ISBN: 978-628-96613-1-6. ISSN: 2414-6390. Digital Object Identifier: https://dx.doi.org/10.18687/LACCEI2025.1.1.1981

Project-Based Software Education for Enterprises: Challenges, Lessons Learned and Best Practices

Freddy Gonzales-Saji, Msc[1], Lady Concha-Diaz[1], Lisbeth Ortiz-Huarachi[1], Edwar Saire-Peralta[1], Giovanni Cabrera-Málaga[1], Rene Nieto-Valencia[1]

[1]*Universidad Nacional de San Agustín de Arequipa, Perú*

fgonzaless@unsa.edu.pe, lconcha@unsa.edu.pe, lortizh@unsa.edu.pe, esaire@unsa.edu.pe, gcabrerama@unsa.edu.pe, rnietov@unsa.edu.pe

Abstract– ***This paper outlines the process by which the Professional School of Systems Engineering at UNSA implemented the Software Engineering Project course, in which students developed technological solutions for entities with specific needs. A qualitative and descriptive research methodology was used, based on project-based learning (PBL), combined with agile approaches, with the objective of orchestrating the development of software. Students worked in teams according to the SCRUM methodology and documented project development from initial conception to completion.***

Three projects were analyzed: an ERP system for a school, a research laboratory management platform and a sports tournament management software. The evaluation of the final products showed a positive impact on the beneficiary institutions, as evidenced by the use of the systems in production and customer recognition. Obstacles such as insufficient experience in agile methodologies and time management were detected, suggesting improvements in student education. Additionally, the relevance of documentation and the implementation of collaborative tools to ensure the long-term viability of the projects is highlighted.

The results obtained show that the combination of PBL and SCRUM is an effective strategy for training in software engineering, strengthening the link with the productive sector and promoting the acquisition of key competencies for professional development.

Keywords-- ***Project-based learning, agile methodologies, SCRUM, software engineering, software development, engineering education***

Enseñanza Basada en Proyectos de Software para Empresas: Retos, Lecciones Aprendidas y Mejores Prácticas

Freddy Gonzales-Saji, Msc[1], Lady Concha-Diaz[1], Lisbeth Ortiz-Huarachi[1], Edwar Saire-Peralta[1], Giovanni Cabrera-Málaga[1], Rene Nieto-Valencia[1]

[1]*Universidad Nacional de San Agustín de Arequipa, Perú*

fgonzaless@unsa.edu.pe, lconcha@unsa.edu.pe, lortizh@unsa.edu.pe, esaire@unsa.edu.pe, gcabrerama@unsa.edu.pe, rnietov@unsa.edu.pe

Resumen*– Este documento expone el proceso mediante el cual la Escuela Profesional de Ingeniería de Sistemas de la UNSA implementó el curso Proyecto de Ingeniería de Software, en el que los estudiantes desarrollaron soluciones tecnológicas para entidades que presentan necesidades específicas. Se recurrió a una metodología de investigación cualitativa y descriptiva, basada en el aprendizaje basado en proyectos (ABP), combinado con enfoques ágiles, con el objetivo de orquestar la elaboración de programas informáticos. Los estudiantes trabajaron en equipos conforme a la metodología SCRUM y documentaron el desarrollo del proyecto desde su concepción inicial hasta su culminación..Se analizaron tres proyectos: un sistema ERP para un colegio, una plataforma de gestión de laboratorios de investigación y un software de administración de torneos deportivos. La evaluación de los productos finales reflejó un impacto positivo en las instituciones beneficiarias, evidenciado por el uso de los sistemas en producción y el reconocimiento de los clientes. Se detectaron obstáculos tales como la insuficiencia de experiencia en metodologías ágiles y la administración del tiempo, lo que sugiere mejoras en la educación estudiantil. Adicionalmente, se subraya la relevancia de la documentación y la implementación de herramientas colaborativas para asegurar la viabilidad a largo plazo de los proyectos, los resultados obtenidos demuestran que la combinación de PBL y SCRUM es una estrategia efectiva para la formación en ingeniería de software, fortaleciendo la vinculación con el sector productivo y promoviendo la adquisición de competencias clave para el desarrollo profesional.

***Palabras clave*-- Aprendizaje basado en proyectos, metodologías ágiles, SCRUM, ingeniería de software, desarrollo de software, educación en ingeniería**

I. Introducción

La Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional de San Agustín de Arequipa(UNSA) ha obtenido la acreditación de la Comisión de Acreditación para Ingeniería y Tecnología(ABET), lo cual asegura que su programa satisface los estándares de calidad más elevados en la educación en ingeniería [1].

Como parte de su plan de estudios, la escuela ofrece el curso "Proyecto de Ingeniería de Software", diseñado como una asignatura tipo Capstone. Este curso, impartido en el último año de la carrera, tiene como objetivo integrar y aplicar los conocimientos adquiridos por los estudiantes a lo largo de su formación en proyectos reales, evaluando de manera integral sus competencias y preparación profesional [2].

La implementación de cursos Capstone en programas de ingeniería es reconocida por su eficacia en el desarrollo de habilidades técnicas y blandas en los estudiantes, promoviendo la resolución de problemas complejos y el trabajo en equipo [3]. En el contexto de la UNSA, la acreditación ABET refuerza el compromiso de la institución con estándares internacionales de calidad educativa, asegurando que los egresados estén preparados para contribuir significativamente en el campo de la ingeniería de software [4].

Diversas iniciativas han demostrado la importancia de estos cursos en la formación de los estudiantes. Un ejemplo es la experiencia presentada en la Feria de Proyectos Capstone y de Control de la Escuela Profesional de Ingeniería de Sistemas, donde los estudiantes aplican metodologías ágiles para desarrollar soluciones innovadoras en distintos sectores. Estas ferias no solo validan la calidad de los proyectos desarrollados, sino que también permiten evaluar el impacto del aprendizaje basado en proyectos en la educación superior en ingeniería [2].

El propósito de este artículo es examinar la experiencia de la Escuela Profesional de Ingeniería de Sistemas en la puesta en marcha del curso Proyecto de Ingeniería de Software curso CAPSTONE, resaltando los retos afrontados y las estrategias implementadas con el objetivo de vincular la educación académica con las exigencias del sector de producción. Se analizan elementos tales como el contacto con compañías reales, la administración de proyectos y la valoración de habilidades, ofreciendo una perspectiva completa de las tácticas utilizadas para potenciar la enseñanza y el aprendizaje en este escenario.

II. Marco de referencia

El programa académico "Proyecto de Ingeniería de Software" se clasifica dentro de las asignaturas de tipo Capstone, reconocidas por su eficacia en la integración de conocimientos teóricos y prácticos en contextos reales. Estas

disciplinas educativas habilitan a los alumnos para afrontar retos análogos a los que se enfrentarán en su trayectoria profesional, promoviendo el desarrollo de habilidades técnicas y competencias blandas fundamentales para su rendimiento en el sector del software [6].

La cooperación entre entidades académicas y corporaciones es esencial en este escenario. La implicación activa de las corporaciones en iniciativas académicas ofrece a los estudiantes una perspectiva práctica y contemporánea de las exigencias del mercado laboral, mientras que las organizaciones se benefician de soluciones innovadoras y del talento emergente [7]. Esta sinergia fomenta el robustecimiento de la capacitación profesional y el progreso tecnológico de las entidades participantes.

No obstante, la instauración de proyectos colaborativos entre instituciones universitarias y corporaciones enfrenta una serie de obstáculos. Estos incluyen la sincronización de los objetivos académicos con las exigencias corporativas, la administración eficaz de la comunicación entre las partes involucradas y la adaptación a las metodologías de desarrollo de software empleadas en el sector industrial [5]. La superación de estos impedimentos demanda una meticulosa planificación, consensos explícitos y una disposición receptiva al cambio por parte de todos los actores implicados.

Dentro del contexto académico, la implementación de metodologías ágiles ha evidenciado ser ventajosa para la administración de proyectos de software. Estas metodologías fomentan la adaptabilidad, la colaboración continua y la entrega progresiva de productos, facilitando a los estudiantes una rápida adaptación a las modificaciones y una mejora continua de sus procesos académicos [6], La incorporación de prácticas ágiles en el programa educativo capacita a los profesionales futuros para afrontar contextos laborales dinámicos y orientados hacia la consecución de resultados.

III. MÉTODO

1. Enfoque de la Investigación

Esta investigación sigue un enfoque cualitativo, ya que se centra en la descripción y análisis de la experiencia educativa dentro del curso "Proyecto de Ingeniería de Software", sin intervención experimental directa [8]. Además, adopta un diseño descriptivo y exploratorio, permitiendo documentar los desafíos y estrategias aplicadas en la ejecución de proyectos de software con empresas reales [9].

2. Contexto del Estudio

El estudio se llevó a cabo en la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional de San Agustín de Arequipa(UNSA), específicamente en la asignatura Proyecto de Ingeniería de Software, impartida en el último año de la carrera. Este curso alineado con los criterios de ABET, tiene como objetivo evaluar integralmente las competencias adquiridas por los estudiantes a lo largo de su formación [10].

Durante el periodo de estudio, los estudiantes trabajaron en equipos de desarrollo para diseñar e implementar soluciones tecnológicas en respuesta a problemas reales planteados por diferentes instituciones. Los proyectos fueron gestionados bajo metodologías ágiles, promoviendo iteraciones continuas y retroalimentación constante entre los equipos y las empresas involucradas [11].

3. Fuentes de Información

Para documentar la experiencia y extraer conclusiones relevantes, se analizaron las siguientes fuentes de información:

- Documentos de proyectos: Incluyen los informes finales, cronogramas y planificaciones detalladas elaboradas por los estudiantes.
- Repositorios de código: Se revisaron los repositorios utilizados por los equipos para evaluar el desarrollo y evolución del software.
- Manuales de usuario y documentación técnica: Proporcionaron información sobre la calidad de la documentación generada por los estudiantes.
- Notas parciales y finales: Se utilizaron para evaluar el desempeño de los estudiantes en los distintos hitos del proyecto.
- Informes individuales y de equipo: Permitieron identificar desafíos enfrentados y estrategias aplicadas en el desarrollo del software.

4. Criterios de Selección de Proyectos

Dado que el curso "Proyecto de Ingeniería de Software" se imparte a varios grupos de estudiantes, se establecieron criterios específicos para la selección de los proyectos analizados en este estudio:

- Relevancia del problema: Se seleccionaron proyectos que abordaron problemas significativos en las instituciones colaboradoras.
- Calidad de la documentación: Se priorizaron aquellos proyectos con informes detallados y manuales bien estructurados.
- Variedad en el uso de tecnologías: Se escogieron proyectos que implementaron diferentes herramientas y metodologías ágiles.
- Disponibilidad de datos: Se consideraron únicamente aquellos proyectos cuyos documentos y repositorios estuvieran completos y accesibles para el análisis.

5. Análisis de Datos

El análisis de la información se realizó siguiendo un enfoque de análisis de contenido, categorizando los principales desafíos encontrados y las estrategias implementadas para su resolución [9]. Se identificaron patrones comunes en los documentos analizados, permitiendo extraer conclusiones sobre la efectividad del curso y su impacto en la formación de los estudiantes.

6. Limitaciones del Estudio

Si bien este estudio proporciona una visión detallada sobre la ejecución del curso "Proyecto de Ingeniería de Software", presenta algunas limitaciones que deben considerarse:

- Falta de encuestas: No se recolectaron datos directos de los estudiantes ni de las empresas colaboradoras, lo que limita la perspectiva sobre su experiencia y satisfacción con el proyecto.
- Datos basados en documentos: Toda la información analizada proviene de informes, notas y repositorios, lo que puede dejar fuera aspectos cualitativos importantes como la percepción del aprendizaje o la dinámica de trabajo en equipo.
- Muestra limitada: Aunque se seleccionaron proyectos representativos, los hallazgos no necesariamente reflejan la totalidad de experiencias en el curso.

A pesar de estas limitaciones, el estudio permite identificar patrones relevantes y sugerir estrategias para mejorar la enseñanza de la ingeniería de software en un entorno universitario.

Aunque este estudio se enfoca en tres proyectos escogidos bajo criterios particulares de importancia, documentación y variedad tecnológica, se admite que los alumnos del curso han elaborado proyectos con un nivel de sofisticación superior, integrando tecnologías en auge como el aprendizaje automático, IoT, computación en la nube y blockchain. No obstante, debido al enfoque cualitativo y exploratorio de este estudio, se seleccionó una muestra representativa que facilitara la realización de un estudio detallado de los procesos de enseñanza y administración. Para futuros estudios, se aconseja expandir la selección de proyectos y tener en cuenta factores extra en la valoración, como la implementación de tecnologías en auge, el nivel de innovación o su divulgación en eventos académicos, con el objetivo de robustecer la generalización de los descubrimientos.

IV. Resultados y Análisis

1. Descripción General de los Proyectos Analizados

Para el presente estudio, se seleccionaron tres proyectos desarrollados en el curso. Estos proyectos fueron elegidos con base en criterios de relevancia del problema abordado, calidad de la documentación, diversidad de tecnologías utilizadas y disponibilidad de datos. La tabla I resume sus características principales:

TABLA I
DESCRIPCIÓN DE PROYECTOS

Nombre del Proyecto	Institución	Problema	Tipo	Tecnologías
Sistema Integrado Sitio Web y ERP para Colegio Kinderland Markkhan	Colegio Kinderland Markkhan (personal y padres de familia)	Falta de un sistema integrado; todo se realizaba manualmente	Web responsiva	Django, PostgreSQL
Sistema de Administración de Laboratorios e Institutos de Investigación (SILEII)	Comunidad universitaria (investigadores de laboratorios e institutos de investigación)	Necesidad de apoyo administrativo y difusión de trabajos de investigación	Web responsiva	Django, Python, PostgreSQL
Gestión de Torneos Deportivos Multidisciplinarios del Colegio de Ingenieros del Perú	Colegio de Ingenieros del Perú - Consejo Departamental Arequipa (deportistas en campeonatos internos)	Falta de información sobre resultados, estadísticas y programaciones de torneos	Web responsiva	Angular, MySQL, Spring Boot

En general, estos proyectos fueron desarrollados para digitalizar y optimizar procesos administrativos y de gestión en distintas instituciones. Abarcan desde la automatización de tareas escolares y la administración de recursos de laboratorios hasta la organización de eventos deportivos. Cada uno de ellos implementó tecnologías modernas y metodologías ágiles para mejorar la eficiencia y accesibilidad de los sistemas implementados.

2. Desafíos Identificados en la Ejecución de los Proyectos

Durante el desarrollo de los proyectos, los equipos enfrentaron diversos desafíos comunes que impactaron la ejecución y gestión del software. Uno de los principales problemas fue el pobre conocimiento de la metodología ágil SCRUM, lo que generó inconvenientes en la organización del trabajo. Además, el liderazgo ineficaz en algunos equipos obligó a cambios frecuentes en la asignación de roles, afectando la toma de decisiones y la coordinación eficiente.

Otro obstáculo importante fue la carga académica y laboral de los estudiantes, ya que, al tratarse de alumnos en los últimos años de carrera, muchos también trabajaban, dificultando la disponibilidad de tiempo para avanzar en el proyecto.

Estos factores, combinados con el desconocimiento en estándares de calidad, impactaron el control sobre el desarrollo de software, lo que concuerda con estudios previos sobre la importancia de la capacitación en metodologías ágiles para mejorar la eficiencia y la calidad del producto final [12].

3. Estrategias Aplicadas para Resolver los Desafíos

Para mitigar estos desafíos, se implementaron diversas estrategias. Se asignó un tutor docente a cada equipo para guiar el proceso y se invitaron egresados con experiencia en la industria para compartir sus conocimientos. Adicionalmente, se exigieron evidencias de capacitaciones en SCRUM y en las tecnologías utilizadas, fortaleciendo las competencias técnicas de los estudiantes.

El acompañamiento continuo y la capacitación son estrategias ampliamente respaldadas en la literatura sobre formación en ingeniería de software, mejorando la calidad del aprendizaje y la capacidad de los estudiantes para gestionar proyectos reales [13]. Además, la implementación de reuniones periódicas para revisar avances permitió detectar problemas a tiempo y aplicar estrategias correctivas oportunas, favoreciendo la vinculación entre la formación académica y las necesidades de la industria [14].

4. Evaluación de los Productos Finales.

Para evaluar el desempeño de los equipos y la calidad de los productos finales, se establecieron criterios comunes de evaluación para todos los proyectos. Estos incluyeron la entrega del informe final, donde se documenta el desarrollo del proyecto desde la planificación hasta el cierre, un acta de cierre firmada por el cliente, garantizando la conformidad del usuario final, y la entrega de manuales de usuario y de calidad, donde se evidencian las pruebas realizadas y el cumplimiento de estándares de calidad. La tabla II muestra la evaluación por proyecto:

TABLA II
EVALUACIÓN DE PROYECTOS

Nombre del Proyecto	Calificación Promedio	Comentarios del Docente	Opinión del Cliente	Estado Actual
Sistema Integrado Sitio Web y ERP para Colegio Kinderland	18 - 20	Proyecto con mejor desempeño en todos los ítems de calificación	Positiva, sistema en uso	En producción
Sistema de Administración de Laboratorios e Institutos (SILEII)	16 - 18	Buen nivel de documentación y funcionalidad	Positiva, sistema en uso	En producción
Gestión de Torneos Deportivos del Colegio de Ingenieros	14 - 16	Informe docente menos favorable, ya que presenta deficiencias en documentación	Positiva, con posibilidad de continuar	En producción

La evaluación de estos proyectos refleja el impacto de un enfoque estructurado en la ejecución de proyectos de software en entornos educativos. Según la literatura, la implementación de estándares de calidad y el seguimiento por parte de clientes mejoran la preparación profesional de los estudiantes [15]. Además, la retroalimentación tanto de docentes como de clientes permite a los equipos identificar áreas de mejora en sus procesos de desarrollo, alineándolos con las demandas reales del sector tecnológico [16].

Otro aspecto destacable es que todos los productos han entrado en producción, lo que indica su utilidad y aceptación por parte de las instituciones beneficiarias. Este factor es un indicador de éxito clave, ya que demuestra la aplicabilidad de los conocimientos adquiridos en el curso y su impacto real en el entorno profesional [17].

Finalmente, el proyecto con menor evaluación académica, el de gestión de torneos deportivos, obtuvo valoraciones menos favorables en la revisión docente, aunque el cliente mostró interés en continuar con su uso y desarrollo, lo que resalta la importancia de considerar tanto la perspectiva académica como la del usuario final [18].

5.- Impacto y Aplicabilidad de los Proyectos.

Los proyectos desarrollados han generado un impacto significativo en las instituciones beneficiarias. Como parte del proceso de colaboración, las instituciones interesadas enviaron una solicitud formal para participar y asignaron un responsable que validara los avances. Como resultado, se recibieron cartas de agradecimiento y reconocimiento, lo que indica una percepción positiva sobre el aporte de los proyectos a sus procesos.

La tabla III muestra la descripción del impacto y aplicabilidad de los proyectos desarrollados en el año 2024

TABLA III
DESCRIPCIÓN DEL IMPACTO DE PROYECTOS

Aspecto	Detalles
Impacto en las instituciones	Se recibieron cartas de agradecimiento y reconocimiento por la mejora en sus procesos administrativos.
Mantenimiento	En el caso del colegio y laboratorios, existe intención de contratar a miembros del equipo para el mantenimiento del software. Para el software deportivo, se continuará con un nuevo equipo de estudiantes en el siguiente semestre.
Replicabilidad	Los proyectos quedan bajo custodia de la Escuela Profesional, formando parte de un repositorio de proyectos accesible para futuras tesis o cursos bajo supervisión docente.

La colaboración universidad-empresa en el desarrollo de software ha demostrado ser una estrategia efectiva para generar soluciones tecnológicas aplicables en contextos reales [19]. La recepción de cartas de reconocimiento por parte de las instituciones beneficiarias confirma la relevancia de estos proyectos en la mejora de procesos administrativos y operativos [20].

En cuanto a la sostenibilidad, la posibilidad de que algunos estudiantes sean contratados para el mantenimiento de

los sistemas evidencia una transición exitosa entre la formación académica y la inserción laboral [21]. Además, la creación de un repositorio de proyectos dentro de la Escuela Profesional permite que futuras generaciones de estudiantes continúen desarrollando y mejorando estas soluciones, promoviendo la innovación y la transferencia de conocimiento [22].

6. Evidencia de los proyectos.

Para respaldar la ejecución y aceptación de los proyectos desarrollados, se presentan a continuación algunas evidencias clave. Estas incluyen un acta de cierre, que certifica la entrega y validación del software por parte de las instituciones beneficiarias; capturas del repositorio GitLab, que reflejan la gestión del código fuente y la colaboración entre los estudiantes; y una muestra del manual de usuario, el cual garantiza la correcta utilización y mantenimiento del sistema implementado.

Debido a las limitaciones de extensión en este artículo, solo se incluyen estas evidencias representativas. Sin embargo, cada proyecto cuenta con documentación detallada que respalda su desarrollo, implementación y éxito en producción.

La Figura 01 muestra el repositorio GitLab del proyecto de Gestión de Torneos Deportivos, donde se gestionó el código fuente, el control de versiones y la colaboración del equipo. El uso de GitLab permitió documentar el desarrollo del software, garantizar la trazabilidad del código y mejorar la organización del proyecto.

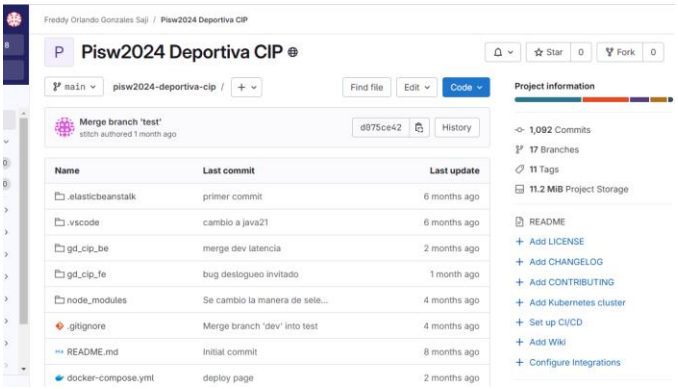

Figura 01. Repositorio GitLab del Proyecto de Gestión Deportiva

La Figura 02 presenta un fragmento del Manual de Usuario, el cual proporciona instrucciones detalladas para la correcta utilización del software. Este documento facilita la adopción del sistema por parte de los usuarios finales y asegura su operatividad sin necesidad de asistencia técnica constante.

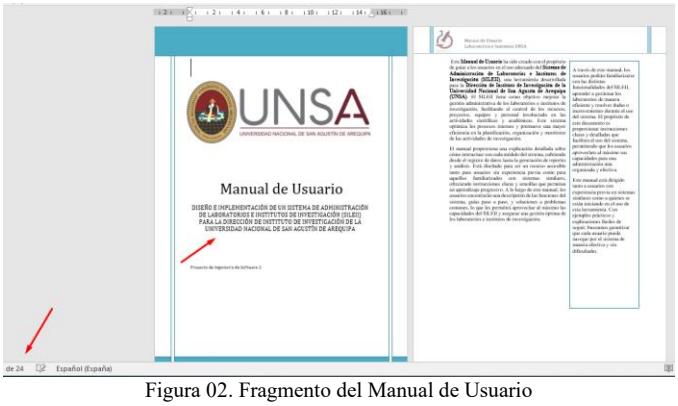

Figura 02. Fragmento del Manual de Usuario

La Figura 03 muestra un fragmento del Acta de Entrega, donde se evidencia la firma del cliente y del líder del equipo de desarrollo. Este documento certifica la conformidad del producto final y valida la aceptación del software por parte de la institución beneficiaria, asegurando el cierre formal del proyecto.

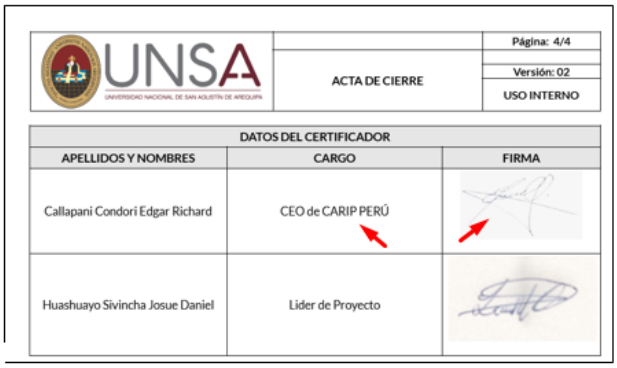

Figura 03. Fragmento del Acta de Entrega

La documentación y trazabilidad de los proyectos de software desempeñan un papel fundamental en la validación de su impacto y calidad [23]. La presentación de actas de cierre firmadas por los clientes demuestra la aceptación de los productos desarrollados, asegurando que cumplen con las necesidades planteadas. Por otro lado, el uso de repositorios de control de versiones, como GitLab, no solo facilita la gestión del código y el trabajo colaborativo, sino que también permite la auditoría del desarrollo, garantizando buenas prácticas en la programación y documentación del software [24].

Asimismo, la elaboración de manuales de usuario es un componente clave para la sostenibilidad de los sistemas, ya que permite a las instituciones beneficiarias mantener y utilizar eficientemente las soluciones implementadas sin depender exclusivamente de los desarrolladores originales [25]. Estos elementos reflejan la importancia de inculcar en los estudiantes una cultura de documentación y estandarización en

sus proyectos, aspecto altamente valorado en la industria del software [26].

IV. CONCLUSIONES

Los hallazgos derivados de esta investigación evidencian que la ejecución de proyectos prácticos en el curso de Proyecto de Ingeniería de Software constituye una estrategia eficaz para la educación profesional de los alumnos. La implicación de entidades externas en la formulación de requerimientos y verificación de entregables ejerció un efecto positivo en el aprendizaje y desarrollo de competencias técnicas y en la administración de proyectos.

La evaluación de los productos finales y el reconocimiento de las instituciones beneficiarias reflejan la calidad de los proyectos implementados. Adicionalmente, la intención de contratación de algunos estudiantes para el mantenimiento de los sistemas y la continuidad de un proyecto con otro grupo de estudiantes refuerzan la aplicabilidad y sostenibilidad de este enfoque educativo.

Sin embargo, se identificaron desafíos en la organización de los equipos, la falta de experiencia en metodologías ágiles y la gestión del tiempo, lo que resalta la importancia de mejorar ciertos aspectos en futuras ediciones del curso.

VI. RECOMENDACIONES

Fortalecer la capacitación en metodologías ágiles: Incluir talleres previos sobre SCRUM y otras metodologías de gestión de proyectos para mejorar la organización y ejecución del trabajo en equipo.

Mejorar la coordinación con las instituciones beneficiarias: Establecer reuniones periódicas más estructuradas para garantizar una mejor comunicación y alineación en las expectativas del proyecto.

Implementar un sistema de mentoría: Asignar docentes o egresados con experiencia en desarrollo de software para brindar seguimiento y apoyo a los estudiantes en aspectos técnicos y de gestión.

Reforzar la evaluación de la calidad del software: Exigir auditorías técnicas y pruebas de calidad más rigurosas para garantizar que los proyectos entregados cumplan con estándares adecuados.

Fomentar la continuidad de proyectos exitosos: Integrar los proyectos en un repositorio institucional accesible para que futuras generaciones puedan mejorarlos y expandir sus funcionalidades.

Incluir proyectos con tecnologías emergentes y mayor diversidad: Considerar, en futuras investigaciones, la incorporación de una muestra más amplia de proyectos que aborden tecnologías emergentes como inteligencia artificial, internet de las cosas, blockchain, análisis de datos, computación en la nube, entre otras. Esto permitirá una evaluación más integral del alcance y nivel de innovación alcanzado por los estudiantes. Asimismo, se recomienda definir criterios complementarios de evaluación que consideren no solo la funcionalidad y documentación, sino también el grado de innovación, escalabilidad y la posibilidad de publicación en eventos académicos como ferias, congresos o revistas especializadas.

AGRADECIMIENTO/RECONOCIMIENTO

Se agradece a la Escuela Profesional de Ingeniería de Sistemas de la UNSA por el apoyo en la implementación del curso y la facilitación de recursos. Asimismo, se reconoce el esfuerzo de los estudiantes y docentes involucrados, así como la participación de las instituciones beneficiarias que confiaron en estos proyectos y contribuyeron a su éxito. Se hace un reconocimiento especial a los colegas que contribuyeron en la creación de este articulo por la revisión de toda la documentación necearía para contar la experiencia en esta escuela profesional.

REFERENCES

[1] Universidad Nacional de San Agustín de Arequipa, "Escuela profesional de Ingeniería de Sistemas recibe acreditación de ABET por 10 años," 31 de octubre de 2022. [En línea]. Disponible en: https://www.unsa.edu.pe/escuela-profesional-de-ingenieria-de-sistemas-recibe-acreditacion-de-abet-por-10-anos/. [Accedido: 1-feb-2025].

[2] F. O. Gonzales Saji, C. Baluarte-Araya, K. Guevara-Puente-de-la-Vega, y O. Sharhorodska, "Feria De Proyectos Cursos Capstone Y De Control Una Experiencia En La Escuela Profesional De Ingeniería De Sistemas," enero de 2021. [En línea]. Disponible en: https://www.researchgate.net/publication/348759654_Feria_De_Proyectos_Cursos_Capstone_Y_De_Control_Una_Experiencia_En_La_Escuela_Profesional_De_Ingenieria_De_Sistemas. [Accedido: 1-feb-2025].

[3] C. A. Urrea Merchán, J. A. Duque Mogollón y E. Vásquez González, "Capstone Design en los Programas de Ingeniería: Una Revisión Sistemática," Hamut'ay, vol. 8, no. 2, pp. 31-42, 2021. [En línea]. Disponible en: https://dialnet.unirioja.es/servlet/articulo?codigo=8099399. [Accedido: 1-feb-2025].

[4] Universidad Nacional de San Agustín de Arequipa, "Programa de Estudios Ingeniería de Sistemas - FIPS – UNSA," [En línea]. Disponible en: https://fips.unsa.edu.pe/ingenieriadesistemas/. [Accedido: 1-feb-2025].

[5] L. H. Silva, R. X. Castro y M. C. Guimarães, "Supporting Real Demands in Software Engineering with a Four Steps Project-Based Learning Approach," arXiv preprint arXiv:2102.01631, 2021. [En línea]. Disponible en: https://arxiv.org/abs/2102.01631. [Accedido: 1-feb-2025].

[6] M. A. Nelson y L. Ponciano, "Experiences and insights from using GitHub Classroom to support Project-Based Courses," arXiv preprint arXiv:2103.07242, 2021. [En línea]. Disponible en: https://arxiv.org/abs/2103.07242. [Accedido: 1-feb-2025].

[7] C. A. Urrea Merchán, J. A. Duque Mogollón y E. Vásquez González, "Capstone Design en los Programas de Ingeniería: Una Revisión

Sistemática," Hamut'ay, vol. 8, no. 2, pp. 31-42, 2021. [En línea]. Disponible en: https://dialnet.unirioja.es/servlet/articulo?codigo=8099399. [Accedido: 1-feb-2025].
[8] Y. S. Lincoln y E. G. Guba, Naturalistic Inquiry, Beverly Hills, CA: Sage Publications, 1985. [En línea]. Disponible en: https://archive.org/details/naturalisticinqu00linc. [Accedido: 1-feb-2025].
[9] M. Q. Patton, Qualitative Research & Evaluation Methods: Integrating Theory and Practice, 4ª ed., Thousand Oaks, CA: Sage Publications, 2015. [En línea]. Disponible en: https://archive.org/details/michael-quinn-patton-qualitative-research-evaluation-methods-integrating-theory-. [Accedido: 1-feb-2025].
[10] Universidad Nacional de San Agustín de Arequipa, "Programa de Estudios Ingeniería de Sistemas - FIPS – UNSA," [En línea]. Disponible en: https://fips.unsa.edu.pe/ingenieriadesistemas/. [Accedido: 1-feb-2025].
[11] K. Beck et al., "Manifesto for Agile Software Development," 2001. [En línea]. Disponible en: https://agilemanifesto.org/. [Accedido: 1-feb-2025].
[12] A. H. F. N. Segundo, J. S. da Cunha Neto, P. C. S. Barbosa, y R. F. Santana, "Applying PBL in the Development and Modeling of Kinematics for Robotic Manipulators with Interdisciplinarity between Computer-Assisted Project, Robotics, and Microcontrollers," arXiv preprint arXiv:2203.16927, 2022. Disponible en: https://arxiv.org/abs/2203.16927.
[13] L. H. Silva, R. X. Castro, y M. C. Guimarães, "Supporting Real Demands in Software Engineering with a Four Steps Project-Based Learning Approach," arXiv preprint arXiv:2102.01631, 2021. Disponible en: https://arxiv.org/abs/2102.01631.
[14] A. Al-Ajlan, "A Comparative Study Between Web 2.0 and Web 3.0," International Conference on Advanced Computer Theory and Engineering, 2009. Disponible en: https://ieeexplore.ieee.org/document/5349513.
[15] L. Straccia, P. Zanetti, y M. F. Pollo-Cattaneo, "Definición de un estándar para la evaluación de calidad de software educativo," XXV Congreso Argentino de Ciencias de la Computación, 2019. Disponible en: https://grupogemis.com.ar/wp-content/uploads/2019/10/CACIC-2019.-Straccia-Zanetti-Pollo.-Definicion-de-un-estandar-sw-educativo.pdf.
[16] "La retroalimentación efectiva en estudiantes desde la perspectiva de los docentes," Transdigital, vol. 1, no. 1, pp. 45-60, 2020. Disponible en: https://www.revista-transdigital.org/index.php/transdigital/article/view/63/127.
[17] L. Straccia, P. Zanetti, y M. F. Pollo-Cattaneo, "Normalización de la evaluación de calidad de producto software para el ámbito educativo," XXIV Congreso Argentino de Ciencias de la Computación, 2018. Disponible en: https://sedici.unlp.edu.ar/bitstream/handle/10915/73090/Documento_completo.pdf-PDFA.pdf?isAllowed=y&sequence=1.
[18] "La retroalimentación: la clave para una evaluación orientada al aprendizaje," Redalyc, 2016. Disponible en: https://www.redalyc.org/pdf/567/56752038009.pdf.
[19] Laredo Health, TAMIU engineering students partner for high-tech solutions. Laredo Morning Times, 16 de febrero de 2025. https://www.lmtonline.com/local/article/laredo-public-health-tamiu-partnership-2025-tech-20166613.php
[20] El Camino de la Miranda presenta 'IncluInnovación: desarrollo de proyectos tecnológicos-sociales en la FP'. Cadena SER, 13 de febrero de 2025. https://cadenaser.com/castillayleon/2025/02/13/el-camino-de-la-miranda-presenta-incluinnovacion-desarrollo-de-proyectos-tecnologicos-sociales-en-la-fp-radio-palencia/
[21] Agile Minds, Innovative Solutions, and Industry-Academia Collaboration: Lean R&D Meets Problem-Based Learning in Software Engineering Education. arXiv preprint, 22 de julio de 2024. https://arxiv.org/abs/2407.15982
[22] Innovar en compañía: el Repositorio de Innovación Educativa de la UNAM. Revista Digital Universitaria, 2020. https://www.revista.unam.mx/2020v21n2/innovar_en_compania_el_repositorio_de_innovacion_educativa_de_la_unam/
[23] Visure Solutions. "Requisitos de la matriz de trazabilidad para software: garantizar el...". Disponible en: https://visuresolutions.com/es/blog/software/matriz-de-trazabilidad/
[24] Guru. "Cómo usar los repositorios de GitLab: una guía completa". Disponible en: https://www.getguru.com/es/reference/how-to-use-gitlab-repositories-a-comprehensive-guide
[25] Distillery. "Documentación de software, un enfoque basado en el valor". Disponible en: https://distillery.com/es/blog/documentacion-de-software-el-valor-de-una-documentacion-adecuada-a-su-finalidad/
[26] In-Com. "Comprender la importancia de la trazabilidad del código". Disponible en: https://www.in-com.com/es/blog/code-traceability/