

<https://doi.org/10.23913/ride.v14i28.1850>

Artículos científicos

Comparativa del aprendizaje basado en proyectos para el desarrollo de software en grupo

Comparison of project-based learning for group software development

Comparaçãõ de aprendizagem baseada em projetos para desenvolvimento de software em grupo

José Mendivil-Torres

Universidad Autónoma de Sinaloa, Facultad de Ingeniería Mochis, México
jose.mendivil@uas.edu.mx
<https://orcid.org/0000-0002-7621-8722>

Alan Ramírez-Noriega

Universidad Autónoma de Sinaloa, Facultad de Ingeniería Mochis, México
alandramireznoriega@uas.edu.mx
<https://orcid.org/0000-0002-8634-9988>

Carolina Tripp-Barba

Universidad Autónoma de Sinaloa, Facultad de Informática Mazatlán, México
ctripp@uas.edu.mx
<https://orcid.org/0000-0002-4811-0247>

José Alfonso Aguilar-Calderón

Universidad Autónoma de Sinaloa, Facultad de Informática Mazatlán, México
ja.aguilar@uas.edu.mx
<https://orcid.org/0000-0003-2048-9600>

Samantha Jiménez

Universidad Autónoma de Baja California, México
samantha.jimenez@uabc.edu.mx
<https://orcid.org/0000-0003-0938-7291>

Resumen

El aprendizaje basado en proyectos es una estrategia metodológica que implementa tareas sustentadas en la resolución de problemas a través de un proceso de investigación o creación por parte de los estudiantes, quienes trabajan con relativa autonomía, alto nivel de implicación y cooperación para procurar conseguir un producto. Por ende, esta investigación tiene como objetivo determinar el impacto del aprendizaje basado en proyectos en una carrera de ingeniería de *software*. En concreto, se realizó un experimento donde se aplicaron tres proyectos de desarrollo de *software* a tres grupos diferentes, para lo cual se monitoreó el desempeño de ellos y se analizó el conocimiento técnico, metodológico y organizacional de cada uno para medir el impacto en su desempeño académico. Los resultados muestran diferentes niveles de organización y capacidades entre los grupos para cumplir con el proyecto en los tiempos establecidos.

Palabras claves: aprendizaje basado en proyectos, ingeniería de *software*, desarrollo de *software*, trabajo en equipo, proyecto grupal.

Abstract

Project-based learning is a methodological strategy that implements tasks based on problem solving, through a process of research or creation by students who work relatively autonomously and with a high level of involvement and cooperation that culminates in a product. This research aims to determine the impact of project-based learning on a software engineering career. However, the research has the characteristic that the whole group focuses on developing the same project. An experiment was carried out where three software development projects were applied to three different groups, for which the performance of the groups was monitored. In the end, a survey was applied to determine the impact of the development of projects in a group. In the results, the groups showed different levels of organization and capacities to fulfill the project in the established times. The groups' technical, methodological, and organizational knowledge was analyzed to measure the impact on their academic performance. This research is helpful since it allows measuring the impact of project-based learning in groups and defines some guidelines to carry out said pedagogical strategy in software development.

Keywords: Project-based learning, software engineering, software development, teamwork, group project.

Resumo

A aprendizagem baseada em projetos é uma estratégia metodológica que implementa tarefas baseadas na resolução de problemas através de um processo de pesquisa ou criação por parte dos alunos, que trabalham com relativa autonomia, elevado nível de envolvimento e cooperação para tentar alcançar um produto. Portanto, esta pesquisa tem como objetivo determinar o impacto da aprendizagem baseada em projetos na carreira de engenharia de software. Especificamente, foi realizada uma experiência onde foram aplicados três projetos de desenvolvimento de software a três grupos diferentes, para os quais foi monitorado o seu desempenho e analisado o conhecimento técnico, metodológico e organizacional de cada um para medir o impacto no seu desempenho acadêmico. Os resultados mostram diferentes níveis de organização e capacidades entre os grupos para concluir o projeto dentro dos prazos estabelecidos.

Palavras-chave: aprendizagem baseada em projetos, engenharia de software, desenvolvimento de software, trabalho em equipe, projeto em grupo.

Fecha Recepción: Agosto 2023

Fecha Aceptación: Marzo 2024

Introducción

La habilidad más importante en la era digital que deben adquirir los estudiantes es la de aprender a aprender, de ahí que el aprendizaje haya pasado de ser una construcción individual de conocimiento a un proceso social. Sin embargo, aun para los docentes más experimentados, mantener a los estudiantes comprometidos y motivados constituye un gran reto (Zambrano Briones *et al.*, 2022), aunque existen diversas estrategias didácticas que pueden ser aplicadas tomando en cuenta las características de los estudiantes. Algunos de ellos, según Chimbo Jumbo y Larreal Bracho (2023), pueden ser los siguientes: 1) enfoque didáctico para la individualización, como la enseñanza programada, el aprendizaje autodirigido y la tutoría académica; 2) enfoque de socialización didáctica, con modelos como el método del caso, seminario y tutorías entre iguales, y 3) enfoque globalizado, como la resolución de problemas y el aprendizaje basado en proyectos (ABP), el cual es el objeto de análisis en la presente investigación.

El ABP es una metodología en la que los alumnos tienen un rol activo con el fin de promover la motivación académica. En este modelo, la actividad principal para la adquisición de los objetivos formativos se basa en el desarrollo de un proyecto que trata de responder a una necesidad real, lo cual, normalmente, supone la creación de un producto final. Para

lograrlo, los alumnos deben adquirir o practicar las competencias que el profesor se ha planteado para la instrucción. Habitualmente, esta técnica va unida a un método de trabajo en grupo, de forma que el proyecto es desarrollado por un equipo de estudiantes que se organizan para alcanzar los objetivos finales (Cyrulies y Schamne, 2021; Marnewick, 2023).

Este enfoque puede ser abordado de forma intradisciplinaria e interdisciplinaria, lo cual permite favorecer el trabajo colaborativo entre docentes y con el centro educativo (Al Mulhim y Eldokhny, 2020; Sotomayor *et al.*, 2021). Su principal característica, por tanto, es brindar a los alumnos un contexto real e involucrarlos directamente en el proceso de enseñanza-aprendizaje. En otras palabras, el estudiante es el encargado de tomar una serie de decisiones para resolver una tarea de cierto nivel de complejidad (Abella García *et al.*, 2020; Morais *et al.*, 2021; Pérez y Rubio, 2020).

En este sentido, se ha demostrado que los alumnos que han participado en un ABP muestran una mejor capacidad para resolver asignaciones, además de ser más autosuficientes y comprometidos con su aprendizaje, de modo que contribuye a fomentar la autonomía, la confianza en sí mismos y a incrementar la motivación. Esto supone abandonar la enseñanza mecánica a favor de una metodología donde las tareas se plantean como retos en una dinámica común, en lugar de una asignación descontextualizada de trabajos inconexos (Abella García *et al.*, 2020).

Si bien el ABP es útil para muchas áreas, cobra especial relevancia en profesiones donde la producción o servicio se basa en la generación de proyectos. Por ello, y aunque hay muchas carreras donde se puede aplicar, esta investigación se centra en la Ingeniería de Software (IS), la cual se basa en la aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo de programas de cómputo. Esto incluye la creación de herramientas, métodos y técnicas para apoyar la producción de *software*, así como todos los aspectos relacionados con la gestión de proyectos, que involucran personas, procesos y herramientas tecnológicas.

En esta carrera se combinan aspectos técnicos de las ciencias de la computación con habilidades blandas como la comunicación, negociación, colaboración y trabajo en equipo, entre otros. De esta manera, el ingeniero de *software* puede desarrollar soluciones informáticas que satisfagan las necesidades de los interesados y aumenten la eficiencia de los procesos de negocio afectados (Gómez Álvarez *et al.*, 2015).

La importancia de las habilidades blandas en el ejercicio profesional del ingeniero de *software* ha impulsado el desarrollo de estrategias dirigidas a su integración en el proceso de enseñanza-aprendizaje, las cuales buscan crear entornos de colaboración donde los

estudiantes puedan cultivar la creatividad y habilidades sociales fundamentales para su práctica ingenieril, como la comunicación efectiva, el liderazgo, la capacidad de negociación y el trabajo en equipo. Estas experiencias también procuran que los estudiantes se involucren en la realidad de las organizaciones para reducir la brecha entre la universidad y la empresa, específicamente en lo que respecta al desarrollo de las habilidades requeridas por la industria y aquellas que se desarrollan durante la formación profesional del estudiante (Gómez Álvarez *et al.*, 2015; Nurbekova *et al.*, 2020). La integración de estas habilidades blandas en el proceso de enseñanza de ingeniería de *software* se ha llevado a cabo mediante 1) casos de estudio para simular entornos reales de desarrollo de *software* y 2) la ejecución de proyectos de *software* universidad-empresa.

En el ámbito educativo, el objetivo general de la asignatura de Ingeniería de Software es que los alumnos adquieran las habilidades para desarrollar proyectos de *software* desde su concepción hasta su entrega al cliente. Por lo tanto, los estudiantes deben adquirir los conocimientos y habilidades necesarias para el desarrollo completo de un proyecto de *software* (Sánchez y Blanco, 2012). Los estudiantes de la carrera Ingeniería de Software, al enfrentarse a un ABP, deben 1) enfrentar información incompleta e imprecisa, 2) autorregularse y comprometerse con el trabajo, de modo que puedan comprometerse con el proceso de aprendizaje al definir sus propios objetivos dentro de los límites establecidos, 3) cooperar y trabajar en grupos para dividir la carga de trabajo entre ellos e integrar las diferentes partes desarrolladas por cada uno, y (4) trabajar con temas multidisciplinarios, habilidades prácticas que son requisitos importantes en el ámbito empresarial e industrial.

El ABP es relevante para el desarrollo de *software*, ya que es un método de enseñanza sistemático en el cual los estudiantes participan en la aplicación de conocimientos esenciales y habilidades para la vida a través de un extenso y estructurado proceso de indagación influenciado por el estudiante alrededor de complejas preguntas auténticas con el fin de crear productos de alta calidad (Villalobos-Abarca *et al.*, 2018). En otras palabras, mediante el uso del ABP, los estudiantes se vuelven más responsables de desarrollar su conocimiento.

Asimismo, se cree que fomenta un aprendizaje más profundo en comparación con las conferencias tradicionales, donde los estudiantes retienen menos de lo que se les enseña. De hecho, se estima que la evaluación en el ABP también se ve influenciada por la cercanía a situaciones del mundo real. Dado que los artefactos construidos durante el proyecto son en sí mismos representaciones del aprendizaje, es importante proporcionar retroalimentación auténtica y constructiva para los objetivos del problema (Da Cunha, 2005).

El ABP ha sido ampliamente adoptado en la educación en ingeniería, y se define como un método instruccional de enseñanza constructivista que utiliza problemas reales como elemento motivador para el aprendizaje. Varios estudios en educación informática han evidenciado que el ABP facilita el compromiso y la capacidad de aprendizaje de los alumnos, ya que contribuye al desarrollo de habilidades como el trabajo en equipo, visión holística, pensamiento crítico y resolución de problemas. De hecho, algunas investigaciones han demostrado que el ABP es especialmente adecuado para la enseñanza de la Ingeniería de Software, pues permite a los estudiantes estimular el aprendizaje profundo y aplicarlo desde el inicio del curso. Aunado a esto, el hecho de ver materializada una solución de *software* que no se aleja de la práctica industrial y que utiliza tecnologías de última generación representa un importante estímulo adicional (Adorjan y Solari, 2021).

Considerando estos aspectos, la presente investigación aplicó un caso de estudio de ABP en una carrera de Ingeniería de Software para determinar el impacto de la metodología considerando grupos completos de trabajo, es decir, la totalidad del grupo sobre el mismo proyecto. Para eso, se formaron tres grupos con diferentes proyectos de desarrollo de *software*, partiendo desde cero. Los grupos debían generar una aplicación de *software* como entregable, así como documentación sobre dicho proceso. Una característica especial fue que los proyectos estaban organizados por grupos, es decir, todo el grupo estaba enfocado en desarrollar el mismo proyecto, por lo que los roles del desarrollo de *software* estaban distribuidos entre equipos.

La pregunta de investigación planteada fue la siguiente: ¿cuáles son las principales competencias que se pueden destacar en la práctica del aprendizaje grupal basado en proyectos para el desarrollo de *software*?

La estructura del artículo es la siguiente: en la sección 2 se ofrece el estado de la cuestión en torno al ABP. En el tercer apartado se destaca el método de investigación empleado, mientras que en la sección 4 se detalla el experimento realizado y los resultados obtenidos. En la sección 5 se discuten los resultados y limitaciones del estudio, y las conclusiones e ideas para trabajos futuros se presentan en la sección 7.

Trabajos relacionados

El proceso de aprendizaje requiere cambios en las metodologías pedagógicas. En tal sentido, una de las más populares en los últimos años es el aprendizaje basado en proyectos (ABP), el cual representa una de las pedagogías más exitosas centradas en el estudiante y una de las más ampliamente utilizadas en cursos de desarrollo de *software* (Macías, 2012). La principal característica de este método es que los estudiantes se involucran en un escenario real donde participan directamente en el proceso de enseñanza-aprendizaje para resolver una tarea de alto nivel (Abella García *et al.*, 2020). Con este enfoque se procura fomentar la motivación, la colaboración con los miembros del equipo, la proactividad y las habilidades blandas, herramientas clave para el éxito en proyectos y en entornos de enseñanza-aprendizaje (Sánchez y Blanco, 2012).

En un estudio realizado por Sánchez y Blanco (2012), los autores compararon la diferencia entre estudiantes que recibían conferencias y aquellos matriculados en proyectos prácticos e inmersivos en escenarios reales. Según los resultados hallados, los alumnos que aprendieron utilizando el enfoque práctico mejoraron sus calificaciones finales en aproximadamente un 35 %, lo cual sugiere que el aprendizaje puede estimularse de mejor manera a través de este método.

Otros investigadores, como Macías (2012), se centraron en evaluar la motivación y la percepción del estudiante en este tipo de entorno de aprendizaje. Para ello, realizaron una encuesta con el fin de evaluar la satisfacción de los estudiantes y utilizaron las evaluaciones de la calidad educativa de los estudiantes (SEEQ, Student Evaluations of Educational Quality). Este es un cuestionario ampliamente usado en evaluación académica para medir características psicométricas valiosas como confiabilidad, validez, consistencia interna, etc. Sus hallazgos sugieren que mejores calificaciones de los estudiantes han complementado el aumento en su satisfacción; además, se produjo un aumento del 40 % respecto a años anteriores en el número de alumnos que aprobaban la carrera, y un aumento del 30 % en la tasa de alumnos que obtienen mejores notas (Macías, 2012). Los instructores involucrados en el experimento se reunieron al final y analizaron el progreso en los laboratorios de ingeniería de *software*. Sobre esto, informaron una clara mejora en la homogeneidad gracias al proceso, aunque, como punto negativo, todos los instructores señalaron una mayor carga de trabajo con el nuevo sistema, particularmente al completar las rúbricas y escribir comentarios detallados a los estudiantes (Macías, 2012). Esto demuestra que uno de los

principales problemas del ABP es la evaluación, ya que el proceso debe considerar las contribuciones de cada miembro del equipo al proyecto.

Aun así, Abella García *et al.* (2020) sugieren el uso de rúbricas, componentes de calificación y un porcentaje asignado a cada uno (Cyrulies y Schamne, 2021), de modo que permita considerar todos los aspectos para mejorar los criterios de calificación y reducir el número de injusticias. No obstante, Da Cunha (2005) enumera algunas experiencias negativas en la implementación de este enfoque. En su experimento, por ejemplo, participaron 88 estudiantes y aproximadamente el 95 % aprobaron la asignatura. A pesar de que en general la experiencia fue positiva, los estudiantes enfrentaron algunos problemas, como trabajar en grupo y conflictos debido a opiniones divididas. Además, tomaron decisiones deficientes sobre algunas herramientas y fue necesario cambiar las aplicaciones un par de veces, así como migrar el proyecto a una nueva aplicación.

Aun así, cabe destacar que la implementación del ABP en las carreras de Ingeniería de Software ha reducido la brecha entre la industria y el enfoque teórico de todas las universidades o instituciones educativas (Villalobos-Abarca *et al.*, 2018). De hecho, aunque en algunos casos su empleo ha ocasionado algunas dificultades para los estudiantes, la experiencia también ha enseñado que tarde o temprano los estudiantes enfrentarán este tipo de problemas en escenarios reales.

Materiales y métodos

La presente investigación se sustentó en un enfoque mixto, ya que se combinaron tanto métodos cualitativos como cuantitativos. Es decir, por una parte, se realizan evaluaciones constantes de datos numéricos y estadísticos, y, por otra, se efectuó una inmersión en los grupos para definir cómo evolucionan los proyectos. Este enfoque mixto permite aprovechar las fortalezas de ambos métodos y compensar las debilidades de cada uno por separado, con lo cual se consigue un análisis objetivo e interpretativo de acuerdo con la experiencia de la inmersión en los grupos.

La investigación, por otra parte, tuvo un componente de trabajo de campo, lo cual significa estar inmerso en el proceso de desarrollo del proyecto para guiar y asesorar a los grupos, así como explicar los aspectos técnicos del proceso. Esto se concreta en dos escenarios:

1. El estudio se inicia en el aula de clases de la Facultad de Ingeniería Mochis, donde se brindan las asesorías del proyecto y se cubren los aspectos técnicos de la materia.

2. Los estudiantes trabajan en sus hogares, espacio donde se lleva a cabo el mayor avance del proyecto debido a que existen menos distractores y, por tanto, se pueden incrementar los niveles de concentración.

Tipo de investigación

Se efectuó un cuasiexperimento, ya que los grupos estaban preasignados de acuerdo con la organización académica de la Facultad de Ingeniería Mochis (Universidad Autónoma de Sinaloa). Estos grupos se formaron al inicio de la carrera y pueden experimentar cambios debido a rendimientos académicos bajos o motivos personales, pero la base de ellos se mantiene constante desde el inicio.

Asimismo, la investigación adoptó un enfoque exploratorio con el objetivo de identificar relaciones entre variables que puedan explicar el comportamiento de los grupos. Luego, se realizó un análisis descriptivo para identificar aspectos importantes basados en tablas y gráficas y, posteriormente, se efectuó un análisis basado en pruebas de hipótesis, lo cual permitió contrastar variables para identificar aquellas que afecten el rendimiento de los grupos.

La población de estudio estuvo conformada por los alumnos de la carrera de Ingeniería de Software de la Facultad de Ingeniería Mochis de la Universidad Autónoma de Sinaloa, específicamente aquellos que cursan el sexto semestre. Estos estudiantes son en su mayoría hombres entre 20 y 22 años con un perfil tecnológico que incluye conocimientos sobre el manejo de computadoras, lenguajes de programación, bases de datos y algunos con experiencia en desarrollo web. El experimento incluyó a tres grupos por semestre: dos vespertinos y uno nocturno, es decir, se consideraron todos los alumnos que actualmente cursan dicha materia.

Procedimiento

El proyecto de investigación recopiló información para análisis de la siguiente forma:

- Los estudiantes entregan avances del desarrollo del proyecto periódicamente, los cuales son presentados ante el grupo. Luego, el profesor registra estos avances y lleva un control del estado del proyecto.
- Se recopilan datos a través de una encuesta general aplicada a todos los miembros del grupo. Esta evaluación se realizó mediante formularios en línea.

- Se realizó una observación participativa durante las clases en el aula con los grupos, donde se explicaron conceptos y se brindó asesoramiento para resolver dudas sobre los proyectos.

El procedimiento seguido en el experimento fue el siguiente:

1. Se generaron tres proyectos relacionados con el desarrollo de *software* y su lista de requerimientos. Los nombres de los proyectos fueron 1) desarrollo de un punto de venta, 2) automatización de la estimación de *software* con *planning poker*, y 3) construcción de un *software* colaborativo para el lenguaje de señas mexicano. Cada proyecto es diferente entre sí, pero todos requieren una aplicación web y generar la documentación de ingeniería de *software*.
2. Los proyectos se asignaron de forma aleatoria utilizando un sistema web gratuito que genera números aleatorios. Al grupo 1 se asignó el proyecto desarrollo de un punto de venta, al grupo 2 el proyecto automatización de la estimación de *software* con Planning Poker, y al grupo 3 el proyecto construcción de un *software* colaborativo para el lenguaje de señas mexicano.
3. Se explicó la teoría de los roles en los proyectos de *software* para que los alumnos los conocieran y eligieran dos: uno principal y uno secundario. Esto se hizo para distribuir los roles entre los estudiantes y evitar que solo seleccionaran uno. Los roles incluyeron analista, diseñador, programador, *tester*, documentador y líder de proyecto.
4. La materia donde se desarrolló todo el proyecto fue Desarrollo de Aplicaciones Web, la cual sirvió como base para explicar temas como HTML, CSS, JavaScript y el lenguaje del lado del servidor PHP. Todos los proyectos debían aplicar estas tecnologías, aunque no era obligatorio; si algún grupo quería emplear otras tecnologías, eran libres de hacerlo.
5. El profesor asignado en la materia fungió como el cliente del proyecto, es decir, la persona que solicitaba el proyecto. Al inicio, cada grupo debía fijar fechas para entrevistas con el cliente y empezar a generar los requerimientos del *software*. Al final, se contrastaban los requerimientos generados por el grupo con los originales establecidos por el profesor, lo cual servía como punto de comparación para medir el entendimiento del problema por parte de los analistas. Posteriormente, se proporcionaban los requerimientos originales del proyecto para evitar confusiones.
6. Los grupos de trabajo debían desarrollar diversos artefactos de Ingeniería de Software, tales como documentación de generación de requerimientos, diagramas de

casos de uso, casos de uso, diagramas de clases, diagramas entidad-relación, diseño de interfaces, entre otros.

7. Conforme se desarrollaban las clases, se solicitaban avances sobre el proyecto, y cada grupo debía explicar sus avances de acuerdo al progreso de sus roles. Los líderes de los proyectos fungían como coordinadores del grupo, y la comunicación y exigencia hacia los equipos se realizaba a través de ellos.
8. Al finalizar el semestre cada grupo debía presentar su proyecto ya terminado, al menos hasta el nivel avanzado.

Estos aspectos fueron los mismos para todos los estudiantes, aunque los proyectos eran diferentes para evitar copias de código, documentación o ideas entre los grupos; además, cada grupo debía desarrollar su proyecto de forma independiente.

Proyectos

Los proyectos asignados a los grupos se describen de la siguiente manera:

1. Desarrollo de un punto de venta: Desarrollar un sistema de punto de venta en línea con soporte para cotizaciones. Este debe permitir dar de alta productos y gestionar el *stock* de ventas. Además, debe funcionar con una pistola de captura de código de barras.
2. Automatización de la estimación de *software* con Planning Poker: Se trata de implementar un sistema de estimación de historias de usuario para Scrum utilizando el método de Planning Póker. El sistema debe permitir estimar la complejidad de las historias de usuario a través de cartas virtuales Planning Póker. Además, debe incluir funciones de gestión de proyectos, historias de usuarios, miembros y un historial de estimaciones.
3. Construcción de un *software* colaborativo para el lenguaje de señas mexicano: El proyecto implica desarrollar un repositorio de videos de corta duración para el lenguaje de señas mexicano. Este debe permitir dar de alta videos y realizar búsquedas, así como incluir la funcionalidad de evaluar videos para su posible eliminación si los usuarios así lo deciden.

Instrumento

Al finalizar las fechas de entrega de los proyectos, se aplicó una encuesta para recopilar datos y realizar un análisis que permitiera identificar posibles relaciones entre variables. A continuación, se detallan los diversos bloques que conformaron el instrumento utilizado en dicha encuesta.

Datos generales

Se recabaron datos básicos como nombre, apellido, edad, sexo, promedio hasta la fecha, grupo al que pertenecen, rol desempeñado en el proyecto y si se trata de un alumno regular o no.

Preguntas relacionadas al proyecto grupal

Las preguntas planteadas en esta sección se estructuraron en formato Likert de 5 puntos: totalmente en desacuerdo (1), en desacuerdo (2), neutral (3), de acuerdo (4) y totalmente de acuerdo (5). A continuación, se presentan las preguntas o afirmaciones planteadas:

1. He participado en otro proyecto grupal similar en este semestre o anteriormente (sí, no)
2. El proyecto grupal me ha parecido muy útil para la colaboración entre los miembros del grupo (Likert de 5 puntos).
3. El proyecto grupal me ha parecido muy útil para percibir la función de cada rol en un proyecto de *software* (Likert de 5 puntos).
4. Me gustaría desarrollar otro proyecto similar en otras materias (Likert de 5 puntos).
5. La temática del proyecto me pareció muy compleja (Likert de 5 puntos).
6. El proyecto era muy complejo para nuestros conocimientos técnicos (uso de lenguajes de programación, gestores de bases de datos, repositorios de código y documentos, etc.) (Likert de 5 puntos).
7. El proyecto era muy complejo para nuestros conocimientos metodológicos (metodologías de desarrollo de software, actividades de los roles, desarrollo de documentación, etc.) (Likert de 5 puntos).
8. El proyecto era muy complejo para la capacidad organizacional del grupo (organización de actividades de trabajo, comunicación entre grupos, asistencia a reuniones) (Likert de 5 puntos).



9. Consideras que el tiempo de inicio y fin del del proyecto era (poco tiempo, tiempo adecuado, mucho tiempo).
10. Consideras que hizo falta otro rol para que el proyecto se desarrollará de mejor forma (sí [cual], no).
11. Cómo calificarías tu desempeño en el proyecto grupal (muy malo, malo, neutral, bueno, muy bueno).
12. Hay algún aspecto en general (que quisieras comentar) que hubiera generado que el proyecto grupal se desarrollará en mejor forma.
13. Si el proyecto se repitiera, ¿elegirías el mismo rol?
14. Elegiste cambiar de rol, ¿cuál rol elegirías?
15. ¿Porque cambiarías de rol?

Competencias transversales o genéricas

Estas competencias son esenciales para que los individuos sean productivos desde su ingreso al mundo laboral, y no están específicamente relacionadas con un área profesional en particular. Las respuestas a estas preguntas se basaron en el formato Likert, que evalúa la capacidad de la persona para cumplirlas, del siguiente modo: muy incapaz (1), incapaz (2), neutral (3), capaz (4), y muy capaz (5). A continuación, se detallan las competencias junto con su definición:

1. Comunicación oral y escrita: Transmitir conocimientos, expresar ideas y argumentos de manera clara, rigurosa y convincente, tanto de forma oral como escrita, utilizando los recursos gráficos y los medios necesarios adecuadamente que permitan adaptarse a las características de la situación y de la audiencia.
2. Análisis y síntesis de información: Reconocer y describir los elementos constitutivos de una realidad, y proceder a organizar la información significativa según criterios preestablecidos adecuados a un propósito.
3. Planteamiento y resolución de problemas: Analizar los elementos constitutivos de un problema para idear estrategias que permitan obtener, de forma razonada, una solución contrastada y adecuada para ciertos criterios preestablecidos.
4. Modelación de soluciones: Analizar los fundamentos y propiedades de modelos existentes, y traducir e interpretar los elementos del modelo en términos del mundo real.
5. Aprendizaje autónomo: Aprender por iniciativa e interés propio a lo largo de la vida.

6. Trabajo en equipo: Participar de manera efectiva en equipos diversos y colaborar de forma activa en la consecución de objetivos comunes.
7. Toma de decisiones: Identificar patrones que anticipan posibles explicaciones y/o soluciones a los problemas industriales, tecnológicos y operativos para una adecuada toma de decisiones.
8. Uso efectivo de herramientas de TIC: Capacidad de actualización respecto al uso de la tecnología en el área que repercuta en su mejora continua, incluyendo las nuevas tecnologías.
9. Responsabilidad en la actuación: Entendimiento de los aspectos profesionales, éticos, legales, de seguridad y sociales, así como de la responsabilidad inherente en cada uno de ellos.
10. Visión sobre el impacto de las soluciones: La habilidad para analizar el impacto local y global de las soluciones de TI en las personas, organizaciones y en la sociedad en general.

Competencias específicas

Las preguntas 11 a 22 corresponden a las competencias específicas del perfil de ingeniero de *software* definidas por la Asociación Nacional de Instituciones de Educación en Tecnologías de Información (ANIEI) (2022) y el Consejo Nacional de Acreditación en Informática y Computación (CONAIC) (2023), las cuales deben de adquirirse mientras estudian la licenciatura en Ingeniería en Software. Las respuestas tipo Likert fue de 5 puntos: muy incapaz (1), capaz (2), neutral (3), capaz (4) y muy capaz (5):

1. Realiza ingeniería de requisitos de *software*: Reconocer el contexto, necesidades e involucrados en un sistema empleando técnicas para identificar, obtener, analizar, priorizar, documentar, verificar y validar los requisitos en el contexto de los ciclos de vida y procesos del desarrollo de *software*.
2. Diseña *software*: Diseñar el comportamiento, arquitectura e interfaz de soluciones de *software* a partir de requerimientos y utilizando estrategias, métodos, técnicas y lenguajes de modelado propios del diseño de *software*.
3. Construye *software*: Desarrollar *software* para diferentes tipos de aplicaciones, utilizando metodologías y paradigmas de programación en el contexto de los ciclos de vida y procesos del desarrollo de *software*, con los atributos de calidad requeridos.

4. Realiza pruebas de *software*: Planear, asignar y ejecutar tipos, técnicas, procesos y controles dentro de escenarios de pruebas conforme a los atributos de calidad requeridos.
5. Realiza mantenimiento de *software*: Aplicar tipos, procesos y técnicas de mantenimiento conforme a los atributos de calidad requeridos.
6. Administra proyectos de *software*: Usar métodos, estrategias, procesos, herramientas y técnicas para la gestión de proyectos de *software*.
7. Estima parámetros del proyecto de *software*: Aplicar métricas para la estimación del software (tamaño, costo, esfuerzo, personal, tiempo, productividad, calidad y documentación) conforme a los modelos de ciclos de vida de los sistemas.
8. Asegura la calidad del *software*: Utilizar técnicas, herramientas y estrategias para planificar, asegurar y controlar la calidad de un producto de *software*.
9. Establece mecanismos de seguridad: Crear o proponer métodos y estrategias para evaluar la seguridad y la selección de los criterios que eviten vulnerabilidades en seguridad del *software*.
10. Emplea ciclos de vida: Emplear los elementos y criterios en el uso de los modelos de ciclos de vida conforme al contexto de procesos del desarrollo de *software*.
11. Verifica calidad de soluciones de *software*: Emplear diversos modelos de pruebas para garantizar la calidad del producto de *software*.
12. Usa herramientas para creación de *software*: Utilizar métodos industriales y herramientas CASE para las diferentes fases en el proceso de *software*.

Para el tratamiento y análisis de los datos, se utilizó la herramienta estadística SPSS. Asimismo, se llevó a cabo un análisis intrasujetos, lo que significa que el mismo experimento se aplicó a todos los estudiantes y se examinaron aspectos internos de la muestra para explorar posibles relaciones entre variables. Para identificar estas relaciones, el análisis de datos incluyó diversas pruebas estadísticas, como la prueba de Ji cuadrada, la prueba exacta de Fisher, la prueba t de Student y la prueba de Kruskal-Wallis (Flores-Ruiz *et al.*, 2017; Hernández-Sampieri y Mendoza, 2019).

Resultados

El experimento se enfocó en desarrollar proyectos de *software* en tres grupos de la carrera de Ingeniería de Software del sexto semestre: dos grupos del turno vespertino y uno del nocturno. Las características de los grupos se detallan en la tabla 1. El símbolo %



representa el porcentaje del número a la izquierda con respecto al total de alumnos. La palabra *DE* entre paréntesis representa la desviación estándar.

Tabla 1. Datos generales de los grupos participantes en el experimento.

| Grupos | Alumnos (%) | Hombres (%) | Mujeres (%) | Calificaciones (DE) | Edad (DE) | Irregulares (%) |
|---------|-------------|-------------|-------------|---------------------|--------------|-----------------|
| 1 | 23 (46.9) | 19 (82.6) | 4 (17.4) | 8.8 (0.75) | 20.91 (0.95) | 9 (39.1) |
| 2 | 19 (38.8) | 14 (73.7) | 5 (26.3) | 8.2 (0.78) | 21.21 (1.36) | 9 (47.4) |
| 3 | 7 (14.3) | 7 (100) | 0 (0) | 8.1 (0.92) | 25.43 (5.91) | 0 (0) |
| Totales | 49 (100) | 40 | 9 | 8.4 | 22.51 | 18 |

Fuente Elaboración propia

Como se puede observar en la tabla 1, participaron 49 alumnos, principalmente hombres. El promedio de calificaciones mencionado en la tabla corresponde al nivel de rendimiento académico reportado por los estudiantes en ese momento de la carrera, y puede variar ligeramente. Las edades de los estudiantes se sitúan alrededor de los 22.5 años, con un aumento notable en las edades de los estudiantes del turno nocturno, quienes suelen ser trabajadores y algunos tienen responsabilidades familiares.

Tabla 2. Distribución de estudiantes por rol y grupo al inicio (I) y fin (F) del semestre

| | Grupo | | | | | | Total | |
|--------------------|-------|----|----|----|----|---|-------|----|
| | 1 | | 2 | | 3 | | | |
| Rol | I | F | I | F | I | F | I | F |
| Analista | 5 | 5 | 3 | 3 | 3 | 2 | 11 | 10 |
| Diseñador | 6 | 6 | 3 | 3 | 2 | 0 | 11 | 9 |
| Programador | 5 | 4 | 7 | 6 | 4 | 0 | 16 | 10 |
| Tester | 4 | 4 | 3 | 2 | 2 | 2 | 9 | 8 |
| Documentador | 3 | 2 | 4 | 3 | 3 | 1 | 10 | 6 |
| Líder del proyecto | 2 | 2 | 2 | 2 | 2 | 2 | 6 | 6 |
| Total | 25 | 23 | 22 | 19 | 16 | 7 | 63 | 49 |

Fuente Elaboración propia

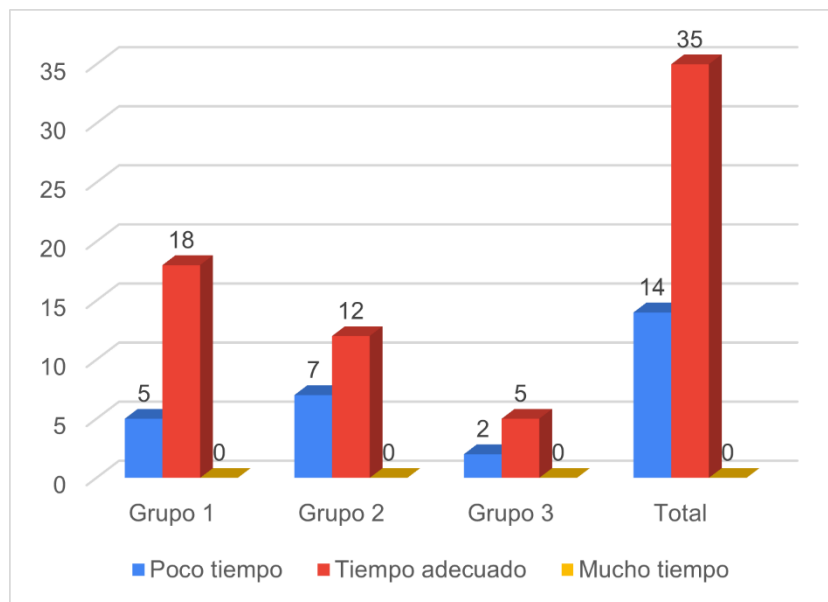
La tabla 2 muestra los roles que los alumnos seleccionaron al inicio del semestre. Aunque un total de 63 alumnos inicialmente eligieron un rol, al final del semestre solo 49 alumnos completaron la encuesta, lo que sugiere un alto índice de deserción en la materia, especialmente en el grupo 3. Según la tabla 3, en el grupo 3 se registraron deserciones en los

roles de diseñador y programador, lo que implicó que los demás miembros del grupo tuvieran que asumir esas responsabilidades adicionales.

Asimismo, se aplicó una encuesta a los estudiantes sobre su experiencia en el desarrollo de los proyectos. Cuando se les preguntó sobre el trabajo en grupos, el 94 % respondió que no tenían experiencia previa. Respecto al tiempo asignado para el desarrollo de los proyectos (seis meses, de enero a junio), la mayoría consideró que fue adecuado, aunque una minoría expresó que fue insuficiente. La figura 1 resume estas respuestas por grupos de trabajo.

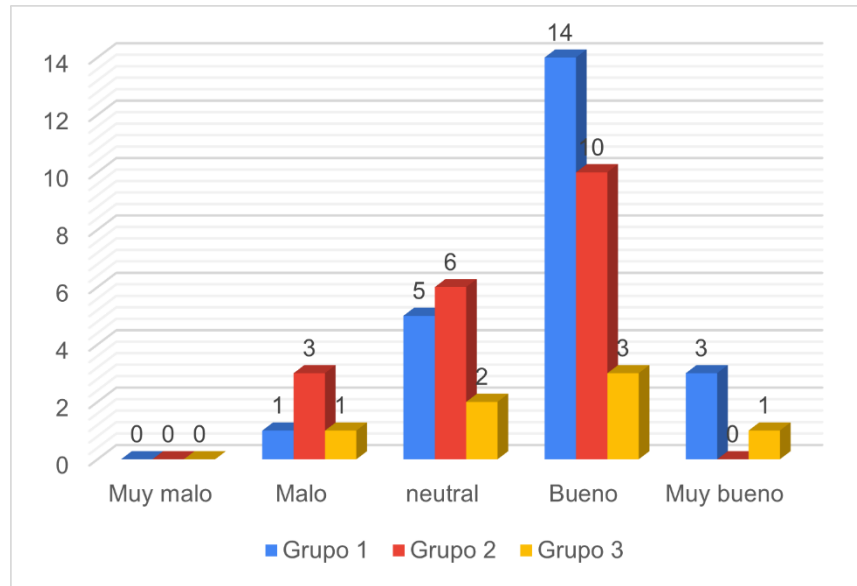
La siguiente pregunta se centró en la percepción del desempeño de los estudiantes en el desarrollo del proyecto. Solo un pequeño número indicó un desempeño negativo (10.2 %). Por el contrario, la mayoría percibió su desempeño como bueno o muy bueno (63.3 %) (figura 2). Finalmente, se les preguntó sobre su conformidad con el rol que desempeñaron en el proyecto. La mayoría afirmó haber elegido el rol con el que se sentían más cómodos (82 %). Sin embargo, el 18 % indicó que cambiaría su rol si tuvieran la oportunidad de participar en otro proyecto similar.

Figura 1. Tiempo dedicado al desarrollo del proyecto



Fuente: Elaboración propia

Figura 2. Desempeño de la participación en el desarrollo del proyecto



Fuente: Elaboración propia

La tabla 3 muestra los entregables de los grupos al finalizar el semestre, donde se consideran los rubros que se evaluaron en una medida de 0 a 100 %. Se destaca que el grupo 1 fue el que tuvo un mayor porcentaje de entrega.

Tabla 3. Estado final de los proyectos por grupo

| | Actividades | Grupo 1 | Grupo 2 | Grupo 3 |
|----------------|--|----------------|-----------------------|-----------------------|
| | Proyecto | Punto de venta | <i>Planning poker</i> | Repositorio de videos |
| Análisis | Entrevista a cliente | 100 % | 100 % | 100 % |
| | Elaboración de lista de requerimientos | 100 % | 60 % | 80 % |
| Diseño | Elaboración de casos de uso | 100 % | 100 % | 100 % |
| | Diseño de prototipo | Figma | - | Adobe XD |
| | Repositorio de documentos | Google Drive | Git Hub | One Drive |
| | Diseño de modelo entidad-relación | 100 % | 0 % | 100 % |
| | Diseño de la interfaz | 0 % | 40 % | 100 % |
| | Diseño de la estética | 0 % | 40 % | 100 % |
| | Diseño del contenido | 0 % | 40 % | 100 % |
| | Diseño arquitectónico | 100 % | 80 % | 100 % |
| Implementación | Programación de los casos de uso | 90 % | 50 % | 70 % |
| Pruebas | Pruebas del sistema | 100 % | 20 % | 0 % |
| | Revisión de diagrama entidad-relación | 100 % | 0 % | 0 % |
| | Evaluación de diseño de base de datos | 100 % | 0 % | 0 % |
| | Revisión de los casos de uso | 100 % | 0 % | 0 % |

Fuente: Elaboración propia

De acuerdo con la tabla 4, el 51 % de los estudiantes considera útil el proyecto grupal para colaborar con los miembros del equipo. En cambio, el 30.6 % no expresó una clara opinión a favor ni en contra de la idea. Finalmente, el porcentaje restante no consideró útil la actividad grupal para establecer colaboración.

Por otro lado, como se observa en la tabla 5, el 75.5 % de los estudiantes que participaron en la investigación consideran útil este tipo de metodologías aplicadas al desarrollo de proyectos para percibir la función de cada rol. En contraste, el 6.1 % no lo considera útil, mientras que el resto mantuvo una postura neutral.

En general, hubo mucha división en cuanto al deseo de desarrollar otro proyecto similar. El 46.8 % de los estudiantes está de acuerdo con esta idea; sin embargo, el 38.7 %

no estaría interesado en repetir la forma de trabajar en el proyecto. El resto de los estudiantes tiene una opinión neutral al respecto (tabla 6).

Tabla 4. El proyecto grupal me ha parecido muy útil para la colaboración entre los miembros del grupo

| | Grupo | | | Total |
|--------------------------|--------------|--------------|-------------|--------------|
| | 1 | 2 | 3 | |
| Totalmente en desacuerdo | 0 (0.0 %) | 3 (15.8 %) | 0 (0.0 %) | 3(6.1 %) |
| En desacuerdo | 1 (4.3 %) | 4 (21.1 %) | 1 (14.3 %) | 6 (12.2 %) |
| Neutral | 5 (21.7 %) | 8 (42.1 %) | 2 (28.6 %) | 15 (30.6 %) |
| De acuerdo | 7 (30.4 %) | 3 (15.8 %) | 2 (28.6 %) | 12 (24.5 %) |
| Totalmente de acuerdo | 10 (43.5 %) | 1 (5.3 %) | 2 (28.6 %) | 13 (26.5 %) |
| Total | 23 (100.0 %) | 19 (100.0 %) | 7 (100.0 %) | 49 (100.0 %) |

Fuente: Elaboración propia

Tabla 5. El proyecto grupal me ha parecido muy útil para percibir la función de cada rol en un proyecto de *software*

| | Grupo | | | Total |
|--------------------------|--------------|--------------|-------------|--------------|
| | 1 | 2 | 3 | |
| Totalmente en desacuerdo | 0 (0.0 %) | 1 (5.3 %) | 1 (14.3 %) | 2 (4.1 %) |
| En desacuerdo | 0 (0.0 %) | 1 (5.3 %) | 0 (0.0 %) | 1 (2.0 %) |
| Neutral | 3 (13.0 %) | 5 (26.3 %) | 1 (14.3 %) | 9 (18.4 %) |
| De acuerdo | 5 (21.7 %) | 8 (42.1 %) | 2 (28.6 %) | 15 (30.6 %) |
| Totalmente de acuerdo | 15 (65.2 %) | 4 (21.1 %) | 3 (42.9 %) | 22 (44.9 %) |
| Total | 23 (100.0 %) | 19 (100.0 %) | 7 (100.0 %) | 49 (100.0 %) |

Fuente: Elaboración propia

Tabla 6. Me gustaría desarrollar otro proyecto similar en otras materias

| | Grupo | | | Total |
|--------------------------|--------------|--------------|-------------|--------------|
| | 1 | 2 | 3 | |
| Totalmente en desacuerdo | 2 (8.7 %) | 8 (42.1 %) | 3 (42.9 %) | 13 (26.5 %) |
| En desacuerdo | 2 (8.7 %) | 3 (15.8 %) | 1 (14.3 %) | 6 (12.2 %) |
| Neutral | 4 (17.4 %) | 4 (21.1 %) | 2 (28.6 %) | 10 (20.4 %) |
| De acuerdo | 4 (17.4 %) | 3 (15.8 %) | 0 (0.0 %) | 7 (14.3 %) |
| Totalmente de acuerdo | 11 (47.8 %) | 1 (5.3 %) | 1 (14.3 %) | 13 (26.5 %) |
| Total | 23 (100.0 %) | 19 (100.0 %) | 7 (100.0 %) | 49 (100.0 %) |

Fuente: Elaboración propia

Aunque la mayoría de los estudiantes (42.9 %) consideran que el proyecto asignado es complejo, el grupo 2 es el que aporta el mayor porcentaje de estudiantes que consideran su proyecto complejo en cuanto a la temática (tabla 7). Además, los estudiantes fueron cuestionados sobre la complejidad del proyecto considerando los conocimientos técnicos, metodológicos y la capacidad organizativa del grupo.

En general, consideran que, en términos de sus conocimientos técnicos, el proyecto no es complejo. Sin embargo, un porcentaje alto del grupo 2 cree que sí existe complejidad técnica (tabla 8). Es decir, no poseen los conocimientos necesarios sobre el uso de lenguajes de programación, gestores de bases de datos, repositorios de código y generación de documentación, entre otros.

Tabla 7. La temática del proyecto me pareció muy compleja

| | Grupo | | | Total |
|--------------------------|--------------|--------------|-------------|--------------|
| | 1 | 2 | 3 | |
| Totalmente en desacuerdo | 1 (4.3 %) | 0 (0.0 %) | 0 (0.0 %) | 1 (2.0 %) |
| En desacuerdo | 9 (39.1 %) | 0 (0.0 %) | 2 (28.6 %) | 11 (22.4 %) |
| Neutral | 9 (39.1 %) | 5 (26.3 %) | 2 (28.6 %) | 16 (32.7 %) |
| De acuerdo | 4 (17.4 %) | 8 (42.1 %) | 0 (0.0 %) | 12 (24.5 %) |
| Totalmente de acuerdo | 0 (0.0 %) | 6 (31.6 %) | 3 (42.9 %) | 9 (18.4 %) |
| Total | 23 (100.0 %) | 19 (100.0 %) | 7 (100.0 %) | 49 (100.0 %) |

Fuente: Elaboración propia

Tabla 8. El proyecto era muy complejo para nuestros conocimientos técnicos

| | Grupo | | | Total |
|--------------------------|--------------|--------------|-------------|--------------|
| | 1 | 2 | 3 | |
| Totalmente en desacuerdo | 2 (8.7 %) | 1 (5.3 %) | 1 (14.3 %) | 4 (8.2 %) |
| En desacuerdo | 8 (34.8 %) | 6 (31.6 %) | 0 (0.0 %) | 14 (28.6 %) |
| Neutral | 12 (52.2 %) | 5 (26.3 %) | 3 (42.9 %) | 20 (40.8 %) |
| De acuerdo | 1 (4.3 %) | 4 (21.1 %) | 0 (0.0 %) | 5 (10.2 %) |
| Totalmente de acuerdo | 0 (0.0 %) | 3 (15.8 %) | 3 (42.9 %) | 6 (12.2 %) |
| Total | 23 (100.0 %) | 19 (100.0 %) | 7 (100.0 %) | 49 (100.0 %) |

Fuente: Elaboración propia

En cuanto a los conocimientos metodológicos, los grupos 2 y 3 indican que en su mayoría carecen de conocimientos sobre metodologías de desarrollo, las actividades de los roles y el desarrollo de la documentación (tabla 9). Asimismo, los tres grupos consideran que el proyecto era complejo en cuanto a la capacidad organizacional que tenían, y un alto porcentaje considera que hubo muchos problemas para organizarse en cuestiones como la división de actividades de trabajo, la comunicación entre grupos y la asistencia a reuniones (tabla 10).

El promedio de calificaciones asignadas a cada grupo está relacionado con el desempeño de los estudiantes en el proyecto. El grupo 1 obtuvo el promedio más alto con 8.9 en una escala de 0 a 10. El grupo 3 es el segundo en promedio (con 6.0) y, por último, el grupo 2 obtuvo el promedio más bajo con 6.2.

Tabla 9. El proyecto era muy complejo para nuestros conocimientos metodológicos

| | Grupo | | | Total |
|--------------------------|--------------|--------------|-------------|--------------|
| | 1 | 2 | 3 | |
| Totalmente en desacuerdo | 1 (4.3 %) | 1 (5.3 %) | 1 (14.3 %) | 3 (6.1 %) |
| En desacuerdo | 8 (34.8 %) | 3 (15.8 %) | 0 (0.0 %) | 11 (22.4 %) |
| Neutral | 12 (52.2 %) | 3 (15.8 %) | 2 (28.6 %) | 17 (34.7 %) |
| De acuerdo | 1 (4.3 %) | 6 (31.6 %) | 1 (14.3 %) | 8 (16.3 %) |
| Totalmente de acuerdo | 1 (4.3 %) | 6 (31.6 %) | 3 (42.9 %) | 10 (20.4 %) |
| Total | 23 (100.0 %) | 19 (100.0 %) | 7 (100.0 %) | 49 (100.0 %) |

Fuente: Elaboración propia

Tabla 10. El proyecto era muy complejo para la capacidad organizacional del grupo

| | Grupo | | | Total |
|--------------------------|--------------|--------------|-------------|--------------|
| | 1 | 2 | 3 | |
| Totalmente en desacuerdo | 2 (8.7 %) | 0 (0.0 %) | 0 (0.0 %) | 2 (4.1 %) |
| En desacuerdo | 6 (26.1 %) | 1 (5.3 %) | 1 (14.3 %) | 8 (16.3 %) |
| Neutral | 6 (26.1 %) | 6 (31.6 %) | 2 (28.6 %) | 14 (28.6 %) |
| De acuerdo | 8 (34.8 %) | 1 (5.3 %) | 0 (0.0 %) | 9 (18.4 %) |
| Totalmente de acuerdo | 1 (4.3 %) | 11 (57.9 %) | 4 (57.1 %) | 16 (32.7 %) |
| Total | 23 (100.0 %) | 19 (100.0 %) | 7 (100.0 %) | 49 (100.0 %) |

Fuente: Elaboración propia

La siguiente lista muestra las competencias que establece la ANIEI (2022) para un ingeniero de *software*. La numeración de estas competencias incluyendo el promedio obtenido por cada grupo está representadas en la tabla 11:

1. Comunicación oral y escrita.
2. Análisis y síntesis de información.
3. Planteamiento y resolución de problemas.
4. Modelación de soluciones.
5. Aprendizaje autónomo.
6. Trabajo en equipo.
7. Toma de decisiones.
8. Uso efectivo de herramientas de TIC.



9. Responsabilidad en la actuación.
10. Visión sobre el impacto de las soluciones.

A continuación, se enumeran las competencias específicas de un ingeniero en *software*. La tabla 12 incluye, además, el promedio obtenido por cada grupo.

1. Realiza ingeniería de requisitos de *software*.
2. Diseña *software*.
3. Construye *software*.
4. Realiza pruebas de *software*.
5. Realiza mantenimiento de *software*.
6. Administra proyectos de *software*.
7. Estima parámetros del proyecto de *software*.
8. Asegura la calidad del *software*.
9. Establece mecanismos de seguridad.
10. Emplea ciclos de vida.
11. Verifica calidad de soluciones de *software*.
12. Usa herramientas para creación de *software*.

Tabla 11. Promedio de competencias transversales o genéricas por grupo

| Grupo | Competencias | | | | | | | | | | Prom. |
|-------|--------------|------|------|------|------|------|------|------|------|------|-------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| 1 | 3.83 | 3.83 | 3.96 | 3.74 | 3.52 | 3.96 | 3.39 | 3.78 | 4.04 | 3.78 | 3.78 |
| 2 | 3.89 | 3.37 | 3.58 | 3.68 | 3.63 | 3.53 | 3.89 | 4.05 | 3.74 | 3.58 | 3.69 |
| 3 | 4.00 | 4.00 | 4.14 | 3.86 | 4.43 | 4.43 | 3.86 | 4.14 | 4.57 | 4.29 | 4.17 |

Fuente: Elaboración propia

Tabla 12. Promedio de competencias específicas de la Ing. de Software por grupo

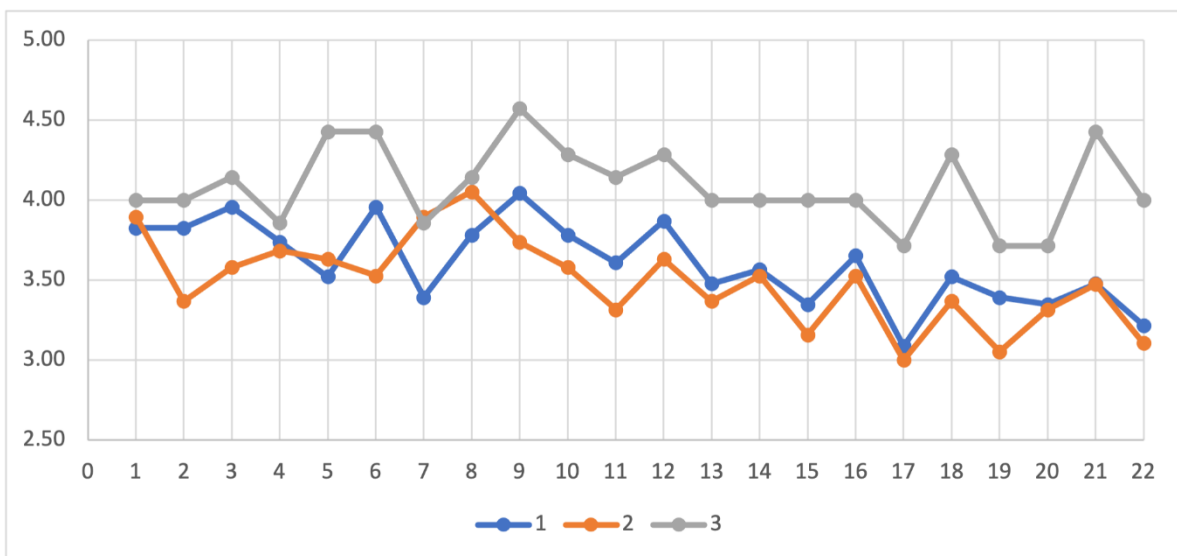
| Grupo | Competencias | | | | | | | | | | | Prom. | |
|-------|--------------|------|------|------|------|------|------|------|------|------|------|-------|------|
| | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | | 22 |
| 1 | 3.61 | 3.87 | 3.48 | 3.57 | 3.35 | 3.65 | 3.09 | 3.52 | 3.39 | 3.35 | 3.48 | 3.22 | 3.47 |
| 2 | 3.32 | 3.63 | 3.37 | 3.53 | 3.16 | 3.53 | 3.00 | 3.37 | 3.05 | 3.32 | 3.47 | 3.11 | 3.32 |
| 3 | 4.14 | 4.29 | 4.00 | 4.00 | 4.00 | 4.00 | 3.71 | 4.29 | 3.71 | 3.71 | 4.43 | 4.00 | 4.02 |

Fuente: Elaboración propia

En la tabla 11 se muestran las competencias transversales o genéricas de los grupos que participaron en esta investigación. Los datos de la tabla reflejan que mayoritariamente el grupo 2 es el que tiene los promedios más bajos, mientras que el grupo 3 tiene los más altos.

Por otro lado, la tabla 12 enseña el promedio por grupo de las competencias específicas para un ingeniero en *software*. Nuevamente, el grupo 2 tiene los promedios más bajos. De hecho, en todas las competencias de forma individual se hallan los valores más bajos de los tres grupos. Es posible apreciar cómo las competencias genéricas tienen promedios más altos que las competencias específicas. Además, de forma general, se visualizan promedios más bajos para el grupo 2.

Figura 3. Promedios de las competencias



Fuente: Elaboración propia

Análisis estadístico

El análisis de los resultados llevó a la identificación de relaciones entre variables, lo que permitió formular las siguientes hipótesis.

Hipótesis 1: Hay una diferencia significativa en el rendimiento obtenido por los grupos durante el semestre.

La variable *rendimiento* se basa en las calificaciones finales de los alumnos al concluir el semestre. En tal sentido, los grupos fueron previamente definidos como 1, 2 y 3. Para evaluar las disparidades en las medias entre los grupos, se empleó la prueba de Kruskal-Wallis debido a que los datos no presentaban una distribución normal. Según los resultados

de esta prueba estadística, se acepta la hipótesis alternativa al obtener un valor de significancia de 0.000. Con un nivel de significancia de $0.000 < 0.05$, se confirma la existencia de una diferencia significativa en las calificaciones obtenidas por los grupos 1, 2 y 3 durante el semestre.

La tabla 13 proporciona una representación de los resultados generados por el *software* estadístico SPSS. En dicha tabla, se observa que el grupo 1 exhibe una diferencia significativa en comparación con los grupos 2 y 3, pero no entre estos dos últimos (tabla 13).

Tabla 13. Comparativo entre grupos (diferencia en calificaciones)

| | |
|------------------|--------------------------------------|
| | Calificación obtenida en el proyecto |
| Kruskal-Wallis H | 25.670 |
| Df | 2 |
| Sig. | .000 |

Fuente: Elaboración propia

Para evaluar el tamaño del efecto, se aplicó la prueba *eta squared*, recomendada por Tomczak y Tomczak (2014). Los resultados obtenidos muestran un coeficiente *eta* de 0.725 y *squared* de 0.527.

Hipótesis 2: La complejidad de la temática del proyecto impacta en el rendimiento del estudiante.

Las variables analizadas fueron transformadas para reducir las categorías y evitar una tabla de contingencia extensa. La variable *complejidad de la temática* se basa en la afirmación “La temática del proyecto me pareció muy compleja”, mientras que la variable *rendimiento* se refiere a la calificación final del alumno en el proyecto al finalizar el semestre.

Inicialmente, la variable *complejidad de la temática* estaba en una escala de Likert de 5 puntos, pero se transformó a una escala de Likert de 3 puntos. Se crearon tres categorías: desacuerdo (incluyendo “totalmente en desacuerdo” y “en desacuerdo”), neutral y acuerdo (incluyendo “totalmente de acuerdo” y “de acuerdo”). La variable *rendimiento* también se redujo a tres categorías: no deseable (calificaciones de 5 y 6; categoría 1), regular (calificaciones de 7 y 8; categoría 2) y deseable (calificaciones de 9 y 10; categoría 3).

Los resultados de la prueba exacta de Fisher en la tabla 14 indican una significancia bilateral de 0.004, lo que conduce a la aceptación de la hipótesis ($0.004 < 0.05$) de que la complejidad de la temática del proyecto afecta el rendimiento del estudiante.

Tabla 14. Complejidad de la temática vs. rendimiento escolar

| | | Rendimiento | | | Total |
|----------------------------|---------------|-------------|---------|----------|-------|
| | | No deseable | Regular | Deseable | |
| Complejidad de la temática | En desacuerdo | 1 | 1 | 10 | 12 |
| | Neutral | 5 | 2 | 9 | 16 |
| | De acuerdo | 13 | 4 | 4 | 21 |
| Total | | 19 | 7 | 23 | 49 |

Fuente: Elaboración propia

Hipótesis 3: El nivel de conocimiento técnico para desarrollar el proyecto afecta el rendimiento del estudiante.

La variable *conocimiento técnico* se obtiene a partir de la respuesta Likert a la afirmación “El proyecto era muy complejo para nuestros conocimientos técnicos”. Esta variable evalúa si el estudiante posee el conocimiento técnico necesario para llevar a cabo el proyecto. Al igual que las variables anteriores, se dividió en 3 categorías: en desacuerdo, neutral y de acuerdo.

Según los resultados de la tabla 15, la prueba exacta de Fisher muestra una significancia bilateral de 0.001. Dado que este valor cumple con la condición $0.001 < 0.05$, se concluye que existe una relación significativa entre el conocimiento técnico necesario para desarrollar el proyecto y el rendimiento del estudiante. Por lo tanto, se acepta la hipótesis que afirma que “el conocimiento técnico para desarrollar el proyecto influye en el rendimiento del estudiante”.

Tabla 15. Conocimiento técnico vs. rendimiento escolar

| | | Rendimiento | | | Total |
|----------------------|---------------|-------------|---------|----------|-------|
| | | No deseable | Regular | Deseable | |
| Conocimiento técnico | En desacuerdo | 7 | 1 | 10 | 18 |
| | Neutral | 4 | 3 | 13 | 20 |
| | De acuerdo | 8 | 3 | 0 | 11 |
| Total | | 19 | 7 | 23 | 49 |

Fuente: Elaboración propia

Hipótesis 4: El nivel de conocimiento metodológico para desarrollar el proyecto afecta el rendimiento del alumno.

La variable *conocimiento metodológico* se basa en la respuesta Likert a la afirmación “El proyecto era muy complejo para nuestros conocimientos metodológicos”. Esta variable evalúa si el estudiante posee el conocimiento metodológico necesario para llevar a cabo el

proyecto. Al igual que en las hipótesis anteriores, se dividió en 3 categorías: en desacuerdo, neutral y de acuerdo.

Según los resultados de la tabla 16, la prueba exacta de Fisher mostró una significancia bilateral de 0.001. Dado que este valor cumple con la condición $0.001 < 0.05$, se concluye que existe una relación significativa entre el conocimiento metodológico necesario para desarrollar el proyecto y el rendimiento del alumno. Por lo tanto, se acepta la hipótesis que afirma que “el conocimiento metodológico para desarrollar el proyecto influye en el rendimiento del alumno”.

Tabla 16. Conocimiento metodológico contra rendimiento escolar

| | | Rendimiento | | | Total |
|---------------------------|---------------|-------------|---------|----------|-------|
| | | No deseable | Regular | Deseable | |
| Conocimiento metodológico | En desacuerdo | 5 | 1 | 8 | 14 |
| | Neutral | 3 | 1 | 13 | 17 |
| | De acuerdo | 11 | 5 | 2 | 18 |
| Total | | 19 | 7 | 23 | 49 |

Fuente: Elaboración propia

Hipótesis 5: La capacidad organizacional del grupo para desarrollar proyectos impacta en el rendimiento académico del alumno.

La variable *capacidad organizacional* se define a partir de la respuesta en la escala Likert a la afirmación “El proyecto resultó ser muy complejo en términos de organización para el grupo”. Esta variable también fue categorizada en 3 niveles, como se muestra en la tabla 17.

Sin embargo, los resultados de la prueba exacta de Fisher revelan una significancia bilateral de 0.153. Dado que este valor no cumple con el criterio de significancia establecido ($0.153 > 0.05$), no podemos aceptar la hipótesis que sugiere que “la capacidad organizacional del grupo para desarrollar proyectos influye en el rendimiento académico del alumno”.

Tabla 17. Capacidades organizacionales contra rendimiento escolar

| | | Rendimiento | | | Total |
|--------------------------|---------------|-------------|---------|----------|-------|
| | | No deseable | Regular | Deseable | |
| Capacidad organizacional | En desacuerdo | 1 | 3 | 6 | 10 |
| | Neutral | 5 | 2 | 7 | 14 |
| | De acuerdo | 13 | 2 | 10 | 25 |
| Total | | 19 | 7 | 23 | 49 |

Fuente: Elaboración propia

Discusión

La experiencia de los estudiantes en proyectos grupales fue innovadora, ya que introdujo formas de organización y trabajo diferentes, lo cual suele motivar a los estudiantes y, por tanto, se traduce en mejoras en su desempeño académico. Aun así, también se debe indicar que la estrategia de trabajo puede influir en los estudiantes de diversas maneras, pues —según se observa en el estado final de los proyectos (tabla 3)— el grupo 1 tuvo el mejor rendimiento, mientras que el grupo 2 presentó el peor desempeño. Este último grupo no entregó muchas de las actividades y otras fueron cumplidas de manera incompleta, lo que resultó en calificaciones muy bajas. Además, las actividades enumeradas en la tabla 3 presentan diferentes niveles de complejidad, por lo que calcular un promedio de las calificaciones no sería justo. Una de las actividades principales y más significativas fue la codificación (implementación), donde se refleja el progreso del proyecto de manera más clara, permitiendo apreciar hasta qué punto cada grupo logró alcanzar sus objetivos.

Análisis de los grupos

El grupo 1 demostró el mejor rendimiento en comparación con los otros. La mayoría de sus miembros consideraron que los tiempos asignados al proyecto fueron adecuados, lo que se reflejó en su desempeño muy acorde a las calificaciones finales. De hecho, el 73.9 % de los estudiantes evaluaron su desempeño entre bueno y muy bueno. Aunque seis estudiantes expresaron su disposición a cambiar de rol de ser necesario, lo que representa un porcentaje bajo (26.1 %), esto sugiere que en general el grupo tuvo un buen desempeño, a pesar de que algunos alumnos no estaban completamente satisfechos con sus roles asignados.

Asimismo, el grupo mostró aspectos positivos en cuanto a sus conocimientos, ya que no consideraron el proyecto complejo en términos técnicos y metodológicos, y tampoco encontraron la temática complicada. Aunque evaluaron su capacidad de organización como regular, demostraron buenas actitudes en general durante el desarrollo del proyecto.

Considerando el promedio de sus calificaciones (8.9), los estudiantes del grupo 1 manifestaron una disposición positiva hacia la idea de desarrollar otro proyecto de la misma forma, ya que no consideraron esta forma de trabajar como compleja. A pesar de que el grupo tuvo el mejor desempeño, sus competencias no se destacaron especialmente; de hecho, en cuatro de las competencias genéricas obtuvieron los valores más bajos de los tres grupos. En cuanto a las competencias específicas, no consiguieron los valores más bajos, pero tampoco los más altos.

Por otra parte, el grupo 2 registró el menor rendimiento en los proyectos. El 63.2 % indicó que el tiempo asignado era adecuado, lo que, aunque no constituye una mayoría abrumadora, sugiere una tendencia hacia la conformidad con los plazos establecidos. Solo tres casos (15.8 %) expresaron un desempeño bajo, mientras que el 52.5 % consideró que su desempeño en el proyecto fue bueno. Además, solo tres alumnos (15.8 %) manifestaron su disposición a cambiar de rol en caso de realizar otro proyecto, lo que implica que la mayoría estaba conforme con el rol que desempeñaron.

El grupo reconoció no poseer los conocimientos suficientes para abordar el proyecto. Aunque hubo una distribución equilibrada entre opiniones negativas y positivas sobre los conocimientos técnicos necesarios, no se observó una mayoría que considerara tener el conocimiento necesario. Además, un alto porcentaje indicó carecer de conocimientos técnicos. La capacidad organizacional del grupo para el proyecto fue evaluada como muy baja en su mayoría, lo que se reflejó en la percepción de que la temática del proyecto era compleja. Estos aspectos impactaron negativamente en su desempeño, ya que la baja capacidad para desarrollar los casos de uso afectó su calificación final en la materia, con un promedio de desempeño de 6.2 entre los estudiantes.

Esta experiencia no fue satisfactoria para ellos, ya que expresaron su desagrado y señalaron que no les gustaría realizar otro proyecto de este tipo en el futuro. En cuanto a las competencias, el grupo 2 mostró un nivel más bajo que los otros. En las competencias genéricas, obtuvieron el promedio más bajo en tres de las diez competencias, y en las competencias específicas, tuvieron los promedios más bajos en cada una de las once competencias. Aunque los alumnos tienden a autoevaluarse de forma más generosa, en promedio fueron evaluados con valores más bajos que el resto.

Por último, el grupo 3 demostró un rendimiento regular o inferior en el proyecto. A pesar de que el grupo consideró que el tiempo asignado para el desarrollo del proyecto era adecuado y que su desempeño en el proyecto fue bueno, los resultados obtenidos no respaldan esta percepción. Además, todos los estudiantes confirmaron que el rol que eligieron era el

más adecuado para ellos. Según la opinión del grupo, el proyecto permitió la colaboración entre los miembros del equipo. Sin embargo, consideraron que carecían de los conocimientos técnicos, metodológicos y de la capacidad organizativa necesarios, lo que llevó a que no estuvieran conformes con la idea de repetir un proyecto similar. Si bien lograron un desempeño regular en la implementación de los casos de uso con el 70 %, no pudieron realizar pruebas, lo que determinó que la calificación promedio del grupo fuera de 6.9.

En definitiva, los resultados de las competencias son contradictorios, pues contradicen las afirmaciones sobre los conocimientos teóricos, metodológicos y organizativos del grupo. A pesar de esto, los promedios indican que el grupo 3 obtuvo los puntajes más altos en las competencias genéricas y específicas en comparación con los otros dos grupos. Sin embargo, estos resultados no concuerdan ni con los resultados del proyecto ni con los conocimientos que dicen poseer.

Competencias

Las competencias genéricas con los valores más bajos para el grupo con el peor rendimiento son el análisis y síntesis de información, y el trabajo en equipo. Estos resultados coinciden con el desempeño del grupo al obtener los requisitos del proyecto, los cuales estaban incompletos y, en algunos casos, inconsistentes. Además, el grupo mostró una notable falta de coordinación, comunicación y cooperación entre los miembros, lo que demuestra una deficiencia en el trabajo en equipo. Estos hallazgos concuerdan con lo mencionado por Don y Raman (2019), quienes destacan la importancia del trabajo en equipo para alcanzar objetivos y aumentar la productividad, además de fomentar el compañerismo. Sin embargo, estos datos contrastan con los mencionados por Souza *et al.* (2019), quienes aplicaron ABP en un curso de ingeniería de *software* y encontraron que esta estrategia benefició el aprendizaje de los requisitos de *software*.

En cuanto a las competencias específicas del ingeniero de *software*, el grupo con el rendimiento más bajo mostró una baja puntuación en la estimación de parámetros del proyecto de *software* y en el establecimiento de mecanismos de seguridad para el *software*. La primera competencia se centra en estimaciones como costos, tiempo, personal y documentación, entre otros, aspectos que fueron poco empleados en el proyecto, dado que los plazos estaban fijados y no era necesario calcular un costo para el proyecto; además, la documentación ya estaba definida. Aun así, el grupo indicó que necesita mejorar en esta competencia. Estos resultados son similares a los encontrados por Souza *et al.* (2019),

quienes determinaron que la configuración del *software* fue el aspecto que generó un menor aprovechamiento en los alumnos al aplicar ABP.

En cuanto a la segunda competencia —seguridad del *software*, que se refiere al desarrollo de aspectos que eviten vulnerabilidades del *software*—, aunque no se solicitó específicamente en el proyecto, se requería cumplir con ciertos estándares mínimos de seguridad en el acceso de los usuarios al sistema. A pesar de ello, el grupo 2 indicó tener un bajo nivel en competencias para aspectos de seguridad más avanzados.

Hipótesis

Las calificaciones de los grupos 1, 2 y 3 fueron 8.9, 6.2 y 6.9, respectivamente. Según la prueba de hipótesis Kruskal-Wallis (ver sección *Análisis estadísticos*), existe una diferencia significativa en los promedios de calificaciones entre los grupos. De acuerdo con los rangos de interpretación propuestos por Cohen (1988), el tamaño del efecto es grande ($\eta = 0.725$, $\eta^2 = 0.527$), lo que indica que la diferencia de medias es notable al menos en algunos grupos. El grupo 1 obtuvo la mejor calificación y demostró ser más unido, con niveles de conocimiento similares, por lo que se destaca de los otros dos grupos.

Los grupos 2 y 3 tienen características similares, pero el grupo 3 muestra un mejor rendimiento, probablemente debido a su menor número de alumnos, lo que les permitió una mejor organización y enfrentamiento de los problemas surgidos durante el semestre, con lo cual superan al grupo 2. Aunque los proyectos eran diferentes, los grupos exhiben características distintas según las competencias que indican poseer.

De acuerdo con la prueba exacta de Fisher, la complejidad de la temática del proyecto influye en el rendimiento. Según la opinión de los estudiantes, el grupo 2 consideraba el proyecto complejo, mientras que el grupo 3 lo consideraba en menor medida. Sin embargo, debido al mayor número de alumnos en el grupo 2, la prueba de hipótesis resultó positiva, además del contraste con el grupo 1, que no consideraba complejo el proyecto. Por lo tanto, se concluye que la complejidad del proyecto impacta en la calificación del estudiante.

Esto concuerda con las investigaciones de Wang *et al.* (2022) y Luo *et al.* (2017), quienes indican que la complejidad de los proyectos tiene un impacto negativo en su desarrollo, por lo que se requiere un mecanismo para reducir el riesgo. Por tanto, si al principio el proyecto es considerado complejo por los estudiantes, será necesario modularizar el proyecto e iniciar el desarrollo en iteraciones o utilizar otros mecanismos para reducir la complejidad.

En tal sentido, la prueba estadística indica que el conocimiento técnico para desarrollar un proyecto de *software* influye en el rendimiento académico del estudiante. Esto tiene sentido, ya que el conocimiento técnico representa uno de los aspectos más importantes para llevar a cabo el proyecto, dado que implica el manejo de herramientas básicas como el lenguaje de programación y los gestores de bases de datos; sin estos, es imposible que el proyecto termine adecuadamente. Lo extraño es que los estudiantes señalan en un porcentaje alto no tener los conocimientos técnicos para afrontar este proyecto, cuando para el semestre en cuestión ya deberían cubrir estas necesidades. Esto ocasiona que la prueba estadística sea positiva.

A pesar de la importancia del conocimiento técnico, Sedelmaier y Landes (2014) sostienen que no solo este aspecto es importante, sino que también debe combinarse con habilidades blandas para tener un conocimiento integral. Los resultados de esta investigación coinciden con los de Ceh-Varela *et al.* (2023), quienes afirman que el ABP les permite aumentar sus habilidades técnicas, necesarias en trabajos de la vida real relacionados con el desarrollo de *software*. A su vez, Adorjan y Solari (2021) explican que, aunque las técnicas fundamentales de la ingeniería de *software* son estables, el cambio tecnológico impacta continuamente en la plataforma de desarrollo, ejecución de *software* y en las herramientas de desarrollo.

De acuerdo con la prueba de hipótesis, se confirma que el conocimiento metodológico está relacionado con el rendimiento del estudiante. El conocimiento metodológico es fundamental para el desarrollo del *software*, ya que dicta la organización necesaria para completar la tarea. No obstante, se considera menos significativo que el conocimiento técnico previamente mencionado, ya que la organización puede ser abordada mediante metodologías diferentes a las específicas del *software*, siempre y cuando exista orden en el trabajo y en la asignación de tareas. Esto permite que los estudiantes puedan cubrir las necesidades con conocimientos previos que posean, aunque algunos alumnos no pudieron encontrar una forma adecuada de organización, lo que llevó a que consideraran no tener los conocimientos metodológicos necesarios para el proyecto. Esto se refleja en que aquellos alumnos que afirmaban tener los conocimientos metodológicos también obtuvieron calificaciones altas, mientras que aquellos que afirmaban no tenerlos obtuvieron calificaciones bajas.

En síntesis, el desarrollo de *software* no es una tarea sencilla, por lo que existen numerosas metodologías de desarrollo, algunas de las cuales tienen aciertos y otras errores, por lo que en ocasiones la intuición y el entusiasmo del desarrollador por hacer bien su trabajo

resultan valiosos. Aun así, contar con guías es bastante útil, especialmente para desarrolladores novatos (O'Regan, 2022).

De acuerdo con los resultados, la hipótesis “La capacidad organizacional del grupo para desarrollar proyectos influye en el rendimiento escolar del alumno” resultó negativa, pues hubo discrepancia entre los estudiantes en cuanto a si su capacidad organizativa era suficiente para abordar las necesidades de desarrollo del proyecto. Algunas personas que estaban de acuerdo con la afirmación obtuvieron buenas calificaciones a pesar de ello, mientras que otras que no estaban de acuerdo también obtuvieron buenas calificaciones, lo que indica la falta de una tendencia clara para que la hipótesis resultara positiva.

Finalmente, si bien determinadas tareas en el desarrollo de *software* pueden ejecutarse en paralelo, otras deben comenzar cuando finaliza la tarea anterior, lo cual demanda una capacidad organizativa excepcional para generar el mejor *software* con un costo mínimo (Alsaqqa *et al.*, 2020). Por eso, Ceh-Varela *et al.* (2023) concluyen que la comunicación entre el líder y los miembros del equipo es de vital importancia para identificar los requerimientos de la aplicación de manera efectiva. En tal sentido, al utilizar un enfoque ABP, los estudiantes mejoraron su capacidad de autonomía, confianza en sí mismos, trabajo en equipo y habilidades de aprendizaje.

Conclusiones

El aprendizaje basado en proyectos (ABP) es un enfoque pedagógico en el que los estudiantes adquieren conocimientos y habilidades a través de la realización de proyectos concretos y significativos. En lugar de enseñar conceptos de forma aislada, el ABP involucra a los estudiantes en la resolución de problemas o la realización de tareas que les permiten aplicar y consolidar su aprendizaje de manera más efectiva. El objetivo principal del ABP es ayudar a los estudiantes a desarrollar habilidades de pensamiento crítico, resolución de problemas, colaboración y comunicación, al mismo tiempo que fomenta su motivación y entusiasmo por el aprendizaje.

En esta investigación, se aplicó el ABP en el contexto de la ingeniería de *software*, específicamente en un proyecto grupal donde los estudiantes fueron organizados en equipos. En estos comúnmente se trabajan en equipos de tres a cuatro integrantes con el fin de distribuir las tareas y asumir diversos roles para desarrollar el producto. Sin embargo, en esta investigación se propuso organizar a los estudiantes en equipos de trabajo enfocados en roles

específicos de la ingeniería de *software* y luego se asignó a cada grupo una tarea específica dentro del proceso.

Ahora bien, en cuanto a la pregunta de investigación formulada al inicio de esta investigación (¿cuáles son las principales competencias que se pueden destacar en la práctica del aprendizaje grupal basado en proyectos para el desarrollo de *software*?), se puede indicar lo siguiente. En primer lugar, y desde el punto de vista teórico, se puede afirmar que los proyectos de desarrollo de *software* utilizando el ABP permite a los estudiantes aprender habilidades fundamentales para el trabajo en equipo, comunicación efectiva y resolución de problemas, las cuales son altamente valoradas en el ámbito laboral, ya que fomentan la colaboración para alcanzar objetivos comunes. Además, al trabajar en proyectos concretos, los estudiantes tienen la oportunidad de aplicar y consolidar sus conocimientos teóricos de manera práctica y contextualizada, lo que mejora significativamente su comprensión y retención de conceptos.

En segundo lugar, y desde el punto de vista práctico, esta estrategia permite a los estudiantes desarrollar habilidades técnicas prácticas, como la programación, la resolución de problemas y el uso de herramientas y tecnologías específicas. Esta experiencia les brinda la oportunidad de adquirir competencias prácticas directamente relacionadas con la industria del *software*, con lo cual se preparan para enfrentar desafíos reales en su futuro profesional. Además, este enfoque hace que el aprendizaje sea más dinámico y atractivo, de modo que puede resultar más interesante e impactar de forma positiva en su rendimiento académico.

Por otra parte, y de manera más específica, la mencionada pregunta puede responderse desde varios ángulos. En tal sentido, se puede señalar que, según el análisis de competencias generales, 1) este tipo de proyecto no debe ser asignado a grupos de trabajo con un bajo nivel de análisis y síntesis de información, y 2) es recomendable que el grupo pueda trabajar en equipo de manera efectiva, dado que este aspecto es vital para la coordinación y organización del proyecto.

Considerando las competencias específicas, podemos indicar que, aunque los grupos no mencionaron directamente problemas en las competencias específicas más utilizadas, se pudo identificar a través de la observación y revisiones la dificultad del grupo 2 para llevar a cabo la ingeniería de requisitos de *software*. Además, esta competencia guarda relación con la competencia general de análisis y síntesis de información.

Por último, respecto al análisis de los resultados de las hipótesis, se puede concluir lo siguiente:

1. Se observaron diferencias en los niveles de conocimiento entre los grupos de estudio, pues el grupo 1 alcanzó niveles más altos, mientras que los grupos 2 y 3 mostraron un equilibrio hacia niveles más bajos.
2. La complejidad de la materia se identificó como un factor determinante para la obtención de buenos resultados, lo que sugiere que los temas por desarrollar deben ser de fácil comprensión para los estudiantes.
3. Se evidenció que este tipo de proyectos funciona mejor cuando los estudiantes poseen conocimientos técnicos adecuados, lo que resalta la importancia de evaluar este ítem antes de aplicar el ABP en grupo.
4. Es crucial contar con conocimientos metodológicos del desarrollo de *software* para llevar a cabo adecuadamente el proyecto.

Trabajo futuro

Esta investigación ha identificado varios aspectos importantes en el tema de estudio, pero también ha generado áreas de investigación futuras. Por eso, se propone, en primer lugar, la aplicación de una segunda prueba para contrastar los resultados obtenidos en este estudio y alcanzar conclusiones más sólidas, lo cual permitiría evitar posibles sesgos que pudieron surgir en la primera investigación.

En segundo lugar, se sugiere replicar el mismo proyecto con otro grupo de estudiantes para comparar su comportamiento y los resultados del proyecto, lo cual ayudaría a confirmar las habilidades que son necesarias para que los grupos utilicen eficazmente el aprendizaje basado en proyectos.

Finalmente, como tercera línea de investigación, se plantea la posibilidad de desarrollar una serie de directrices para la implementación del aprendizaje basado en proyectos en diferentes grupos, y luego probar la eficacia de estas directrices en otro caso práctico.

Referencias

- Abella García, V., Ausín Villaverde, V., Delgado Benito, V. y Casado Muñoz, R. (2020). Aprendizaje basado en proyectos y estrategias de evaluación formativas: percepción de los estudiantes universitarios. *Revista Iberoamericana de Evaluación Educativa*, 13(1), 93. <https://doi.org/10.15366/riece2020.13.1.004>
- Adorjan, A. and Solari, M. (2021). Software Engineering Project-Based Learning in an Up-To-Date Technological Context. *2021 IEEE URUCON, URUCON 2021*, 486–491. <https://doi.org/10.1109/URUCON53396.2021.9647348>
- Al Mulhim, E. N. and Eldokhny, A. A. (2020). The Impact of Collaborative Group Size on Students' Achievement and Product Quality in Project-Based Learning Environments. *International Journal of Emerging Technologies in Learning (IJET)*, 15(10), 157–174. <https://doi.org/10.3991/ijet.v15i10.12913>
- Alsaqqa, S., Sawalha, S. and Abdel-Nabi, H. (2020). Agile Software Development: Methodologies and Trends. *International Journal of Interactive Mobile Technologies (IJIM)*, 14(11), 246–270. <https://doi.org/10.3991/ijim.v14i11.13269>
- Asociación Nacional de Instituciones de Educación en Tecnologías de Información (ANIEI) (2022). *Asociación Nacional de Instituciones de Educación en Tecnologías de Información*. <http://www.aniei.org.mx/ANIEI/>
- Ceh-Varela, E., Canto-Bonilla, C. and Duni, D. (2023). Application of Project-Based Learning to a Software Engineering course in a hybrid class environment. *Information and Software Technology*, 158. <https://doi.org/10.1016/j.infsof.2023.107189>
- Chimbo Jumbo, J. J. y Larreal Bracho, A.J. (2023). Metodologías educativas para el desarrollo de competencias científicas. *Ciencia Latina Revista Científica Multidisciplinar*, 7(1), 7021-7048. https://doi.org/10.37811/cl_rcm.v7i1.4942
- Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences* (2nd ed.). Lawrence Erlbaum Associates.
- Consejo Nacional de Acreditación en Informática y Computación (CONAIC) (6 de febrero de 2023). *Consejo Nacional de Acreditación en Informática y Computación*. <https://www.conaic.net/>
- Cyrulies, E. y Shamne, M. (2021). El aprendizaje basado en proyectos: una capacitación docente vinculante. *Páginas de Educación*, 14(1), 1-25. <https://doi.org/10.22235/pe.v14i1.2293>

- Da Cunha, P. R. (2005). Teaching Software Engineering using Project-Based Learning. *Exploring Innovation in Education and Research*, 1–4. <https://www.researchgate.net/publication/215575463>
- Flores-Ruiz, E., Miranda-Navales, M. G. y Villasís-Keever, M. Á. (2017). El protocolo de investigación VI: cómo elegir la prueba estadística adecuada. *Estadística inferencial. Revista Alergia México*, 64(3), 364. <https://doi.org/10.29262/ram.v64i3.304>
- Gómez Álvarez, M. C., Manrique-Losada, B. y Gasca-Hurtado, G. P. (2015). Propuesta de evaluación de habilidades blandas en ingeniería de software por medio de proyectos universidad-empresa. *Revista Educación en Ingeniería*, 10(19), 131–140.
- Hernández-Sampieri, R. y Mendoza, C (2018). *Metodología de la investigación. Las rutas cuantitativa, cualitativa y mixta*. Editorial Mc Graw Hill Education
- Luo, L., He, Q., Jaselskis, E. J. and Xie, J. (2017). Construction Project Complexity: Research Trends and Implications. *Journal of Construction Engineering and Management*, 143(7), 4017019. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0001306](https://doi.org/10.1061/(ASCE)CO.1943-7862.0001306)
- Macías, J. A. (2012). Enhancing project-based learning in software engineering lab teaching through an e-portfolio approach. *IEEE Transactions on Education*, 55(4), 502–507. <https://doi.org/10.1109/TE.2012.2191787>
- Marnewick, C. (2023). Student experiences of project-based learning in agile project management education. *Project Leadership and Society*, 4, 1-10. <https://doi.org/10.1016/j.plas.2023.100096>
- Morais, P., Ferreira, M. y Veloso, B. (2021). Improving Student Engagement With Project-Based Learning: A Case Study in Software Engineering. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, 16(1), 21-28. <https://doi.org/10.1109/RITA.2021.3052677>
- Nurbekova, Z., Grinshkun, V., Aimicheva, G., Nurbekov, B. and Tuenbaeva, K. (2020). Project-Based Learning Approach for Teaching Mobile Application Development Using Visualization Technology. *International Journal of Emerging Technologies in Learning (iJET)*, 15(8), 130-143. <https://www.learntechlib.org/p/217072/>.
- O'Regan, G. (2022). *Concise Guide to Software Engineering. From Fundamentals to Application Methods* (2nd ed.) Springer.
- Pérez, B. and Rubio, a. L. (2020). *A Project-Based Learning Approach for Enhancing Learning Skills and Motivation in Software Engineering*. 51st ACM Technical

- Symposium on Computer Science Education (SIGCSE '20), 309–315.
<https://doi.org/10.1145/3328778.3366891>
- Sánchez, P. y Blanco, C. (2012). *Implantación de una metodología de aprendizaje basada en proyectos para una asignatura de Ingeniería del Software*. XVIII Jornadas de Enseñanza Universitaria de la Informática, 41–48. <http://www.comunio.es/>
- Sedelmaier, Y. and Landes, D. (2014). *Software engineering body of skills (SWEBOS)*. 2014 IEEE Global Engineering Education Conference (EDUCON), 395–401.
- Sotomayor, C., Vaccaro, C. y Téllez, A. (2021). *Aprendizaje basado en proyectos: un enfoque pedagógico para potenciar los procesos de aprendizaje hoy*. Fundación Chile. Centro de Innovación del Ministerio de Educación y el financiamiento de la Embajada de Estados Unidos en Chile.
- Souza, M., Moreira, R. and Figueiredo, E. (2019). *Students perception on the use of project-based learning in software engineering education*. SBES '19: Proceedings of the XXXIII Brazilian Symposium on Software Engineering, 537–546.
<https://doi.org/10.1145/3350768.3352457>
- Wang, T., Chan, A., He, Q. and Xu, J. (2022). Identifying the gaps in construction megaproject management research: a bibliographic analysis. *International Journal of Construction Management*, 22(9), 1585–1596.
<https://doi.org/10.1080/15623599.2020.1735610>
- Tomczak, M. and Tomczak, E. (2014). The need to report effect size estimates revisited. An overview of some recommended measures of effect size. *TRENDS in Sport Sciences*, 1(21), 19–25.
- Villalobos-Abarca, M. A., Herrera-Acuña, R. A., Ramírez, I. G. and Cruz, X. C. (2018). Real project-based learning applied to Software Engineers' education. *Formación Universitaria*, 11(3), 97–112. <https://doi.org/10.4067/S0718-50062018000300097>
- Zambrano Briones, M. A., Hernández Díaz, A. y Mendoza Bravo, K. L. (2022). El aprendizaje basado en proyectos como estrategia didáctica. *Conrado*, 18(84), 172-182.

| Rol de Contribución | Autor (es) |
|---|---|
| Conceptualización | Alan Ramírez-Noriega |
| Metodología | José Mendivil-Torres, Samantha Jiménez (igual) |
| Software | No aplica |
| Validación | Carolina Tripp-Barba, José Alfonso Aguilar-Calderón (igual) |
| Análisis Formal | Carolina Tripp-Barba, José Alfonso Aguilar-Calderón (igual) |
| Investigación | José Alfonso Aguilar-Calderón |
| Recursos | José Mendivil-Torres |
| Curación de datos | José Mendivil-Torres |
| Escritura - Preparación del borrador original | Alan Ramírez-Noriega, Carolina Tripp-Barba (igual) |
| Escritura - Revisión y edición | Samantha Jiménez, Carolina Tripp-Barba (igual) |
| Visualización | Alan Ramírez-Noriega, José Mendivil-Torres (igual) |
| Supervisión | Alan Ramírez-Noriega |
| Administración de Proyectos | Alan Ramírez-Noriega |
| Adquisición de fondos | José Mendivil-Torres, Alan Ramírez-Noriega, Carolina Tripp-Barba, José Alfonso Aguilar-Calderón, Samantha Jiménez (igual) |