

Metodología basada en proyectos para la enseñanza
de ingeniería de software: Experiencia en la carrera
de Tecnología de la Información de la Universidad
Técnica Luis Vargas Torres

Stalin Francis Q.

Universidad Técnica Luis Vargas Torres

Resumen

En este artículo se presenta la experiencia educativa lograda en la asignatura “Ingeniería de Software I” durante el periodo 2024-1S en la Carrera en Tecnologías de la Información de la Universidad Técnica Luis Vargas Torres de Esmeraldas. La experiencia consistió en la aplicación de la metodología de aprendizaje basado en proyectos (ABP), integrada con actividades prácticas que combinaron habilidades técnicas y sociales. Al final del ciclo, se evaluó la estrategia mediante una encuesta de satisfacción, cuyos resultados y retroalimentación contribuyeron a su mejora continua.

PALABRAS CLAVES: Ingeniería de Software, Aprendizaje Basado en Proyecto, Tecnología de la Información.

Abstract

This article presents the educational experience achieved in the subject “Software Engineering I” during the period 2024-1S in the Information Technology Degree at the Luis Vargas Torres Technical University of Esmeraldas. The experience consisted of the application of the project-based learning (PBL) methodology, integrated with practical activities that combined technical and social skills. At the end of the cycle, the strategy was evaluated through a satisfaction survey, whose results and feedback contributed to its continuous improvement.

KEYWORDS: Software Engineering, Project-Based Learning, Information

Technology.

1. Introducción

La asignatura Ingeniería de Software I forma parte de eje de especialización del plan de estudio de la carrera de Tecnología de la Información, en la Facultad de Ingenierías de la Universidad Técnica Luis Vargas Torres de Esmeraldas. La enseñanza de esta asignatura presenta desafíos relacionados con la integración efectiva entre teoría y práctica, lo que a menudo se traduce en un insuficiente preparación para el entorno laboral. Esta situación se ve agravada por las limitaciones propias de la universidad pública, tales como la escasez de recursos técnicos y pedagógicos, la falta de espacios adecuados para el aprendizaje, y el tiempo restringido para tutorías y refuerzo académico. Además el contexto local influye significativamente: Esmeraldas es una ciudad con escaso desarrollo en el ámbito de la tecnología de la información. Como consecuencia, las clases de esta asignatura han sido tradicionalmente impartidas mediante metodologías basadas en la lección magistral y el modelo de escuela tradicional ([Yadav, Subedi, Lundeborg, y Bunting, 2011](#); [Becker, Parker, y Al-Fedaghi, 2012](#)).

En este contexto, se implementó y aplicó la metodología de aprendizaje basada en proyectos (ABP) en la asignatura Ingeniería de Software I, con el objetivo de promover un aprendizaje más activo, autónomo y orientado a la solución de problemas reales. Esta estrategia didáctica busca superar las limitaciones propias del enfoque tradicional, alineándose con las demandas del mercado

laboral, que requiere profesionales capaces de adaptarse a entornos cambiantes y complejos (Dabbagh, 2005).

La puesta en marcha del ABP en la asignatura de Ingeniería de Software I necesitó realizar reajustes a los contenidos del sílabo, planes semestrales y diapositivas. Además, contempló el desarrollo de un proyecto de aula para la creación de un sistema de información. Este proyecto abarcó desde la concepción del tipo de sistema de información a desarrollar hasta la creación, documentación y despliegue de un sistema de información web totalmente funcional llamado SICA. La gestión de este proyecto de aula, cuyo marco de trabajo ya está definido en la teoría del Ciclo de Vida del Software y la metodología ágil (SOMMERVILLE, 2005), requirió de un tiempo mucho mayor a los cuatro meses que dura un ciclo académico, al menos si se asume la tarea de la etapa de implementación. Por lo tanto, para lograr este objetivo titánico, fue necesario trabajar sobre un software ya desarrollado por el docente, garantizando un pleno conocimiento de su arquitectura y facilitando su uso como herramienta pedagógica para los estudiantes (Pressman, 2014).

Los ajustes realizados tanto al sílabo, planes semestrales y guías de prácticas de laboratorio no fueron realizados de forma arbitraria. Estos cambios se llevaron a cabo en función de los resultados de aprendizaje y los contenidos mínimos definidos por el proyecto de carrera (CES, 2017). Además, se consideraron las directrices curriculares para el programa de licenciatura en Tecnología de la Información elaboradas por la Association for Computing Machinery (ACM) y la IEEE

Computer Society (IEEE-CS) (on Information Technology Curricula, 2017), asegurando que el contenido estuviera alineado con los estándares internacionales.

Adicionalmente, la implementación de la metodología ABP también promueve el desarrollo de habilidades blandas, como el trabajo en equipo, la comunicación efectiva y la gestión del tiempo, competencias que son cada vez más valoradas en la industria (Fowler, 2004; Salmon, 2000). Estos aspectos fueron integrados en las actividades académicas mediante una planificación colaborativa y un seguimiento constante por parte del docente, garantizando que los estudiantes no solo desarrollaran capacidades técnicas, sino también interpersonales.

Por último, este enfoque innovador también representa un compromiso con la mejora continua de la calidad educativa, superando barreras institucionales y geográficas, y contribuyendo a formar profesionales con una visión global y capacidad de resolver problemas del mundo real. La experiencia adquirida a través de este modelo pedagógico se alinea con las tendencias actuales en la educación superior y refuerza la importancia de preparar a los estudiantes para enfrentar los desafíos de la cuarta revolución industrial (Schwab, 2017).

2. Metodología

Se realizó un estudio cuantitativo transversal con 50 estudiantes (18 mujeres y 32 hombres) del 5to ciclo de la carrera de Tecnología de la Información de la Universidad Técnica Luis Vargas Torres, en la asigna-

natura de Ingeniería de Software I durante el período 2024-1S. Los estudiantes, al final de un proceso de aprendizaje que duró 4 meses, respondieron una encuesta de satisfacción sobre la estrategia didáctica basada en proyectos aplicada. Este enfoque se fundamenta en investigaciones previas que destacan la eficacia del Aprendizaje Basado en Proyectos (ABP) en disciplinas técnicas (Smith, 2019; Prince, 2004).

El diseño del curso Ingeniería de Software I comprendió cuatro unidades (cuadro #??), en las que cada unidad integró teoría y práctica mediante tres tipos de acti-

vidades: aprendizaje teórico (Actividad A), aprendizaje práctico (Actividad B) y aprendizaje autónomo (Actividad C). Cada dos unidades se aplicó una evaluación integral (Actividad E), resultando en un total de 14 actividades de aprendizaje y evaluación durante el ciclo lectivo. Este enfoque permitió no solo un aprendizaje integral del proceso de desarrollo de software, sino también la creación de un producto final funcional y documentado, como lo sugieren las buenas prácticas en proyectos educativos de ingeniería (Felder y Brent, 2005; Hughes y Cotterell, 2017).

Primer Hemisemestre							Segundo Hemisemestre						
Unidad I			Unidad II			E1	Unidad III			Unidad IV			E2
A1-1	B1-1	C1-1	A1-2	B1-2	C1-2	E1	A2-1	B2-1	C2-1	A2-2	B2-2	C2-2	E2

Cuadro 1: Esquema de Actividades de aprendizaje y evaluación en la UTELVT

El proceso inició con la presentación del docente, los estudiantes y una introducción a la asignatura en dos sesiones iniciales. Durante estas sesiones, se abordaron los fundamentos de la Ingeniería de Software, complementados con una actividad de aprendizaje autónomo basada en la lectura de un artículo académico sobre su importancia (Pressman, 2014). En sesiones posteriores, se introdujo la metodología ágil Scrum, formando grupos de cinco estudiantes con roles definidos (Product Owner, Scrum Master y Team Developers), siguiendo modelos pedagógicos que destacan la relevancia del trabajo colaborativo (Kuz, 2021).

En la primera fase práctica, los estudiantes desarrollaron habilidades para la recolección y análisis de requerimientos. En la

actividad A1-1, se crearon historias de usuario mediante interacción con usuarios ficticios. Posteriormente, en la actividad A1-2, se verificaron estas necesidades con usuarios reales, documentando firmas y evidencias fotográficas. Este proceso culminó en la creación del Documento de Especificaciones Funcionales y No Funcionales (DEFN) en la actividad B1-1, seguido por el diseño de 20 prototipos de interfaces en la actividad B2-2. Estas actividades permitieron cubrir las etapas iniciales del ciclo de vida del software: recolección de requerimientos, análisis y diseño (Boehm, 1988).

En el segundo parcial, los estudiantes se adentraron en las etapas de desarrollo, mantenimiento y despliegue del software. Durante la actividad A2-1, se introdu-

jo la arquitectura Modelo-Vista-Controlador (MVC) y el uso de GitHub. Los estudiantes crearon un módulo funcional denominado SIGECA y lo desplegaron en un servidor real en la actividad A2-2. Finalmente, documentaron y presentaron el módulo en las actividades B2-1 y B2-2. Estas tareas fomentaron un aprendizaje práctico en entornos reales y alineados con las tendencias de la industria tecnológica (Fitzgerald, 2006).

Paralelamente, se asignaron lecturas científicas relacionadas con los temas abordados en las actividades C1-1 y C1-2, fomentando la capacidad analítica y el desarrollo de soluciones creativas a problemas comple-

jos. Este enfoque integral permitió a los estudiantes vivir la teoría y práctica del desarrollo ágil de software, logrando resultados significativos tanto en la adquisición de conocimientos como en la creación de productos funcionales.

Cada una de las actividades que evalúan los temas abordados según el cuadro # 2, fueron organizadas e incluidas en el plan semestral, dosificadas en el tiempo adecuado para que los estudiante puedan cumplir con cada una de ellas sin interferir en las actividades de las otras asignaturas por ellas tomadas, como se detalla en el cuadro # 3.

Cuadro 2: Unidades, temas y sus actividades de aprendizaje para la cátedra Ingeniería de Software I

Unidad	Tema	Tema adaptado	Actividad de aprendizaje
Unidad I	El software y la ingeniería de software	Introducción a la Ingeniería de software	<ul style="list-style-type: none"> ■ A1-1: Participación en clase respondiendo preguntas sobre el tema tratado. ■ B1-1: Taller para la creación de historias de usuario y levantamiento de requerimiento. ■ C1-1: Lectura de artículo científico. “¿Existe una situación de crisis del software educativo?”

Unidad	Tema	Tema adaptado	Actividad de aprendizaje
Unidad II	Gestión del software	Metodología Ágil Scrum	<ul style="list-style-type: none"> ■ A1-2: Participación en clase respondiendo preguntas sobre el tema tratado. ■ B1-2: Taller para la creación de prototipos y dramatización sobre metodología ágil Scrum. ■ C1-2: Lectura de artículo científico. <i>“Metodologías Ágiles en el Desarrollo de Software”</i> ■ E1: Evaluación sumativa primer parcial.
Unidad III	Análisis de requisitos del sistema y del software	Arquitectura del Software MVC, framework y Github	<ul style="list-style-type: none"> ■ A2-1: Participación en clase respondiendo preguntas sobre el tema tratado. ■ B2-1: Creación, mantenimiento de una aplicación web (uso de Git, Github, framework, MVC, Servidor web). ■ C2-1: Lectura de artículo científico. <i>“Frameworks PHP basados en la arquitectura Modelo-Vista-Controlador para desarrollo de aplicaciones web.”</i>

Unidad	Tema	Tema adaptado	Actividad de aprendizaje
Unidad IV	Dimensiones del Software	Documentación del software	<ul style="list-style-type: none"> ■ A2-2: Participación en clase respondiendo preguntas sobre el tema tratado. ■ B2-2: Mantenimiento y evolución de la aplicación web). ■ C2-2: Lectura de artículo científico. <i>“DevOps in Practice for Education Management Information System at ECNU”</i> ■ E2: Evaluación sumativa final.

3. Resultados

La asignatura de Ingeniería de Software I, se la dictó en el quinto ciclo con una duración de cuatro meses, con la carga horaria de cinco horas semanales (tres de teoría y

dos de laboratorio). La planificación contempló un total de 32 sesiones, distribuidas en dos sesiones por semana. En el cuadro # 3 se presenta la planificación semestral, donde se detallan las 32 sesiones junto con los temas a impartir y un resumen general de las actividades programadas.

Cuadro 3: Plan semestral de la asignatura Ingeniería de Software I.

Sesión	Fecha y duración	Lugar	Tema	Actividades desarrolladas	Metodologías pedagógicas
1	Martes, 6-08-2024 (3h)	Aula de Clase	Presentación del docente, la clase, prueba de diagnóstico y entrega de planificación.	<ul style="list-style-type: none"> ■ Presentación del docente y los estudiantes. ■ Prueba diagnóstica. ■ Entrega del silabo, plan semestral y proyecto de aula. 	Aprendizaje Basado en la Inducción

Sesión	Fecha y duración	Lugar	Tema	Actividades desarrolladas	Metodologías pedagógicas
2	Miércoles, 07-08-2024 (2 horas)	Aula de clase	Socialización, análisis de la planificación y el proyecto de aula.	<ul style="list-style-type: none"> ■ Socialización de los resultados de la prueba de diagnóstico: fortalezas y debilidades. ■ Análisis del sílabo, plan semestral y proyecto de aula. ■ Preguntas y respuesta sobre la planificación a desarrollar (Actividad A1-1: Participación en clase). 	Aprendizaje colaborativo
3	Martes, 13-08-2024 (3h)	Aula de clase	Introducción a la ingeniería de software.	<ul style="list-style-type: none"> ■ Charla introductoria con énfasis en el ciclo de vida de software y la metodología ágil Scrum. ■ Creación de grupos scrum (5 integrantes). ■ Preguntas y respuestas sobre el tema tratado (Actividad A1-1: Participación en clase) 	Clase Magistral y Discusión Dirigida

Sesión	Fecha y duración	Lugar	Tema	Actividades desarrolladas	Metodologías pedagógicas
4	Miércoles, 14-08-2024 (3 horas)	Laboratorio o Aula google meet	Registro en el sistema SICA.	<ul style="list-style-type: none"> ■ Socialización del artículo científico. “¿Existe una situación de crisis del software educativo?”. ■ Asignación de la actividad C1-1. ■ Introducción a la plataforma SICA ■ Registro de nuevos usuarios. ■ Exploración de módulo como evento y portafolio. 	Aprendizaje Basado en proyecto
5	Martes, 20-08-2024 (2 horas)	Aula de clase	Foro de discusión, ejecución de la actividad C1-1	<ul style="list-style-type: none"> ■ Cada grupo Scrum se presentara para discutir el tema del artículo. ■ La actividad es grabada en video para su posterior evaluación. 	Aprendizaje Basado en Casos.
6	Miércoles, 21-08-2024 (2 horas)	Lab. de Computación	Exploración del sistema SICA	<ul style="list-style-type: none"> ■ Revisión de los módulos implementados. ■ Navegación por los módulo de “Persona”, “documento”, “distributivo” y otros de SICA. 	Aprendizaje experimental.

Sesión	Fecha y duración	Lugar	Tema	Actividades desarrolladas	Metodologías pedagógicas
7	Martes, 27-08-2024 (3 horas)	Aula de clase	Introducción a la ingeniería de requerimientos y Creación de historias de usuario.	<ul style="list-style-type: none"> ■ Introducción teórica. ■ Ejecución del taller creación de historias de usuario(Actividad A1-1). ■ Validación de historias de usuario. ■ Conversión en requerimientos funcionales y no funcionales. 	Talleres Prácticos.
8	Miércoles, 28-08-2024 (2 horas)	Aula de clase	Levantamiento de requerimiento(Actividad B1-1).	<ul style="list-style-type: none"> ■ Dramatización del levantamiento de requerimiento. ■ En clases los estudiantes simularan el levantamiento de requerimiento utilizando a otro estudiante como usuario. ■ Entrevista a usuario (Actividad B1-1). 	Aprendizaje Basado en Problemas.
SEGUNDA UNIDAD					
9	Martes, 03-09-2024 (2 horas)	Aula de clase	Metodología Ágil, Scrum.	<ul style="list-style-type: none"> ■ Introducción a las metodologías Ágiles, énfasis en Scrum. ■ Preguntas y respuestas (Actividad A1-2: Participación en clases). 	Aprendizaje Activo.

Sesión	Fecha y duración	Lugar	Tema	Actividades desarrolladas	Metodologías pedagógicas
10	Miércoles, 04-09-2024 (3 horas)	Aula de clase	Socialización de artículo sobre scrum	<ul style="list-style-type: none"> ■ Socialización de artículo sobre scrum. ■ Preguntas y respuestas (Actividad A1-2: Participación en clase). ■ Asignación de actividad C1-2. 	Simulación y Rol.
11	Martes, 10-09-2024 (2 horas)	Aula de clase.	Creación de prototipos, Interfaces de usuario.	<ul style="list-style-type: none"> ■ Introducción a los prototipos . ■ Taller creación de prototipos (Actividad A1-2: Participación en clases). 	Práctica Guiada.
12	Miércoles, 11-09-2024 (3 horas)	Aula de clase	Foro de discusión Actividad C1-2.	<ul style="list-style-type: none"> ■ Los grupos Scrum realizan un foro de discusión sobre los artículos. ■ Se graba en video (Actividad C1-2) 	Aprendizaje Basado en Proyectos.
13	Martes, 17-09-2024 (2 horas)	Aula de clase	Requerimientos vs prototipos.	<ul style="list-style-type: none"> ■ Comparación de requerimientos vs prototipos. ■ Feedback de actividades anteriores. 	Análisis Comparativo.

Sesión	Fecha y duración	Lugar	Tema	Actividades desarrolladas	Metodologías pedagógicas
14	Miércoles, 18-09-2024 (3 horas)	Aula de clase	Dramatización metodología Scrum.	<ul style="list-style-type: none"> ■ Análisis de dramatizaciones anteriores. ■ Planificación de la dramatización actual (Actividad BOA1-2). 	Dramatización.
15	Martes, 24-09-2024 (2 horas)	Ambiente Externos	Dramatización Metodología Scrum.	<ul style="list-style-type: none"> ■ Dramatización metodología Scrum (Actividad B2-1). 	Aprendizaje Basado en Proyectos.
16	Miércoles, 25-09-2024 (2 horas)	Lab. Audio-visual	Evaluación sumativa primer parcial	<ul style="list-style-type: none"> ■ Actividad E1: Evaluación sumativa primer parcial. 	Dramatización.
TERCERA UNIDAD					
17	Martes, 01-10-2024 (3 horas)	Aula de clase	Documentación de requerimientos.	<ul style="list-style-type: none"> ■ Introducción al documento de requerimientos. ■ Socialización Actividad E1. (Documentación de los requerimientos). 	Clase Teórico-Práctica.
18	Miércoles, 02-10-2024 (3 horas)	Aula de clase	Revisión de la documentación de requerimientos.	<ul style="list-style-type: none"> ■ Revisión de actividad documental (actividad E1). ■ Corrección de mejoras. 	Feedback Constructivo.

Sesión	Fecha y duración	Lugar	Tema	Actividades desarrolladas	Metodologías pedagógicas
19	Martes, 08-10-2024 (2 horas)	Aula de clase	Arquitectura del software(MVC)	<ul style="list-style-type: none"> Charla sobre el modelo-vista-controlador(MVC). Asignación de un módulo SICA a los grupos Scrum. 	Aprendizaje Basado en Proyecto.
20	Miércoles, 09-10-2024 (3h)	Aula de clase	Socialización de artículo científico para la Actividad C2-1.	<ul style="list-style-type: none"> Análisis de artículos sobre MVC y framework. Asignación de Actividad C2-1 Frameworks PHP basados en la arquitectura Modelo-Vista-Controlador para desarrollo de aplicaciones web.	Lectura Crítica y Debate.
21	Martes, 15-10-2024 (2 horas)	Aula de clase	Foro de discusión - Actividad C2-1.	<ul style="list-style-type: none"> Foro de discusión Actividad c2-1. Grabación de la actividad. 	Clase Teórico-Práctico.
22	Miércoles, 16-10-2024 (2 horas)	Laboratorio o Google meet	Taller creación y configuración de cuenta en Github.	<ul style="list-style-type: none"> Creación de cuentas y repositorios. Subida del MVC del módulo asignado. 	Práctica Guiada.
23	Martes, 22-10-2024 (3 horas)	Aula de clase	Despliegue de módulo SICA.	<ul style="list-style-type: none"> Socialización y envío de la actividad B2-1 Deploy de módulo. 	Aprendizaje Basado en Proyecto.

Sesión	Fecha y duración	Lugar	Tema	Actividades desarrolladas	Metodologías pedagógicas
24	Miércoles, 23-10-2024 (2 horas)	Laboratorio Aula Google meet.	Configuración y despliegue en hosting	<ul style="list-style-type: none"> ■ Preparación de la primera versión de la plataforma web. ■ Ejecución actividad B2-1. 	Práctica Guiada.
CUARTA UNIDAD					
25	Martes, 29-10-2024 (3 horas)	Lab. de Computación	Documentación del Software	<ul style="list-style-type: none"> ■ Importancia de la documentación en el ciclo de vida del software. ■ Tipo de documentación y características. ■ Github para documentar (ejemplo: documentar un función). 	Clase Práctica.
26	Miércoles, 30-11-2024 (2 horas)	Laboratorio Aula google meet	Taller documentación.	<ul style="list-style-type: none"> ■ El docente junto con un voluntario documentaran una función. ■ Cada estudiante documentara un función de su proyecto. 	Presentación Dirigida.
27	Martes, 05-11-2024 (3h)	Lab. de Computación	Mantenimiento y ajustes.	<ul style="list-style-type: none"> ■ Socialización de artículo científico sobre DevOps. ■ Evaluación Actividad A2-2 (Participación en clase). 	Práctica guiada.

Sesión	Fecha y duración	Lugar	Tema	Actividades desarrolladas	Metodologías pedagógicas
28	Miércoles, 06-11-2024 (2h)	Aula de clase	Foro Actividad C2-2	<ul style="list-style-type: none"> ■ Ejecución de Actividad c2-2 DevOps in Practice for Education Management Information System at ECNU. 	Lectura Crítica y Debate.
29	Martes, 19-11-2024 (3h)	Laborat. o Aula meet	Taller de programación, proyecto final.	<ul style="list-style-type: none"> ■ Con un grupo a la vez el docente trabaja en el proyecto final. ■ Todos los grupos deben estar atento a las instrucciones del docente. ■ La participación será calificado como parte de la Actividad A2. 	Clase Magistral.
30	Martes, 26-11-2024 (3h)	Laborat.	Presentación proyecto final.	<ul style="list-style-type: none"> ■ Exposición de actividad grupal. ■ Edición de materiales. 	Aprendizaje Basado en Proyectos.
31	Miércoles, 27-12-2024 (3h)	Laborat.	Aplicación final	<ul style="list-style-type: none"> ■ Presentación final de la aplicación web. ■ Todos los estudiantes deberán evidenciar su participación en Github. 	Aprendizaje Basado en Proyectos.

Sesión	Fecha y duración	Lugar	Tema	Actividades desarrolladas	Metodologías pedagógicas
32	Miércoles, 04-12-2024 (3h)	Aula de clase	Evaluación final sumativa.	<ul style="list-style-type: none"> ■ Actividad E2: Evaluación sumativa final. ■ Encuesta de satisfacción (Actividad E2). 	Evaluación Sumativa.

El proceso de aprendizaje comenzó en la **Sesión #1**, realizada el martes 6 de agosto del 2024, en el paralelo A, con el tema **“Presentación del docente, la clase, prueba de diagnóstico y entrega de planificación”**. Durante esta sesión, el docente se presentó y solicitó que cada estudiante realizara una breve introducción sobre el mismo, preguntas como “¿Cuáles son sus nombres y apellidos?”, “¿De que colegio proviene?”, “¿Porqué escogió la carrera?” y “¿Qué aspira a ser en su ejercicio profesional?”. Esta actividad permitió establecer una conexión inicial, creando un ambiente más amigable y accesible para los estudiantes. Posteriormente, se aplicó una prueba de diagnóstico con el propósito de evaluar los conocimientos previos que cada estudiante traía consigo al inicio del aprendizaje. Finalmente, se entregó el sílabo de la asignatura para que los estudiantes lo revisaran en preparación para su análisis en la siguiente sesión.

En la **sesión #2** con el tema **“Socialización, análisis de la planificación y el proyecto de aula”**, se llevó a cabo un análisis detallado del sílabo. Durante esta revisión, se destacaron los temas a abordar, las actividades de aprendizaje propuestas y la metodología basada en proyectos que guió el desarrollo de la asignatura. Posteriormente, se examinó el

proyecto de aula, poniendo especial énfasis en cada una de las actividades planificadas. Cada actividad contaba con un tiempo y una duración previamente establecida, la cual no puede ser alterada. Por ello, se resaltó la importancia de respetar los tiempos planificados para garantizar el cumplimiento efectivo de los objetivos y el éxito del proyecto. Al final de esta sesión se dan indicaciones del material que se tiene que revisar (videos y diapositivas) sobre el tema de la siguiente sesión recordando que la actividad A1-1 que consiste en la participación en clase con preguntas sobre la unidad No 1 será evaluada durante las 4 primeras semanas.

En la **sesión #3**, con el tema **“Introducción a la ingeniería de software”**, se abordaron los fundamentos teóricos de esta disciplina, haciendo énfasis en aspectos claves como el ciclo de vida del software, las metodologías de desarrollo y su evolución a lo largo del tiempo. Se profundizó especialmente en la metodología ágil Scrum, destacando su enfoque en la gestión de proyectos y los roles que desempeñan los miembros del equipo. Durante esta sesión, se conformaron grupos de trabajo con cinco estudiantes, asignándoles roles específicos según la metodología Scrum. Al final de la sesión se evaluó en algunos estudiantes la actividad A1-1, con preguntas y

respuestas sobre el tema tratado, y se asignó la actividad de aprendizaje autónomo C1-1 como tarea para consolidar los conceptos trabajados en clase.

En la **sesión #4**, con el tema **“Registro en el Sistema de Información de Control Académica (SICA)”**, se socializó un artículo científico titulado “¿Existe una situación de crisis del software educativo?”, cuyo objetivo era que los estudiantes comprendieran la importancia de desarrollar y mantener sistemas de información educativos, un desafío gestionado en esta asignatura. Luego de lo cual los estudiantes llevaron consigo la Guía Práctica de Laboratorio(GPL-IS1-001), la cual detalla el procedimiento para acceder a la plataforma SICA. Durante la práctica, se realizaron actividades como la creación de un nuevo usuario, el registro en el sistema y la exploración de los módulos disponibles para el perfil de estudiante, tales como Eventos y Portafolio. El objetivo de esta actividad en el laboratorio fue que, tras la lectura del artículo asignado en la actividad C1-1, los estudiantes identificaran las carencias y oportunidades de mejora en un software educativo, conectando la teoría con la práctica, al final de esta sesión se continuó evaluando la actividad A1-1 haciendo preguntas a un grupo diferente de estudiantes y registrando su participación en clase.

En la **sesión #5** con el tema **“Foro de discusión, ejecución de la actividad C1-1”**, se realizó el Foro de discusión donde cada grupo en un lapso de 10 a 15 minutos discutían lo leído y comprendido en el artículo leído. Durante la sesión, se brindó retroalimentación detallada, destacando tanto los puntos fuertes como las áreas de mejora en la pre-

sentación de este tipo de trabajos. El objetivo fue proporcionar a los estudiantes herramientas y recomendaciones para optimizar sus futuras entregas.

En la **sesión #6**, se trabajó en el laboratorio de computación con el tema **“Explorando el sistema de información SICA”**. Con la guía de laboratorio, los estudiantes exploraron los distintos módulos que componen el sistema de información SICA. Sin embargo, por motivos de seguridad, los estudiantes no tuvieron acceso completo a algunos módulos. Durante la sesión, realizaron anotaciones basándose en las explicaciones proporcionadas por el docente.

En la **sesión #7**, con el tema **“Creación de historias de usuario”**, se ofreció una introducción sobre qué son las historias de usuario y cómo elaborarlas. Se socializó la actividad A1-1, que consistió en la creación de historias de usuario. Posteriormente, se realizó un taller en el que cada estudiante desarrolló diez historias de usuario.

En la **sesión #8**, con el tema **“Introducción a la ingeniería de requerimientos”**, y tras haber comprendido teórica y prácticamente el concepto de las historias de usuario, se impartió una introducción a la ingeniería de requerimientos. Se enfatizó el rol del *Product Owner* y la importancia de crear un ambiente adecuado para interactuar con el usuario y recopilar sus necesidades sobre el sistema que se desea desarrollar. Además, se socializó la actividad B1-1, la cual fue asignada como tarea.

En la **sesión #9**, con el tema **“Análisis de levantamiento de requerimientos”**, se revisaron los requerimientos recopilados por cada grupo y se contrastaron con las historias

de usuario. Durante esta sesión, se realizaron observaciones, se corrigieron errores y se descartaron aquellas historias de usuario que no estaban bien elaboradas.

Con las historias de usuario ya revisadas, en la **sesión #10**, los estudiantes asistieron al laboratorio de computación con el tema “**Metodología ágil Scrum**”. Se ofreció una introducción más profunda sobre la metodología ágil Scrum.

En la **sesión #11**, con el tema “**Inspección de interfaces**”, los estudiantes, con sus respectivas guías de laboratorio, inspeccionaron una por una las interfaces de los módulos del sistema de información SICA.

En la **sesión #12**, con el tema “**Creación de prototipos**”, los estudiantes ya tienen claro lo que el usuario espera del sistema. El docente dicta una introducción sobre la creación de prototipos (interfaces de usuario), destacando que la mejor forma de iniciar es realizándolos a mano. Además, se socializa la actividad **B1-2** y se realiza un taller en el que, trabajando en grupo, los estudiantes diseñan varias interfaces de usuario basadas en los requerimientos obtenidos en la actividad **B1-1**.

En la **sesión #13**, con el tema “**Requerimientos vs prototipos**”, se identificaron fallencias en los prototipos creados por los estudiantes. Durante esta sesión, se enfatizó la importancia de la simplicidad en el diseño, tomando como referencia las interfaces del sistema de información SICA ya desarrollado. Asimismo, se estableció un sistema de codificación para cada interfaz creada.

En la **sesión #14**, con el tema “**Dramatización de la metodología Scrum**”, se revisaron videos de trabajos realizados por ciclos

anteriores, analizando sus fortalezas y debilidades. Además, se socializó la actividad **A1-2** y se realizaron pruebas previas para la dramatización.

En la **sesión #15**, con el tema “**Ejecución de la dramatización de la metodología Scrum**”, los estudiantes buscaron el mejor escenario para llevar a cabo la dramatización de la metodología ágil.

En la **sesión #16**, con el tema “**Ejecución de la dramatización Scrum**”, los estudiantes continuaron con la dramatización en un entorno previamente seleccionado y preparado por ellos mismos.

En la **sesión #17**, con el tema “**Documentación de requerimientos**”, se presentó un documento de requerimientos funcionales y no funcionales correspondiente al módulo **Sexo**. Los estudiantes debieron replicar este documento adaptándolo al módulo que les fue asignado.

En la **sesión #18**, con el tema “**Revisión de documentación de requerimientos**”, se revisaron los documentos entregados, destacando sus debilidades y fortalezas. Además, se solicitó a los estudiantes que realizaran las correcciones necesarias en caso de ser requerido.

En la **sesión #19**, con el tema “**Arquitectura del software Modelo-Vista-Controlador y frameworks**”, se impartió en el aula una introducción a la arquitectura de software Modelo-Vista-Controlador (MVC) y a los principales frameworks que utilizan esta arquitectura.

En la **sesión #20**, con el tema “**Socialización de artículo científico: actividad C2-1**”, se socializó un artículo sobre el Modelo-Vista-Controlador (MVC) y los diferentes tipos de

frameworks que utilizan el lenguaje PHP. Esta actividad sirvió como base para la socialización de la actividad de aprendizaje autónomo **C2-1**.

En la **sesión #21**, se trabajó en el laboratorio de computación con el tema **“Uso de GitHub y repositorios”**. Con la guía de laboratorio en mano, los estudiantes crearon sus cuentas en la plataforma GitHub, generaron un repositorio y cargaron el módulo que les fue asignado.

En la **sesión #22**, se trabajó en el aula con el tema **“Práctica de uso de GitHub, Git y repositorios”**. Durante esta sesión, cada estudiante creó una cuenta en GitHub, configuró un repositorio y cargó el módulo de SICA asignado a su grupo.

En la **sesión #23**, los estudiantes trabajaron en el laboratorio con el tema **“Despliegue del módulo SICA”**, realizando el despliegue utilizando la guía de laboratorio.

En la **sesión #24**, se continuó trabajando en el laboratorio de computación con el tema **“Práctica de despliegue del módulo SICA”**, donde los estudiantes siguieron realizando el despliegue.

En la **sesión #25**, se trabajó en el aula de clase con el tema **“Creación de README para el módulo SICA”**. Durante la sesión, se explicó cómo realizar mejoras y mantenimiento al módulo asignado a cada grupo, y se socializó la actividad **A2-1** para ser enviada.

En la **sesión #26**, se trabajó en el laboratorio de computación con el tema **“Socialización de diapositivas del módulo SEXO de SICA”**, enfocándose en realizar mejoras y mantenimiento al módulo.

En la **sesión #27**, se trabajó en el aula de

clase con el tema **“Socialización, mantenimiento y revisión de diapositivas del módulo SEXO de SICA”**. Se socializó la actividad **C2-1**, que consistió en la elaboración de un artículo sobre la importancia de documentar el software.

En la **sesión #28**, se trabajó en el laboratorio de computación con el tema **“Socialización de la actividad C2-2: artículo científico”**, donde, con la guía del laboratorio, los estudiantes construyeron el README del módulo asignado.

En la **sesión #29**, se trabajó en el aula de clases con el tema **“Socialización para la exposición del módulo SICA”**. Durante la clase, se revisó el trabajo realizado en la actividad **C2-1**, brindando indicaciones para que este pueda ser mejorado.

En la **sesión #30**, se trabajó en el aula de clases con el tema **“Defensa de grupo: Arquitectura del software del módulo SICA”**. Durante la clase, se revisó la documentación del módulo **Sexo**, la cual detalla la estructura de cada archivo que conforma el módulo, basada en la arquitectura de software **Modelo-Vista-Controlador (MVC)**. Posterior a la explicación, se envió la actividad **B2-1**, que consiste en que cada grupo replique esta documentación utilizando el módulo que le fue asignado.

En la **sesión #31**, se trabajó en el Laboratorio Aula Visual con el tema **“Defensa de grupo: Arquitectura del software del módulo SICA asignado”**. Los grupos de trabajo realizaron la exposición de sus diapositivas correspondientes a la actividad **A2-2**, las cuales fueron editadas para ser compartidas posteriormente a través del grupo de WhatsApp.

Finalmente, en la **sesión #32**, se abordó en el

aula el tema “Evaluación final sumativa”, durante el cual se llevó a cabo una evaluación diseñada para medir y valorar de manera integral los conocimientos, habilidades y competencias adquiridas por los estudiantes a lo largo de todo el proceso de aprendizaje.

3.1. Encuesta de satisfacción

En la afirmación ¿Qué tan útiles consideras las actividades realizadas para comprender los conceptos clave del ciclo de vida del software?, la categoría de respuesta “Muy útiles” y “Útiles” lideran las menciones (68 % y 30 %) en tanto que la opción “Poco útiles” obtiene un bajo porcentaje (2 %). En la afirmación ¿Cómo calificarías la dinámica de trabajo en equipo utilizando los roles de Scrum?, la categoría de respuesta “Muy efectiva” y “Efectiva” obtuvieron (56 % y 40 %) tanto que la opción “Poco efectiva” e “Inefectiva” (2 % y 2 %). En la afirmación ¿Qué tan claro fue el aprendizaje sobre la metodología ágil Scrum después de la dramatización?, las menciones se concentran en las respuestas “Muy claro” y “Claro” con (50 % y 44 %), mientras que la respuesta “Poco claro” registra el 6 % restante.

En la afirmación ¿Te resultó práctico interactuar directamente con el usuario para el levantamiento de requerimientos?, las menciones se concentran en las categorías de respuesta “Muy práctico” y “Práctico” con 54 % y 36 % respectivamente, mientras que la respuesta “Poco práctico” registra 6 % y “No práctico” 4 %.

En la afirmación ¿Qué tan valiosa fue la experiencia de desarrollar prototipos de interfaces web?, las menciones se concentran

en las categorías de respuestas “Muy valiosa” y “Valiosa” con 50 % y 46 % respectivamente, mientras que la respuesta “Poco valiosa” obtuvo el 4 % restante.

En la afirmación ¿Cómo evalúas la presentación de proyectos en la feria científica y casa abierta?, las menciones se concentran en las categorías “Muy satisfactoria” y “Satisfactoria” con 44 % y 50 % respectivamente, mientras que la respuesta “Poco satisfactoria” obtuvo el 6 % restante.

En la afirmación ¿Qué tan accesible fue el uso de GitLab para cargar el módulo MVC?, las menciones se concentran en las categorías “Muy accesible” y “Accesible” con 45 % y 44 % respectivamente, mientras que la respuesta “Poco accesible” obtuvo el 10 %.

En la afirmación ¿Qué tan atractivo consideraste el uso de la técnica Chroma Key para la presentación virtual sobre la arquitectura MVC?, las menciones se concentran en las categorías “Muy atractivo” y “Atractivo” con 32 % y 54 % respectivamente, mientras que las respuestas “Poco atractivo” y “Nada atractivo” obtuvieron 12 % y 2 %. En la afirmación ¿Prefieres aprender mediante actividades prácticas y proyectos como los realizados, en comparación con solo recibir contenidos y evaluaciones tradicionales?, las menciones se concentraron en las categorías “Definitivamente prefiero las actividades prácticas” y “Prefiero más las actividades prácticas, pero también un poco de teoría tradicional” con 50 % y 50 % respectivamente.

En la afirmación ¿Sientes que el aprendizaje basado en proyectos es más efectivo para tu formación que las evaluaciones tradicionales?, las menciones se concentran

en las categorías “Totalmente de acuerdo” y “De acuerdo” con 44 % y 44 % respectivamente, en cambio la respuesta “En desacuerdo” obtuvo un 12 %.

En la afirmación Consideras que las actividades realizadas te preparan mejor para los desafíos profesionales que los métodos tradicionales?, las menciones se concentran en las categorías “Totalmente de acuerdo” y “De acuerdo” con 48 % y 52 % respectivamente.

En la afirmación “En general, ¿qué tan satisfecho(a) estás con las actividades implementadas en esta asignatura?”, las menciones se concentran en las categorías “Muy satisfecho(a)” y “Satisfecho(a)” con 32 % y

66 % respectivamente, mientras que la respuesta “Poco satisfecho(a)” obtuvo 2 %.

En la afirmación ¿Recomendarías esta metodología a otros estudiantes de la carrera?, las menciones se concentran en las categorías “Definitivamente si” y “Probablemente si” con 60 % y 40 % respectivamente.

3.2. Respuesta a pregunta abierta

Con la finalidad de dar la oportunidad de que los estudiantes pudieran expresarse de forma libre y opiniones que aporte a un proceso de mejoras se dejó una pregunta abierta cuyos resultados se resumen en el siguiente cuadro.

Cuadro 4: Resumen de respuesta a pregunta abierta sobre el proceso de enseñanza aprendizaje de la asignatura Ingeniería de Software I

Categoría	Fortaleza	Debilidad
Conexión teoría-práctica	<ul style="list-style-type: none"> ■ Permitió aplicar conceptos teóricos a problemas reales. ■ Experiencia fue valiosa y abordo todas las etapas del ciclo de vida del software. 	<ul style="list-style-type: none"> ■ Necesidad de más actividades prácticas que reflejen retos reales de la carrera.
Trabajo en equipo	<ul style="list-style-type: none"> ■ Fomenta la colaboración y resolución de problemas reales. 	<ul style="list-style-type: none"> ■ Falta claridad en la asignación de roles dentro de los equipos. ■ Evaluación de contribuciones individuales podría ser más justa. ■ Mejora de la comunicación entre los miembros del equipo.

Categoría	Fortaleza	Debilidad
Retroalimentación	<ul style="list-style-type: none"> ■ Retroalimentación existente es valiosa. 	<ul style="list-style-type: none"> ■ Necesidad de retroalimentación más frecuente durante el desarrollo de los proyectos. ■ Evaluación del proceso, no solo de los resultados finales.
Herramientas y recursos	<ul style="list-style-type: none"> ■ Uso de metodologías ágiles como Scrum y herramientas como GitLab para colaboración. ■ Beneficio de frameworks explicados en clase. 	<ul style="list-style-type: none"> ■ Integración insuficiente de herramientas actuales de desarrollo.
Organización	<ul style="list-style-type: none"> ■ Metodología bien estructurada en general. 	<ul style="list-style-type: none"> ■ Falta de organización en los proyectos. ■ Mejora en la planificación y priorización de tareas. ■ Necesidad de guías más detalladas y definiciones claras de objetivos.
Dinamismo y motivación	<ul style="list-style-type: none"> ■ Actividades interesantes como dramatización y participación en ferias. 	<ul style="list-style-type: none"> ■ Actividades podrían ser más dinámicas para mantener el interés. ■ Algunos estudiante requieren mayor compromiso y empeño en las actividades.
Aplicación real	<ul style="list-style-type: none"> ■ La ABP permitió experimentar desafíos reales como requerimientos cambiantes y trabajo con cliente ficticios. 	<ul style="list-style-type: none"> ■ Poca personalización de metodologías a las necesidades del equipo.

Categoría	Fortaleza	Debilidad
Evaluación	<ul style="list-style-type: none"> ■ Proyectos facilitan aprendizaje práctico. 	<ul style="list-style-type: none"> ■ Falto una evaluación más profunda del proceso y habilidades blandas (liderazgo, trabajo en equipo).
Infraestructura	<ul style="list-style-type: none"> ■ El uso del laboratorio benefició el aprendizaje paso a paso. 	<ul style="list-style-type: none"> ■ El aprovechamiento del laboratorio puede haber sido mejor.
Formación Inicial	<ul style="list-style-type: none"> ■ Introducir la metodología en los primeros semestres construye una base sólida para el aprendizaje. 	<ul style="list-style-type: none"> ■ Falta de preparación inicial en teoría y habilidades básicas para algunos estudiantes.
Complejidad y desafío	<ul style="list-style-type: none"> ■ Las actividades retadoras fueron muy valoradas. 	<ul style="list-style-type: none"> ■ Algunos de las actividades de aprendizaje debieron ser más complejos para aumentar el aprendizaje.

4. Discusión

La metodología de aprendizaje basada en proyectos (ABP) esta propuesta en el modelo educativo de la UTLVT(Canchingre Bone, 2023) y es una decisión del docente aplicarlo en función de la asignatura dictada por él, en la asignatura de Ingeniería de Software I, ofreció una ventaja significativa frente a métodos tradicionales, al permitir a los estudiantes adquirir experiencia práctica y enfrentar problemas reales. Sin embargo, algunos desafíos incluyen la gestión del tiempo y la necesidad de recursos adicionales para implementar estas actividades. Las respuestas a la pregunta abierta destacaron la necesidad de mejorar las actividades de aprendizaje prácticas. Uno de los puntos

críticos señalados fue que el entorno en el que se ejecutaron estas actividades no reflejaba de manera suficiente un contexto realista, ya que se desarrollaron exclusivamente dentro del ámbito académico. Esto impidió que algunos participantes percibieran el entorno como una simulación de situaciones de la vida real.

Además, se identificó la necesidad de un trabajo más profundo en la interiorización y adopción de los roles asignados a los integrantes, ya que no todos lograron asumirlos como era esperado. También se enfatizó la importancia de realizar evaluaciones más precisas y equitativas para cada integrante. En este sentido, una evaluación continua a lo largo del desarrollo de las actividades sería fundamental para garantizar un seguimiento adecuado del desempeño.

Por último, se destacó la relevancia de utili-

zar laboratorios como parte del proceso, ya que estos proporcionan un entorno más técnico y especializado que puede acercar la experiencia académica a un contexto profesional real.

Uno de los grandes desafíos consiste en la generación de competencias esenciales para realizar investigación, según (de Madrid, 2020), estas son Identificar problemas o situaciones problemáticas que requieren investigación, Estructurar el problema, Teorizar acerca de posibles soluciones, Escoger una metodología para investigar alternativas de solución, Generar evidencias con base en la investigación, Analizar información o datos, Utilizar pensamiento inductivo e hipotético-deductivo, Formular inferencias y conclusiones mediante un proceso de investigación con rigor científico.

5. Conclusiones y Recomendaciones

La experiencia obtenida al aplicar la metodología basada en proyecto en la asignatura de Ingeniería de Software I ha demostrado, que el aprendizaje de los estudiantes se ha beneficiado enormemente de un enfoque práctico y basado en proyectos. Se recomienda extender esta metodología a otras asignaturas técnicas y fortalecer la formación inicial de los estudiantes en habilidades básicas. Asimismo, se sugiere implementar evaluaciones continuas y proporcionar recursos adicionales para optimizar los resultados y formar profesionales más preparados para el entorno laboral.

Referencias

- Becker, K., Parker, G., y Al-Fedaghi, S. (2012). *Modeling and simulation in the systems engineering life cycle*. Springer.
- Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, 21(5), 61–72.
- Canchingre Bone, M. E. (2023). *Modelo educativo por competencias con el enfoque constructivista - investigativo - intercultural*. inblue.
- CES. (2017). *Informe técnico de evaluación de la propuesta de rediseño curricular de la carrera de "tecnología de la información" presentada por la universidad técnica luis vargas torres de esmeraldas*. Descargado de <https://repositorioutlvte.org/Repositorio/2024-08-27-FQSA-02293.pdf> (Último acceso: 31 de diciembre de 2024)
- Dabbagh, N. (2005). Pedagogical models for e-learning: A theory-based design framework. *International Journal of Technology in Teaching and Learning*, 1(1), 25–44.
- de Madrid, U. P. (2020). Aprendizaje basado en investigación. *Servicio de Innovación Educativa-UPM*.
- Felder, R. M., y Brent, R. (2005). Understanding student differences. *Journal of Engineering Education*, 94(1), 57–72.
- Fitzgerald, B. (2006). Agile software development: Challenges and implications. *Communications of the ACM*, 49(10), 25–30.
- Fowler, M. (2004). *Uml distilled: A brief*

- guide to the standard object modeling language*. Addison-Wesley Professional.
- Hughes, B., y Cotterell, M. (2017). *Software project management* (7th ed.). McGraw-Hill Education.
- Kuz, A. (2021). Scrum: A new framework applied to education. *Revista Eduweb*, 15(3), 10–17. Descargado de <https://doi.org/10.46502/issn.1856-7576/2021.15.03.1> doi: 10.46502/issn.1856-7576/2021.15.03.1
- on Information Technology Curricula, T. G. (2017). *Information technology curricula 2017: Curriculum guidelines for baccalaureate degree programs in information technology*. Association for Computing Machinery (ACM) and IEEE Computer Society (IEEE-CS). Descargado de <https://www.acm.org/binaries/content/assets/education/it2017.pdf> (A Report in the Computing Curricula Series)
- Pressman, R. S. (2014). *Software engineering: A practitioner's approach* (8th ed.). McGraw-Hill Education.
- Prince, M. (2004). Does active learning work? a review of the research. *Journal of Engineering Education*, 93(3), 223–231.
- Salmon, G. (2000). E-moderating: The key to teaching and learning online. *Kogan Page*.
- Schwab, K. (2017). *The fourth industrial revolution*. Crown Publishing Group.
- Smith, K. A. (2019). Project-based learning in engineering education. *Journal of Engineering Education*.
- SOMMERVILLE, I. (2005). *Ingenieria de software*. Person Educación S.A.
- Yadav, A., Subedi, D., Lundeberg, M. A., y Bunting, C. F. (2011). *Problem-based learning: Influence on students' learning in an electrical engineering course*. Elsevier.