

Tipo de artículo: Artículo original

# Desafíos que enfrentan los estudiantes en la integración de software y funcionalidades: un análisis desde la perspectiva de los participantes en el proyecto FCI-016-2019

## *Challenges faced by students in the integration of software and functionalities: an analysis from the perspective of the participants in the FCI-016-2019 project*

Elsy Rodríguez Revelo <sup>1\*</sup> , <http://orcid.org/0000-0003-4486-0785>

Roberto Gabriel Huacón Lozano <sup>2\*</sup> , <https://orcid.org/0009-0002-5028-1392>

<sup>1</sup> Universidad de Guayaquil. Ecuador. Correo electrónico: [elsy.rodriquezr@ug.edu.ec](mailto:elsy.rodriquezr@ug.edu.ec)

<sup>2</sup> Carrera Ingeniería en Sistemas Computacionales. Facultad de Ciencias Matemáticas y Físicas. Universidad de Guayaquil. Ecuador. [rogellgb@gmail.com](mailto:rogellgb@gmail.com)

\* Autor para correspondencia: [elsy.rodriquezr@ug.edu.ec](mailto:elsy.rodriquezr@ug.edu.ec)

### Resumen

La integración de software de sistemas complejos en la tecnología actual es esencial para optimizar recursos, aunque presenta desafíos para sus desarrolladores. Esta investigación tiene como objetivo analizar los desafíos que enfrentan los estudiantes en la integración de software y funcionalidades en la plataforma SECOED V2 del proyecto FCI- 016-2019 de la Universidad de Guayaquil. Se emplea un enfoque cualitativo y cuantitativo, respaldado por estadísticas descriptivas, análisis de contenido e inteligencia artificial para identificar desafíos estudiantiles, describir funcionalidades, analizar obstáculos y proponer soluciones en la integración de software. La metodología es mixta, utilizando diseño ex post facto para analizar relación entre integración y desafíos, con población de estudiantes graduados y de octavo semestre en Ingeniería en Sistemas Computacionales, usando entrevistas y pruebas de conocimiento. Como resultados fundamentales se logra identificar los desafíos en integración de software que afrontan los estudiantes; y se identifican buenas prácticas para mejorar la formación y eficacia en resolución de desafíos, contribuyendo al desarrollo académico y tecnológico al optimizar experiencia de futuros profesionales.

**Palabras clave:** desafíos; funcionalidades; integración de software; integración continua; obstáculos; plataforma SECOED V2

### Abstract

*The integration of complex systems software into today's technology is essential to optimize resources, although it presents challenges for its developers. This research aims to analyze the challenges that students face in the integration of software and functionalities in the SECOED V2 platform of the FCI-016-2019 project of the University of Guayaquil. A qualitative and quantitative approach, supported by descriptive statistics, content analysis and artificial intelligence, is used to identify student challenges, describe functionalities, analyze obstacles and propose solutions in software integration. The methodology is mixed, using ex post facto design to analyze the relationship between integration and challenges, with a population of graduate and eighth semester students in Computer Systems Engineering, using interviews and knowledge tests. As fundamental results, it is possible to identify the challenges in software integration that students face; and good practices are identified to improve training and effectiveness in solving challenges, contributing to academic and technological development by optimizing the experience of future professionals.*

**Keywords:** challenges; functionalities; software integration; continuous integration; obstacles; SECOED V2 platform



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

**Recibido: 29/08/2023**  
**Aceptado: 26/11/2023**  
**En línea: 01/12/2023**

## Introducción

En una sociedad cada vez más globalizada y caracterizada por un progreso tecnológico creciente, la integración de sistemas complejos se ha vuelto fundamental para optimizar recursos y ofrecer soluciones eficientes (Lwakatere et al., 2019). La conjunción de diferentes componentes de software en un entorno único potencia el funcionamiento integral de una plataforma y permite a los usuarios acceder a diversas funcionalidades de manera coordinada (Saidani et al., 2021). Sin embargo, el éxito de esta integración no es siempre una tarea sencilla, y la falta de un enfoque adecuado puede dar lugar a desafíos significativos (Liu et al., 2020).

La integración de un sistema complejo permite que cada uno de los componentes de software trabajen en forma conjunta para así optimizar recursos (Vassallo, 2019). Sin embargo, cuando ésta es deficiente, puede traer consecuencias para los futuros participantes en la unificación de un proyecto (Zafar et al., 2017). Lo que exige que los responsables cuenten con los conocimientos necesarios de herramientas y técnicas para la incorporación de nuevos módulos con el fin de evitar fallas que comprometan la calidad final del software (Amid et al., 2020).

En el ámbito del desarrollo de software resulta fundamental tener un nivel de conocimiento bastante alto con el fin de alcanzar los objetivos establecidos en la ejecución de un proyecto y superar cualquier obstáculo o desafío que pueda surgir. Por esta razón Moll, Ammerlaan hace hincapié que el personal que se encarga de realizar la integración del sistema con las respectivas pruebas, debe ser poseedor de suficiente conocimiento para hacer un primer diagnóstico cuando surjan problemas de integración (Van Moll & Ammerlaan, 2008). En el entorno de software se habla de conocimientos técnicos sobre el producto y las tecnologías utilizadas. El integrador debe ser capaz de dar una solución temporal ya que posee conocimiento sobre la aplicación. Esto es necesario debido a que con frecuencia se pierde tiempo en notificar a los desarrolladores originales y hacer que entren en acción para hacer un diagnóstico de los problemas que se plantean (Gurcan & Cagiltay, 2019).

En este contexto, los proyectos de integración de software y funcionalidades desempeñan una función fundamental en el avance de las plataformas tecnológicas. Dichos proyectos no solo requieren conocimientos técnicos sólidos, sino también, la capacidad de abordar problemas complejos y superar obstáculos específicos (Rafique et al., 2020). En el ámbito académico, donde los estudiantes participan activamente en proyectos de este tipo, surge la necesidad de entender y abordar los desafíos que enfrentan al realizar esta integración (Bertolino et al., 2020).



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional**  
(CC BY 4.0)

La presente investigación se enfoca en explorar a fondo los desafíos que los estudiantes experimentan al llevar a cabo la integración de software y funcionalidades en sistemas complejos, con un enfoque particular en la plataforma SECOED V2. Esta plataforma, desarrollada en el contexto del proyecto FCI-016-2019, presenta una serie de características y requerimientos que hacen que su integración sea un proceso significativo y a la vez desafiante (Macías Zambrano & Huebla Lema, 2023).

El proyecto FCI-016-2019, cuyo propósito es avanzar con la implementación de una plataforma que permita administrar, gestionar y sistematizar los procesos de formación docente de una manera adecuada, pertenece a la Facultad de Ciencias Matemáticas y Físicas de la Universidad de Guayaquil (Figuroa Zúñiga & Medina Baquero, 2022). El proyecto FCI-016-2019, cobra relevancia en el ámbito de la Educación Superior, dado que posibilita la participación activa de los estudiantes de las carreras tecnológicas para que ellos potencien sus competencias como desarrolladores. En este sentido es importante conocer los inconvenientes que se presentan cuando se debe trabajar con integraciones en sistemas cuya estructura es altamente compleja.

La plataforma SECOED V2 se encuentra en proceso de prueba, y ha sido desarrollada por los estudiantes, los cuales le han integrado módulos en diferentes periodos académicos. Es decir que durante el periodo de titulación los estudiantes avanzan con el desarrollo de una parte específica de la plataforma, que luego se tiene que unificar al proyecto final (Macías Zambrano & Huebla Lema, 2023).

El objetivo principal de este estudio es comprender las dificultades que los estudiantes encuentran en el proceso de integración de software y funcionalidades en la plataforma SECOED V2, desde la perspectiva de los participantes en el proyecto FCI-016-2019. Al explorar y analizar a fondo estas dificultades, se busca proponer soluciones y recomendaciones que permitan mejorar la experiencia y el rendimiento de los estudiantes en este proceso (Bayona Rojas & Pérez Tigrero, 2022).

A través de este estudio, se busca contribuir al ámbito académico y tecnológico al identificar áreas de mejora y ofrecer perspectivas valiosas para la formación y desarrollo de estudiantes en proyectos de integración de software. La optimización de este proceso no solo beneficiará a los futuros profesionales en el campo, sino que también contribuirá al avance de la tecnología y la innovación en sistemas complejos.

## **Materiales y métodos**

En la búsqueda de soluciones, se utilizarán enfoques de investigación cualitativa y cuantitativa y se emplean instrumentos de análisis de contenido y métodos estadísticos descriptivos. El enfoque mixto de investigación se caracteriza por la utilización de una metodología que busca la fusión y colaboración de los métodos cualitativos y cuantitativos dentro del mismo estudio. Este enfoque se sustenta en la premisa de que ambos enfoques pueden



complementarse y enriquecerse mutuamente, siendo capaces de responder a distintas cuestiones y objetivos de investigación. Este enfoque puede adoptar diversos diseños en función del grado de integración deseado, el momento en que se lleva a cabo la recopilación y análisis de los datos, así como el peso relativo que se le otorga a cada método. El estudio se desarrolló en dos momentos, esto implicó que en cada uno se abordara con un tipo de población con características particulares. Esta segmentación en dos grupos permitió una observación más detallada y focalizada. Cada uno de estos posee atributos específicos que los distinguen y confieren un análisis significativo. El enfoque en estos grupos distintos resulta crucial para entender las diferencias y similitudes en sus respuestas y decisiones frente a diferentes estímulos o situaciones. Al considerar las particularidades de cada parte de la población, se obtienen conclusiones más sólidas y valiosas, lo que aporta una visión más profunda y enriquecedora.

Los grupos segmentados en esta población son:

- Grupo 1: El primer grupo poblacional fue de 30 estudiantes ya graduados de la Carrera de Ingeniería en Sistemas Computacionales de la Facultad de Ciencias Matemáticas y Físicas que participaron de los procesos de integración en la Plataforma SECOED V2, entre el semestre CII-2019 hasta el semestre CII 2022-2023, los cuales participaron con un tema que fue desarrollado como trabajo de titulación. Sus edades oscilan entre los 24 y 41 años.
- Grupo 2: El segundo grupo lo conformaron estudiantes de octavo semestre del presente ciclo CI- 2023-2024 de la Carrera de Ingeniería en Sistemas Computacionales. El total de estudiantes en este grupo es de 70 estudiantes que se encuentran inscritos en el horario nocturno y sus edades oscilan entre 23 y 33 años y están próximos a buscar y proponer temas de titulación o participar en proyectos de investigación o FCI impulsados por investigadores de la Facultad de Ciencia Matemáticas y Físicas.

Se optó por utilizar una muestra no probabilística. La selección de esta muestra fue de carácter intencional y por conveniencia, ya que permitió un mayor dominio en la elección de los involucrados. En situaciones donde la población es pequeña, resulta más práctico y eficiente seleccionar a los sujetos de estudio de manera intencionada. La utilización de una muestra intencional posibilitó la incorporación de aquellos estudiantes que estuvieran disponibles y deseosos de participar en la investigación, minimizando el riesgo de rechazos y agilizando la recolección de datos. Además, se procedió a obtener datos y contactos de los estudiantes de octavo semestre de este ciclo académico a través del director de la Carrera de Ingeniería en Sistemas Computacionales, quien otorgó su aprobación para llevar a cabo la recopilación de información.



## Resultados y discusión

La integración de software y funcionalidades en sistemas complejos es un proceso crítico y desafiante en la creación de entornos educativos digitales como la plataforma SECOED V2. Los estudiantes que participan en este proceso pueden enfrentar obstáculos que podrían dificultar su desempeño y su capacidad para lograr los objetivos del proyecto. Por lo tanto, resulta crucial reconocer y examinar estos desafíos desde la perspectiva de los participantes en el proyecto, con el fin de proponer soluciones o recomendaciones que permitan mejorar la experiencia y el rendimiento de los estudiantes en este proceso.

En las entrevistas iniciales se pudo conocer que el factor crítico se presenta cuando los estudiantes inician su participación en el proyecto, en donde las tareas son asignadas para que sean ejecutadas durante el tiempo que dure el proceso de titulación. Una vez cumplida con la actividad se debe integrar este nuevo módulo en la plataforma, sin embargo, en este punto se detectan los grandes inconvenientes. Por ejemplo, algunos estudiantes no dominan ciertos programas que se utilizan o en un caso extremo no saben cómo utilizar alguna herramienta y por ende no saben qué hacer. Además, una vez que se han desarrollado las tareas a partir de las indicaciones recibidas, deben trasladar todos los componentes de software a un ambiente local para su integración. Como resultado, se extiende el tiempo que se tenía programado para finalizar el desarrollo ya que en algunos casos ciertos módulos presentaron grandes fallas que exigió modificarlos en su totalidad.

Basado en estos resultados preliminares, y en la investigación bibliográfica realizada por los autores, fue posible crear la Matriz de causas y consecuencias del problema de integración de software que enfrentan los estudiantes. En la Tabla 1 se presentan los resultados alcanzados.

**Tabla 1.** Matriz de causas y consecuencias del problema.

Causas	Consecuencias
C1. Falta de preparación del estudiante en la integración de software y funcionalidades de sistemas complejos.	E1. Complejidad al momento de hacer las integraciones en sistemas complejos.
C2. Relevo de participantes en la integración software y funcionalidades de sistemas complejos.	E2. Adaptación en la continuidad de los proyectos.
C3 Poco uso de técnicas de integración en sistemas complejos.	E3. Se presentan errores en el proceso de desarrollo.
C4. Módulos mal parametrizados.	E4. Perjudica la experiencia de los que utilizan la plataforma SECOED V2.
C5. Falta de experiencia en el manejo de sistemas complejos.	E5. Tiempo prolongado para presentar resultados.



## Aplicación de instrumentos

### 1. Encuesta a egresados del proyecto FCI-016-2019

Con el objetivo de identificar los desafíos que enfrentan los estudiantes en la integración de software y funcionalidades en la plataforma SECOED V2, se aplicó una encuesta a un grupo de egresados del proyecto FCI-016-2019. Sin embargo, se consideraron 30 estudiantes que participaron en el proyecto FCI-016-2019 de los cuales solo 10 respondieron la encuesta. Con esta encuesta, se pretende identificar, desde la perspectiva de estos profesionales, cuáles fueron los principales inconvenientes que enfrentaron en su etapa como estudiantes vinculados al desarrollo. A continuación se resumen las respuestas otorgadas por los 10 egresados del proyecto FCI-016-2019 que participaron en la encuesta:

**Pregunta 1:** ¿Usted contaba con formación y experiencia previa en la integración de software y funcionalidades en plataformas antes de participar como desarrollador en el proyecto SECOED V2?

**Pregunta 2:** ¿Usted recibió cursos o formación adicional relacionada con la integración de software y funcionalidades?



**Figura 1.** Formación y experiencia previa en la integración de software antes de participar en el proyecto SECOED V2.



**Figura 2.** Participación en cursos o formación adicional relacionada con la integración de software.

### Análisis de los resultados:

Referente a la pregunta 1, los datos obtenidos manifiestan que la mayoría de los egresados no tenían formación ni experiencia previa en la integración de software y funcionalidades en plataformas; 3 de los participantes tenían tanto formación como experiencia. Solo 1 participante tenía experiencia previa y otro mencionó tener formación en el área. Estos resultados indican que la mitad de los entrevistados carecía de formación y experiencia, lo que podría requerir apoyo adicional en el proyecto. Sería importante considerar proporcionar capacitación o información introductoria para nivelar el conocimiento del equipo. En general, hay diversidad en los niveles de formación y experiencia de los estudiantes. Esto puede ser una oportunidad para aprovechar los conocimientos de los experimentados, mientras se brinda apoyo a los menos experimentados en la integración de software y funcionalidades en plataformas.



Referente a la pregunta 2, los resultados muestran que la mayoría de los participantes no contó formación adicional en integración de software y funcionalidades, lo que destaca la importancia de brindar capacitación continua y recursos educativos. En este sentido, los estudiantes con formación extra podrían desempeñar un papel clave en compartir conocimientos y apoyar al equipo en el desarrollo de la plataforma.

**Pregunta 3:** ¿Considera que los conocimientos y habilidades en programación fueron en su momento suficientes para enfrentar los desafíos de la integración en el proyecto SECOED V2?



**Figura 3.** Consideraciones sobre los conocimientos y habilidades en programación relacionada con la integración de software.

**Análisis de los resultados:** Según los datos, el 40% cree que sus habilidades en programación fueron suficientes; el resto de los encuestados se comportó de la siguiente manera: un 10% no lo considera así pues deseaba más experiencia en el *framework*; 10% no tenía conocimientos en el lenguaje de programación; 10% necesitó investigar más; 10% carecía de habilidades, pero buscó aprender para el proyecto; 20% considera sus habilidades en programación no eran suficientes. Las respuestas variadas de los egresados reflejan diferentes niveles de confianza en sus habilidades de programación para los desafíos de integración en el proyecto SECOED V2. Algunos se sintieron adecuadamente preparados, mencionando sus conocimientos y la capacidad para enfrentar los desafíos. Otros reconocieron que, a pesar de tener conocimientos, necesitaron aprender más para cumplir con los requisitos específicos. La falta de experiencia en *frameworks* sugiere dificultades con tecnologías específicas.

**Pregunta 4:** ¿Cuáles fueron los principales desafíos que usted enfrentó en la integración de software y funcionalidades en la plataforma SECOED V2?

En la tabla 1 se presentan los desafíos que más mencionaron los encuestados.



**Tabla 1.** Desafíos enfrentados en la en la integración de software y funcionalidades en la plataforma SECOED V2.

No.	Desafío	Frecuencia
1	Asignación de responsabilidades poco claras y falta de estrategia de integración.	30%
2	Cambios en los requisitos tecnológicos y funcionales.	70%
3	Comprensión inadecuada de los requisitos y las cuestiones de interfaz.	70%
4	Compartir versiones no probadas de los componentes.	80%
5	Detección tardía de defectos en los componentes.	80%
6	Dejar la detección de defectos para la etapa de integración.	90%
7	Dependencias inadecuadas de las tareas relacionadas con las propiedades sistémicas.	60%
8	Dificultad para garantizar la interoperabilidad entre aplicaciones.	50%
9	Falta de coordinación entre equipos distantes.	60%
10	Falta de gestión continua y activa de la arquitectura de software.	40%
11	Falta de integración documentos y manuales.	30%
12	Falta de los conocimientos y competencias necesarios en materia de integración.	90%
13	Falta de recursos técnicos y humanos.	30%
14	Incompatibilidad entre diferentes tecnologías y sistemas.	30%
15	Predicciones poco realistas en el plan de integración.	10%
16	Requisitos que cambian rápidamente e interdependencias técnicas inesperadas.	70%
17	Requisitos de los módulos mal comunicados.	90%
18	Software mal diseñado.	70%
19	Uso de diferentes versiones herramientas.	70%

**Análisis de los resultados:** Referente a la pregunta número 4 sobre los desafíos enfrentados en la integración de software y funcionalidades de la plataforma SECOED V2 por parte de los egresados del proyecto FCI-016-2019, se pudo conocer como principales desafíos que el 90% de los egresados tuvo que enfrentar desafíos en su etapa estudiantil relacionados con la *“Falta de los conocimientos y competencias necesarios en materia de integración”*, esto indica que la mayoría de los estudiantes comenzó las actividades de la etapa de integración sin tener la capacitación necesaria para asumir este rol dentro del proceso de desarrollo del software; este mismo porcentaje describió las consecuencias enfrentadas por los *“Requisitos de los módulos mal comunicados”*, con esto se refieren a que no contaban con una matriz de trazabilidad de requisitos que indicara la dependencia que tenía cada requisito dentro del mismo módulo, o en distintos módulos; estos errores en la trazabilidad de requisitos originan incompatibilidades y complejidades de integración que posteriormente producirán retrasos, costes adicionales, afectan a la calidad e inclusive pueden sumir en

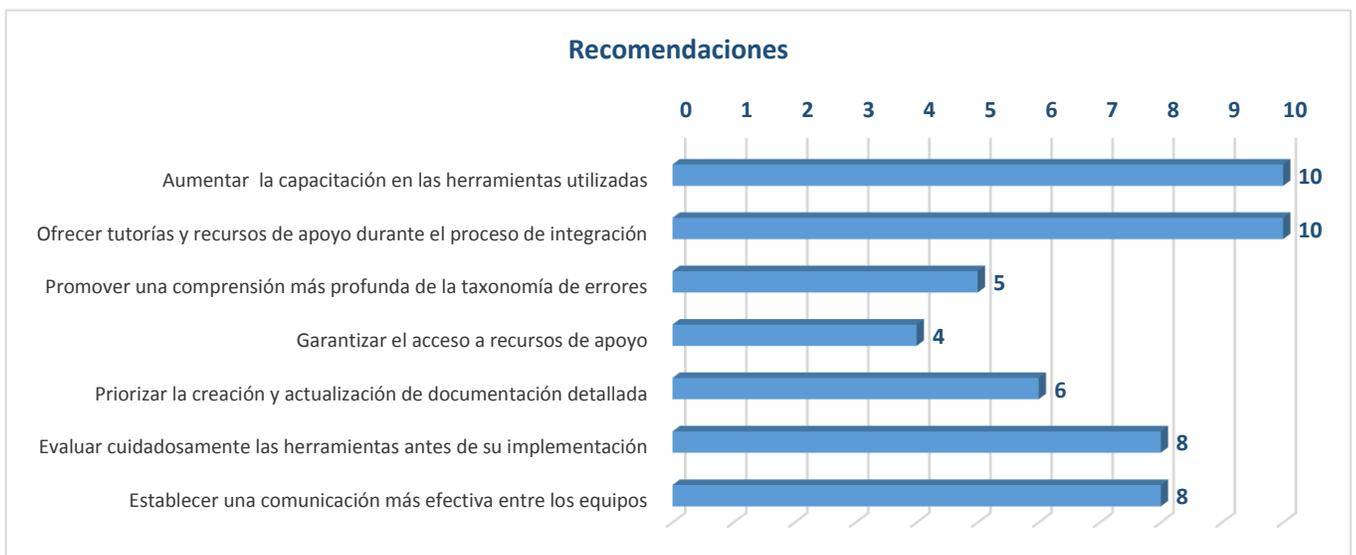


el caos el proyecto. Otro desafío enfrentado fue “Dejar la detección de defectos para la etapa de integración”, esto colapsa los procesos y a los recursos humanos; y retrasa los cronogramas de entrega. Estos problemas ocurren cuando los defectos permanecen ocultos durante la fase de desarrollo, sin embargo, durante la integración del sistema estos emergen.

En este mismo sentido, el 80% de los egresados en algún momento tuvo que afrontar las consecuencias que genera “Compartir versiones no probadas de los componentes”, ya que los componentes no fueron probados de forma individual antes de entrar a la etapa de integración, esto indica que se violaron actividades en la etapa de pruebas. Este mismo porcentaje declara que se realizaba una “Detección tardía de defectos en los componentes”, lo que implicaba una saturación de trabajos a la hora de corregir todos los errores de una vez. En ocasiones, era necesario rehacer el trabajo.

Los resultados muestran que los estudiantes de periodos anteriores comenzaron a realizar tareas de la etapa de integración sin la debida capacitación. Los desafíos identificados constituyen un importante hallazgo, porque a partir de ellos se puede realizar una planificación de capacitación más enfocada en las percepciones y experiencias de los estudiantes.

**Pregunta 5:** ¿Qué soluciones y recomendaciones usted propone para mejorar la integración de software y funcionalidades en la plataforma SECOED V2?



**Figura 4.** Recomendaciones para mejorar la integración de software y funcionalidades en la plataforma SECOED V2.

**Análisis de los resultados:** Todos los egresados encuestados coincidieron en que es recomendable “Mejorar la documentación y la capacitación en las herramientas utilizadas” y “Ofrecer tutorías y recursos de apoyo durante el



*proceso de integración*”. El resto de las recomendaciones se relacionan con la necesidad de capacitar a los estudiantes antes de asignarle tareas de integración; así como la necesidad de brindar acceso y actualización referentes a las herramientas a emplear. En este sentido se recomienda estandarizar los requisitos no funcionales de hardware para garantizar una integración exitosa.

## **2. Test de conocimiento:**

Teniendo en cuenta las experiencias de los estudiantes egresados, identificadas en la encuesta realizada al grupo 1, se aplicó un test de conocimiento con una finalidad diagnóstica, a los estudiantes de octavo semestre que se encuentran legalmente matriculados en la facultad. El test de conocimiento que consta de 20 preguntas. Fueron 70 estudiantes en total a los que se contactó por medio de sus correos personales, institucionales, o por medio de su número de contacto de WhatsApp. Sin embargo, solo 54 estudiantes participaron resolviendo el test. La mayor cantidad de participantes se encuentra entre las edades de 23 y 27 años. El test se enfoca en evaluar el conocimiento y comprensión de la integración de software y funcionalidades en la plataforma SECOED V2. Está diseñado para obtener información sobre diversos aspectos relacionados con la integración de software, desde definiciones y tecnologías utilizadas hasta las prácticas y desafíos que pueden surgir en el proceso. Este consta de varias preguntas que abarcan diferentes áreas de la integración de software.

Los ítems del test de conocimiento incluyen preguntas relacionadas con conceptos básicos de integración, que abarcan fundamentos como la prueba de unidad (*unit testing*), *middleware*, sistemas *legacy* y buenas prácticas en la integración de software y funcionalidades. Asimismo, se explora la comprensión de Metodologías de Integración, como la Integración Continua (CI) y la Arquitectura Orientada A Servicios (SOA), con el objetivo de evaluar la familiaridad del estudiante con estos enfoques y su aplicabilidad.

La información recopilada a partir de este test de conocimiento es valiosa para identificar las áreas de mejora en el conocimiento y experiencia de los estudiantes antes de comenzar a asumir tareas de integración de software y funcionalidades, lo que puede guiar futuras iniciativas de capacitación y desarrollo en este campo. Los resultados del test de conocimiento se tabularon en la Tabla 2. La escala de evaluación fue como sigue: teniendo en cuenta las 16 preguntas técnicas, se evaluó el nivel de conocimiento en una escala de 5 puntos cada una, donde: (1)= Muy Bajo; (2)= Bajo; (3) Medio; (4) Alto; y (5) Muy Alto.



**Tabla 2.** Resultados el test de conocimiento aplicado al grupo 2 de la muestra.

No.	Área del conocimiento relacionada	M	SD
1.	Dominio de los principios y objetivos de la plataforma SECOED V2.	4,02	0,61
2.	Dominio de la arquitectura y funcionalidades de la plataforma SECOED V2.	2,23	1,12
3.	Dominio del objetivo principal de la integración de software y sus funcionalidades	4,48	0,98
4.	Dominio de los principales obstáculos que pueden surgir durante la integración de software y sus funcionalidades	3,63	0,85
5.	Dominio de las tecnologías que comúnmente utiliza en la integración de software	2,44	0,82
6.	Dominio de los conceptos de los sistemas de integración continua	3,52	0,57
7.	Conocimiento del papel del webhook en el contexto de la integración continua	2,48	0,67
8.	Dominio del protocolo SOA en el contexto de la integración continua	2,48	0,61
9.	Conocimiento sobre la migración de datos en el contexto de la de la integración continua	3,22	0,90
10.	Dominio de las herramientas para el proceso de integración de software	3,30	0,82
11.	Conocimiento sobre las pruebas de unidad en el contexto de la integración de software	4,04	1,12
12.	Conocimientos de middlewarer en el contexto de la integración de software	4,46	0,72
13.	Conocimientos de sistemas legacy	2,85	1,14
14.	Conocimiento de las buenas prácticas en la integración de software y funcionalidades	3,39	0,81
15.	Experiencia en el manejo de la taxonomía de errores en la integración de software	1,35	0,87
16.	Dominio de la aplicación de metodologías de Integración.	2,04	1,06
17.	Dominio de los parámetros para evaluar la calidad del software.	3,94	1,19

**M:** Media; **SD:** Desviación estándar

En los resultados obtenidos en el test de conocimiento realizados a estudiantes de quinto año que van a comenzar su trabajo de titulación, se pudo conocer que la mayoría de los estudiantes tienen un nivel de conocimiento Alto sobre los “Principios y objetivos de la plataforma SECOED V2. (M= 4,02)”; sin embargo, los resultados indican que tienen un nivel Bajo en el “Dominio de la arquitectura y funcionalidades de la plataforma SECOED V2 (M=2,23)”. Los mejores resultados se obtuvieron en la caracterización del “Objetivo principal de la integración de software y sus funcionalidades (M= 4, 48)”, donde la mayoría demostró un conocimiento teórico Alto.

El análisis exhaustivo del test de conocimiento ha revelado tendencias significativas que ofrecen una comprensión más profunda de las áreas en las cuales los participantes han demostrado un nivel constante de conocimiento. Estas tendencias arrojan luz sobre las fortalezas y dominios en los que los participantes han logrado una sólida comprensión. En particular, se ha observado que los participantes han demostrado un conocimiento sólido en conceptos esenciales de integración de



software, como la importancia y la aplicación de la prueba de unidad (*unit testing*). Además, se ha destacado una tendencia marcada hacia la comprensión de la función y el propósito de *middleware* en el proceso de integración, así como la valoración generalizada de la implementación de buenas prácticas en la integración de software y funcionalidades.

Un patrón identificado gira en torno a la variabilidad en la comprensión de las metodologías de integración, donde algunos participantes exhiben un entendimiento profundo, mientras que otros muestran una comprensión más superficial. Otro patrón destacado radica en las respuestas limitadas en relación con la evaluación de la calidad del software, lo que sugiere la necesidad de una mayor profundización en esta área específica. Estos patrones ofrecen una visión más matizada de las áreas en las que los participantes podrían requerir mayor orientación y refuerzo. En conjunto, las tendencias y patrones extraídos del análisis del test de conocimiento proporcionan una imagen completa y equilibrada del nivel de competencia de los estudiantes que van a comenzar su proceso de titulación, señalando áreas sólidas y oportunidades de mejora en su comprensión de la integración de software y funcionalidades.

### Buenas prácticas

Teniendo en cuenta los hallazgos resultantes de la encuesta realizada al grupo 1 de la muestra, y los resultados del test de conocimiento alcanzados por el grupo 2 de la muestra, los autores de esta investigación realizaron una revisión de la literatura, para listar las buenas prácticas recomendadas por diversos autores, para fortalecer el proceso de integración de software realizado por los estudiantes que van a comenzar sus proyectos de titulación. La tabla 3 resume, un conjunto de buenas prácticas recopiladas:

**Tabla 3.** Buenas prácticas recomendadas para la integración de software.

No.	Buenas prácticas
1.	Establecer canales de comunicación claros y regulares entre equipos y partes interesadas. Fomentar una cultura de colaboración y comunicación abierta en todos los niveles.
2.	Compartir experiencias exitosas y casos de estudio para enriquecer el conocimiento del equipo.
3.	Controlar, publicar y registrar los fallos de integración en una matriz de errores.
4.	Priorizar la creación y actualización constante de documentación detallada.
5.	Realizar una planificación exhaustiva antes de iniciar el proceso de integración.
6.	Promover la selección adecuada de herramientas a través de evaluaciones minuciosas y pautas claras.
7.	Realizar una evaluación exhaustiva de las herramientas disponibles antes de elegir una.
8.	Invertir en educación y formación continua para mejorar la comprensión de conceptos clave y habilidades técnicas.
9.	Proporcionar asesoría y tutorías para desarrollar habilidades y superar desafíos.
10.	Promover la participación en proyectos prácticos para ganar experiencia.



11.	Establecer programas de mentoría para aprovechar la experiencia de los profesionales más experimentados.
12.	Elegir una estrategia de integración bien definida, basada en un análisis de factibilidad técnica.
13.	Establecer límites bien definidos entre las fases del proceso de diseño, codificación o prueba para dividir el trabajo de modo que pueda asignarse a distintos grupos.
14.	Establecer un mecanismo de comunicación y coordinación eficiente entre los participantes o integrantes del equipo de desarrollo.
15.	Dependiendo de la cantidad de participantes en el proyecto, debería asignarse un equipo o al menos un miembro para supervisar el estado de desarrollo y la confirmación de las normas para permitir una integración sin fisuras.
16.	En la en la planificación del proceso, se debe incluir la planificación de la recopilación de requisitos, las técnicas de análisis y la selección del modelo de software.
17.	Establecer un documento de proceso completo y correcto, que sienta las bases para un desarrollo y una integración eficaces del sistema. Este documento debe ser actualizado tras los cambios de requisitos o diseño.
18.	Crear una matriz de trazabilidad del cambio para todos los artefactos del proyecto (documentación, requisitos, diseño, errores, código fuente, ect.).
19.	Cuando las oficinas externas son los responsables de elaborar una estrategia de integración válida, estos deben conocer todos los cambios realizados en el transcurso del proyecto.
20.	La asignación de tareas sobre una base <i>ad hoc</i> debe realizarse a personas con experiencia o con competencias necesarias, para evitar desencadenar problemas en el momento de la integración.
21.	Deben gestionarse centralmente las propiedades sistémicas, como la huella de memoria y el presupuesto de rendimiento, así como dependencias entre componentes que se construyen en ubicaciones remotas.
22.	Garantizar unas pruebas unitarias y de integración rigurosas para permitir una integración sin fallos.
23.	Es importante tener una buena comunicación personal ya que sin comunicación cara a cara entre equipos remotos, es muy difícil evitar problemas de integración.
24.	Fomentar la colaboración interdisciplinaria para mejorar el proceso de integración.

## Conclusiones

Con la realización de esta investigación se logró identificar una serie de desafíos que los estudiantes enfrentan al realizar la integración de software y funcionalidades en la plataforma SECOED V2. Estos desafíos incluyen: barreras de comunicación; elección inadecuada de herramientas; planificación insuficiente del proceso; problemas técnicos; dificultades en la migración de datos; y comprensión de la taxonomía de errores; falta de información y experiencia; y poco énfasis en los parámetros para evaluar la calidad del software.



Con la encuesta aplicada a egresados del proyecto objeto de estudio, se constató que los estudiantes enfrentan una variedad de desafíos durante el proceso de integración, que van desde cuestiones técnicas hasta problemas de planificación y comunicación. Estos desafíos indican áreas en las que se podrían implementar mejoras para facilitar la integración de software y funcionalidades en la plataforma SECOED V2.

Adicionalmente se logró identificar varias causas raíz de los desafíos en la integración de software y funcionalidades en la plataforma SECOED V2. Estas causas incluyen la falta de documentación y recursos adecuados, la complejidad de las tecnologías utilizadas, la falta de experiencia previa en la resolución de problemas de integración y la falta de comprensión de conceptos clave como la taxonomía de errores. Las causas subyacentes de los desafíos identificados apuntan a la necesidad de abordar problemas de documentación, mejorar la formación en conceptos clave y proporcionar recursos adecuados para que los estudiantes puedan enfrentar los desafíos de la integración de manera más efectiva.

En consecuencia, se ha conseguido reconocer una serie de desafíos inherentes al proceso de integración de software y funcionalidades en la plataforma SECOED V2, y se han presentado buenas prácticas para enfrentar estos desafíos y mejorar el proceso en general.

## Conflictos de intereses

Los autores no poseen conflicto de intereses.

## Contribución de los autores

1. Conceptualización: Roberto Gabriel Huacón Lozano, Elsy Rodríguez Revelo.
2. Curación de datos: Roberto Gabriel Huacón Lozano
3. Análisis formal: Roberto Gabriel Huacón Lozano, Elsy Rodríguez Revelo
4. Investigación: Roberto Gabriel Huacón Lozano, Elsy Rodríguez Revelo
5. Metodología: Elsy Rodríguez Revelo, Roberto Gabriel Huacón Lozano
6. Administración del proyecto: Elsy Rodríguez Revelo
7. Recursos: Roberto Gabriel Huacón Lozano, Elsy Rodríguez Revelo
8. Software: Roberto Gabriel Huacón Lozano, Elsy Rodríguez Revelo
9. Supervisión: Elsy Rodríguez Revelo, Roberto Gabriel Huacón Lozano
10. Validación: Roberto Gabriel Huacón Lozano, Elsy Rodríguez Revelo
11. Visualización: Roberto Gabriel Huacón Lozano, Elsy Rodríguez Revelo



12. Redacción – borrador original: Roberto Gabriel Huacón Lozano, Elsy Rodríguez Revelo

13. Redacción – revisión y edición: Elsy Rodríguez Revelo, Roberto Gabriel Huacón Lozano

## Financiamiento

La investigación no requirió fuente de financiamiento externo.

## Referencias)

- Amid, A., Biancolin, D., Gonzalez, A., Grubb, D., Karandikar, S., Liew, H., Magyar, A., Mao, H., Ou, A., & Pemberton, N. (2020). Chipyard: Integrated design, simulation, and implementation framework for custom socs. *IEEE Micro*, 40(4), 10-21. <https://ieeexplore.ieee.org/abstract/document/9099108/>
- Bayona Rojas, E. X., & Pérez Tigrero, K. B. (2022). *Propuesta de implementación del Sistema “SECOED V2”, que complementa el proceso de evaluación de desempeño docente institucional, incorporando indicadores pedagógicos, didácticos y TIC* Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas ...]. <http://repositorio.ug.edu.ec/handle/redug/59948>
- Bertolino, A., Guerriero, A., Miranda, B., Pietrantuono, R., & Russo, S. (2020). Learning-to-rank vs ranking-to-learn: Strategies for regression testing in continuous integration. Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering,
- Figuroa Zúñiga, A. J., & Medina Baquero, J. A. (2022). *Prototipo de aplicación Web para documentar las fases del proceso de diseño y desarrollo de un proyecto de investigación, FCI-016-2019* Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas ...]. <http://repositorio.ug.edu.ec/handle/redug/65227>
- Gurcan, F., & Cagiltay, N. E. (2019). Big data software engineering: Analysis of knowledge domains and skill sets using LDA-based topic modeling. *Ieee Access*, 7, 82541-82552. <https://ieeexplore.ieee.org/abstract/document/8742548/>
- Liu, C., Jiang, P., & Jiang, W. (2020). Web-based digital twin modeling and remote control of cyber-physical production systems. *Robotics and computer-integrated manufacturing*, 64, 101956. <https://www.sciencedirect.com/science/article/pii/S0736584519300134>
- Lwakatare, L. E., Kilamo, T., Karvonen, T., Sauvola, T., Heikkilä, V., Itkonen, J., Kuvaja, P., Mikkonen, T., Oivo, M., & Lassenius, C. (2019). DevOps in practice: A multiple case study of five companies. *Information and Software Technology*, 114, 217-230. <https://www.sciencedirect.com/science/article/pii/S0950584917302793>



- Macías Zambrano, M. Á., & Huebla Lema, J. A. (2023). *Desarrollo e implementación de modelo de seguridades, módulo de auditorías y reportes de control en la plataforma SECOED V2, FCI 016-2019* Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas ...]. <http://repositorio.ug.edu.ec/handle/redug/68947>
- Rafique, W., Qi, L., Yaqoob, I., Imran, M., Rasool, R. U., & Dou, W. (2020). Complementing IoT services through software defined networking and edge computing: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(3), 1761-1804. <https://ieeexplore.ieee.org/abstract/document/9099866/>
- Saidani, I., Ouni, A., Mkaouer, M. W., & Palomba, F. (2021). On the impact of continuous integration on refactoring practice: An exploratory study on travistorrent. *Information and Software Technology*, 138, 106618. <https://www.sciencedirect.com/science/article/pii/S0950584921000914>
- Van Moll, J., & Ammerlaan, R. (2008). Identifying pitfalls of system integration--an exploratory study. 2008 IEEE International Conference on Software Testing Verification and Validation Workshop,
- Vassallo, C. (2019). Enabling continuous improvement of a continuous integration process. 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE),
- Zafar, A. A., Saif, S., Khan, M., Iqbal, J., Akhunzada, A., Wadood, A., Al-Mogren, A., & Alamri, A. (2017). Taxonomy of factors causing integration failure during global software development. *Ieee Access*, 6, 22228-22239. <https://ieeexplore.ieee.org/abstract/document/8241376/>

