

**Arquitectura de Software: Evaluación del Conocimiento en Estudiantes de Análisis y  
Desarrollo de Software**

David Mauricio Flórez Quintero

Co-Autores

Jesús Ariel Gonzales Bonilla

Jhon Willian Corredor Araujo

Estudiante de Tecnología en Análisis y desarrollo de Software

Servicio Nacional de Aprendizaje SENA

Neiva, Colombia

Noviembre, 2024

Palabras clave: Arquitectura de software, Patrones de diseño, Educación en tecnologías

## **Abstract**

La arquitectura de software es fundamental en el desarrollo de tecnologías modernas mantiene su sostenibilidad, potencial de expansión y adaptabilidad El objetivo de este artículo es evaluar la comprensión que poseen los diseñadores de software con respecto a la arquitectura de software y su competencia en la ejecución de prácticas arquitectónicas y procedimientos ágiles durante aplicaciones del mundo real El enfoque incluye la aplicación de técnicas de diseño, la evaluación de ejemplos de la industria y la incorporación de métodos como Scrum y XP para mejorar la adquisición de conocimientos. Los hallazgos indican que los estudiantes presentan dificultades para vincular conceptos teóricos con su aplicación en la práctica; sin embargo, también subrayan que la inclusión de patrones específicos y métodos didácticos efectivos puede potenciar notablemente su comprensión y rendimiento. En conclusión, este estudio sostiene que la enseñanza de la arquitectura de software debe incorporar estrategias prácticas que permitan a los estudiantes adquirir habilidades relevantes para el ámbito laboral, lo que a su vez asegurará la formación de profesionales competentes y listos para enfrentar los retos tecnológicos actuales.

## **Introducción**

La arquitectura de software se ha vuelto un elemento fundamental en la creación de sistemas contemporáneos, integrando el modelado de negocio, el análisis y el diseño para asegurar una base robusta, sostenible y que pueda crecer con el tiempo (Sánchez Palmero, 2019). Mediante el uso de diversas estructuras y patrones, esta disciplina ofrece un marco de referencia que optimiza la organización y la efectividad en la ejecución de soluciones tecnológicas.

Pero es difícil enseñar y probar la arquitectura de software con estudiantes que aprenden a analizar y construir software. Los futuros desarrolladores deberían aprender tanto las ideas como cómo utilizar los diseños de edificios en proyectos reales.

La intención de este artículo es analizar la familiaridad de los estudiantes con el plan de estudios de Arquitectura de Software y sugerir estrategias pedagógicas para mejorar su experiencia. La importancia de esta investigación radica en el conocimiento del diseño que mejora la eficacia y eficiencia de los sistemas, minimizando los tiempos de innovación y aumentando la versatilidad de los resultados duraderos (Quilindo & Vega, 2023).

Además, documentar y entender las decisiones arquitectónicas, tal como lo plantea Christopher Alexander (citado en Quilindo & Vega, 2023), es crucial para diseñar sistemas que sean tanto robustos como flexibles ante futuras demandas, facilitando así la transformación de sistemas complejos en estructuras organizadas y escalables. Por lo tanto, investigar la incorporación de estos conceptos en la educación técnica es esencial para formar desarrolladores que estén capacitados para enfrentar los retos tecnológicos contemporáneos.

La arquitectura de software se percibe como un enfoque integral para el diseño de sistemas de información, dado que aborda de manera meticulosa tanto los requisitos funcionales como los no funcionales. Este enfoque holístico no solo garantiza la calidad del sistema, sino que también potencia la mantenibilidad y escalabilidad del software, aspectos que son fundamentales para maximizar la inversión en proyectos tecnológicos (Fairbanks, citado en Lenguajes de Patrones de Arquitectura de Software).

## **Resultados de la Encuesta sobre Conocimientos de Arquitectura de Software en Estudiantes de Análisis y Desarrollo de Software**

La entrevista se llevó a cabo mediante un formulario diseñado en Google Forms, con la finalidad de analizar el nivel de conocimiento que tienen los estudiantes del programa de Análisis y Desarrollo de Software, en lo relacionado a la arquitectura de software en las diferentes fases de su formación como tecnólogos; también se pretendía comprender cómo estos conocimientos influyen en su formación en la esfera laboral, considerando las habilidades adquiridas en los niveles de su formación en la educación académica.



**Gráfico 1:** Distribución de estudiantes por semestre.



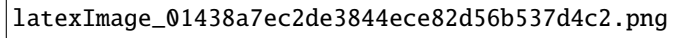
**Gráfico 2:** Conocimientos básicos sobre arquitectura de software antes de iniciar el programa.



**Gráfico 3:** Opinión de los estudiantes sobre sus conocimientos en arquitectura de software.

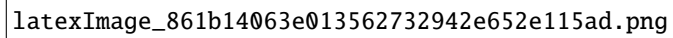


**Gráfico 4:** Principales desafíos en la aplicación de Arquitectura de software



latexImage\_01438a7ec2de3844ece82d56b537d4c2.png

Gráfica 5: Popularidad de los patrones por los encuestados.



latexImage\_861b14063e013562732942e652e115ad.png

Gráfico 6: Opinión de la importancia de la arquitectura de software

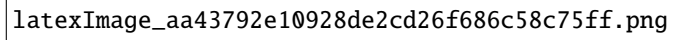


Gráfica 7: Desafíos de implementar arquitectura de software según encuestados



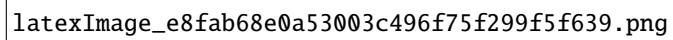
Gráfica 8: En qué creen que podrían mejorar en conocimientos de arquitectura de software





latexImage\_aa43792e10928de2cd26f686c58c75ff.png

Gráfica 9: Opinión sobre la educación recibida sobre arquitectura de software



latexImage\_e8fab68e0a53003c496f75f299f5f639.png

Gráfica 10: Dónde prefieren aprender arquitectura de software los estudiantes.

## Conclusiones de la encuesta

El estudio de las respuestas obtenidas en la encuesta proporciona información importante acerca de cómo los estudiantes ven y comprenden la arquitectura de software.

### Nivel de conocimiento

En cuanto a su nivel de conocimiento, un 70% de los encuestados se califica a sí mismo con un nivel "Básico" en este ámbito, lo que indica que hay un claro interés por una mayor profundización en el tema. Además, un 30% ha participado en algún curso relacionado con la arquitectura de software, pero aún así se consideran con un conocimiento "Básico".

### Experiencia práctica

Más del 56% de los estudiantes ha puesto en práctica algún tipo de arquitectura en sus proyectos de software, lo que demuestra que están aplicando la teoría en situaciones reales.

### Importancia de la arquitectura

Los alumnos identificaron la relevancia de la arquitectura de software en la calidad final de un proyecto. Un notable 83% la valora como "Muy importante" o "Crucial", lo que indica una clara comprensión de su función esencial en el desarrollo de software exitoso. Desafíos en la implementación.

Los principales desafíos percibidos por los estudiantes son:

Falta de conocimientos técnicos (62%)

Falta de tiempo en los proyectos (25%)

Complejidad de la arquitectura (11%)

Limitaciones de recursos (humanos, tecnológicos) (7%)

Estos resultados indican la necesidad de fortalecer las habilidades técnicas y proporcionar más tiempo para la planificación y la implementación de la arquitectura en los proyectos.

#### Áreas de mejora

Los estudiantes identificaron áreas clave en las que desean mejorar su conocimiento:

Patrones de diseño (58%)

Escalabilidad (37%)

Seguridad (25%)

Mantenibilidad (22%)

Estas áreas requieren una mayor atención en la enseñanza de la arquitectura para brindarles a los futuros profesionales herramientas para abordar los desafíos del mundo laboral.

#### Suficiencia del programa

La mayoría de los alumnos, un 69%, opina que el contenido del programa de tecnólogo es "Adecuado, pero podría ser mejor". Por otro lado, un 23% siente que el material no es "No es suficiente" para abordar los retos del mercado laboral.

#### Recursos adicionales

Los estudiantes buscan recursos adicionales para ampliar su conocimiento, principalmente:

Cursos en línea (plataformas como Udemy, Coursera, etc.) (55%)

Prácticas profesionales (31%)

Material académico adicional (14%)

Estos datos sugieren que los estudiantes valoran la flexibilidad y la posibilidad de acceder a recursos complementarios a los ofrecidos en el programa.

## Comentarios y sugerencias

Una mayor profundidad en la enseñanza práctica de los patrones de arquitectura, incluyendo casos de uso reales.

La incorporación de prácticas profesionales para aplicar los conocimientos adquiridos.

Material académico adicional, como libros, artículos y ejemplos de código.

## **Conclusión General**

La encuesta mostró que los estudiantes poseen un entendimiento fundamental sobre la arquitectura de software, sin embargo, es necesario que adquieran habilidades técnicas más avanzadas para enfrentar los retos del entorno laboral. Es crucial enfocar más la enseñanza en la práctica, integrar ejemplos de casos reales y ofrecer recursos adicionales que enriquezcan el proceso de aprendizaje. Aunque el programa existente brinda una base sólida, es esencial que se optimice para satisfacer las demandas de los estudiantes y equiparles adecuadamente para un futuro próspero en el ámbito tecnológico.

## Bibliografía

Palmero, Miguel Angel Sánchez, Martínez, Nemury Silega, & Grass, Olga Yarisbel Rojas. (2019). Revisión de elementos conceptuales para la representación de las arquitecturas de referencias de software. *Revista Cubana de Ciencias Informáticas*, 13(1), 143-157. Recuperado en 11 de noviembre de 2024, de [http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S2227-18992019000100143&lng=es&tlng=es](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992019000100143&lng=es&tlng=es).

Quilindo, luz amanda, y Vega Guarín, J. S. . (2024). Especificando una arquitectura de software: Software architecture specification. *Tecnología Investigación y Academia*, 11(2), 170–181. <https://revistas.udistrital.edu.co/index.php/tia/article/view/18076>

Shvets, A. (2022). Sumérgete en los PATRONES DE DISEÑO <https://refactoring.guru>