo de competencias de programación en proyectos integradores. Esa valoración positiva refleja que la colaboración amplia y diversa puede generar aportes significativos en la resolución de problemas. En contraste, el método en pareja obtiene un 32%, lo que indica una menor percepción de efectividad en contextos que requieren integración multidimensional.

Finalmente, se presentan las competencias actitudinales identificadas en los métodos colaborativos en equipo y en pareja con respecto a la entrega puntual de trabajos/tareas.

Ilustración 8

Comparativa competencias actitudinales trabajo en equipo vs. en pareja



Fuente: Elaboración propia

Los resultados indican que la colaboración en pareja tiene consistencia alta en “siempre” y “casi siempre” con un 72% de puntualidad en las entregas, por el contrario, el trabajo en equipo tiene mayor concentración en la categoría intermedia “a veces”, lo que indica más variabilidad en el cumplimiento de los plazos de entrega, también se identifica una incidencia grave para la categoría de “nunca” con un 4% para la colaboración en equipo que no existe en la colaboración en pareja, ello indica la presencia ocasional de fallas o incumplimiento en las entregas en entornos grupales.

Los trabajos publicados en esta revista están bajo la licencia Creative Commons Atribución – No comercial

La interpretación de estos resultados muestra mayor responsabilidad percibida en pareja y más inconsistencias en el trabajo en equipo por la mayor complejidad para la coordinación y comunicación entre los integrantes.

Reflexiones finales y/o conclusiones:

El análisis comparativo realizado para determinar la efectividad de los métodos de aprendizaje colaborativos, tanto en equipo como en pareja, en el desarrollo de competencias en programación de los estudiantes de Ingeniería en Sistemas de Información del CUR-Chontales evidencia los siguientes resultados:

Primeramente, se identificaron las dificultades conceptuales; la colaboración en pareja resultó ser más efectiva, ya que reduce la percepción de dificultad en la comprensión teórica, en contraste con el trabajo en equipo donde se percibe una mayor complejidad en la asimilación conceptual. En cuanto a competencias conceptuales desarrolladas, el trabajo en pareja propicia la concentración y comprensión inmediatas de contenidos, mientras que el trabajo en equipo puede generar cierta confusión en el proceso de asimilación teórica.

Seguidamente, se abordaron las dificultades procedimentales; el trabajo en equipo presenta mayores retos para la integración del código y el uso adecuado de herramientas colaborativas como Git. En cambio, la colaboración en pareja refleja un equilibrio relacionado con la lógica, sintaxis y comprensión de algoritmos. En cuanto a competencias procedimentales obtenidas, el trabajo en equipo fortalece la capacidad de explicar y justificar mejor las soluciones a problemas de programación complejos, mientras que el trabajo en pareja potencia las prácticas habilitadas relacionadas con la sintaxis y el manejo de datos.

Por otra parte, en las dificultades actitudinales, la colaboración en pareja muestra inconvenientes relacionados con la comunicación y el establecimiento de acuerdos, mientras que en la colaboración en equipo predomina la distribución desigual de la carga de trabajo.

Finalmente, las competencias actitudinales adquiridas en la colaboración en pareja y la entrega puntual de tareas resultan más consistentes frente a la colaboración en equipo, donde existe incumplimiento en las entregas.

En resumen, los estudiantes perciben el método de aprendizaje colaborativo en grupo como más efectivo para potenciar la integración y favorecer el desarrollo de las competencias.

Los trabajos publicados en esta revista están bajo la licencia Creative Commons Atribución – No comercial

Referencias Bibliográficas:

Aburto Jarquín, P. (s.f.). *UNAN-Managua*. UNAN-Managua: https://www.unan.edu.ni/index.php/articulos-reportajes/el-modelo-curricular-por-competencias-aflora-logros-en-el-trabajo-integral-en-los-profesores-y-muy-especialmente-en-los-estudiantes-una-muestra-como-evidencia.odp

Gomes, A., Areias, C., Henriques, J., & Mendes, A. (2008). Aprendizagem de programação de computadores: dificuldades e ferramentas de suporte. *Revista Portuguesa de Pedagogia, 42-2*(161-179). https://doi.org/https://doi.org/10.14195/1647-8614_42-2_9

Herlambang, A., Ramadana, M., Wijoyo, S., & Phadung, M. (2024). Students' Cognitive Load on Computer Programming Instructional Process Using Example-Problem-Based Learning and Problem-Based Learning Instructional Model at Vocational High School. *Elinvo (Electronics, Informatics, and Vocational Education), 9*(2), 309-320. https://doi.org/ https://doi.org/10.21831/elinvo.v9i2.57882

Insuasti, J. (2016). Problemas de enseñanza y aprendizaje de los fundamentos de programación. *Educación y Desarrollo Social, 10*(2), 234-246. https://doi.org/https://doi.org/10.18359/REDS.1966

Qian, Y., & Lehman, J. (2017). Students' Misconceptions and Other Difficulties in Introductory Programming: A Literature Review. *ACM Transactions on Computing Education, 18*(1). https://doi.org/https://doi.org/10.1145/3077618

Quintana Peña, A. (2006). Metodología de investigación científica cualitativa. *Metodología de investigación científica cualitativa*.

UNAN-Managua. (20 de octubre de 2023). UNAN-Managua: https://www.unan.edu.ni/index.php/notas-informativas/mas-de-cien-estudiantes-de-unan-managua-participan-en-hackathon-nicaragua-2023.odp

Universidad Nacional Autónoma de Nicaragua, U.-M. (2023). *Documento Curricular de la carrera de Ingeniería en Sistemas de Información [Documento interno]*. UNAN-Managua, Universidad Nacional Autónoma de Nicaragua;.

Los trabajos publicados en esta revista están bajo la licencia Creative Commons Atribución – No comercial