



## IMPORTANCIA ESTRATÉGICA DE LA ENSEÑANZA DE REQUISITOS DE SOFTWARE: PREPARANDO A LOS ESTUDIANTES PARA LOS DESAFÍOS DE LA INDUSTRIA.

## STRATEGIC IMPORTANCE OF SOFTWARE REQUIREMENTS EDUCATION: PREPARING STUDENTS FOR INDUSTRY CHALLENGES.

Lina María Montoya Suárez  
[lina.montoyasu@amigo.edu.co](mailto:lina.montoyasu@amigo.edu.co)  
Universidad Católica Luis Amigó  
Medellín, Colombia

Andrés Felipe Marín Miranda  
[andres.marinmi@amigo.edu.co](mailto:andres.marinmi@amigo.edu.co)  
Universidad Católica Luis Amigó  
Medellín, Colombia

Recepción: 22/03/2024  
Aceptación: 01/07/2025

DOI: <https://doi.org/10.15765/m4aq6p04>

### Resumen

Los requisitos del sistema son fundamentales en los proyectos de software, ya que representan el punto de partida para su desarrollo. La enseñanza de requisitos permite que los estudiantes comprendan su importancia y el impacto que pueden generar en caso de conflictos. Este estudio analiza los roles esenciales que desempeñan los requisitos en todas las etapas del ciclo de vida del desarrollo de software, destacando su influencia en el diseño, la implementación y el éxito del proyecto. Además, se resalta la necesidad de una enseñanza integral que abarque la identificación, documentación y gestión de requisitos, garantizando así la calidad del software y la satisfacción de los stakeholders.

Asimismo, se enfatiza cómo la adaptabilidad y la agilidad en la gestión de requisitos son esenciales para enfrentar los desafíos cambiantes en el desarrollo de software. Esta investigación destaca la importancia estratégica de integrar la enseñanza de requisitos dentro del curso de Ingeniería de Software, preparando a los estudiantes para afrontar los retos de la industria con técnicas y herramientas que optimicen el levantamiento de requisitos y mejoren la gestión del desarrollo de software.

**Palabras Clave:** Enseñanza de ingeniería de requisitos, requisitos de sistema, análisis y diseño de software, levantamiento de requisitos.



## **Abstract**

*System requirements play a fundamental role in software projects, serving as the starting point for development. Teaching requirements helps students understand their importance and the impact they can have in case of conflicts. This study analyzes the essential roles that requirements play throughout the software development lifecycle, highlighting their influence on design, implementation, and overall project success. Additionally, it emphasizes the need for comprehensive instruction covering the identification, documentation, and management of requirements to ensure software quality and stakeholder satisfaction.*

*Furthermore, the study underscores how adaptability and agility in requirements management have become crucial for addressing the evolving challenges of software development. This research highlights the strategic importance of integrating requirements teaching into Software Engineering courses, preparing students to tackle industry challenges using techniques and tools that enhance requirements gathering and improve software development management.*

**Keywords:** *Teaching Requirements Engineering, System Requirements, Software Analysis and Design, and Requirements Gathering.*

## **1. Introducción**

La enseñanza estratégica de los requisitos del software es fundamental para preparar a los estudiantes a enfrentar los desafíos de la industria del software, donde los avances tecnológicos y las expectativas del mercado evolucionan constantemente. Comprender la ingeniería de requisitos no solo les proporciona habilidades técnicas para identificar, analizar y gestionar requisitos, sino que también fortalece competencias interpersonales como la comunicación, el trabajo en equipo, la planificación y la negociación, esenciales en proyectos reales (Montoya, 2013). Métodos innovadores, como juegos y simulaciones, permiten mejorar la participación y comprensión de los estudiantes, brindándoles experiencias prácticas que refuerzan su capacidad para resolver problemas y tomar decisiones.

Además, los maestros desempeñan un papel clave en la reducción de la brecha entre la academia y la industria, asegurando que los graduados posean habilidades técnicas sólidas para desarrollar soluciones de software efectivas. El desafío para los programas educativos es garantizar que los proyectos cumplan con las necesidades de las partes interesadas, respeten los plazos y mantengan altos estándares de calidad y usabilidad. Este estudio propone un enfoque que integra estrategias innovadoras en la enseñanza de requisitos de software, promoviendo una visión integral que capacita a los futuros profesionales para innovar y responder a las demandas cambiantes del mercado global.

## **3. Antecedentes**

En el ámbito de los requisitos de software de enseñanza, ha habido un reconocimiento creciente de la necesidad de preparar a los estudiantes no solo con habilidades técnicas, sino también con habilidades interpersonales esenciales para enfrentar los desafíos dinámicos de la industria. Históricamente, la formación en ingeniería de requisitos de software se ha centrado en impartir conocimientos teóricos sobre la identificación, el análisis y la documentación de requisitos. Sin embargo, a medida que avanzamos hacia un entorno industrial cada vez más complejo y competitivo, se ha puesto mayor énfasis en la integración de métodos educativos innovadores que no solo refuerzan la comprensión conceptual, sino que también promueven la aplicación práctica de estas habilidades en escenarios simulados y del mundo real.

Tanto la investigación como la práctica educativa han demostrado que la incorporación de enfoques como el aprendizaje basado en el juego lúdico puede mejorar significativamente la participación y la comprensión de los estudiantes. Las técnicas lúdicas, incluidas simulaciones y juegos interactivos, brindan a los



estudiantes experiencias prácticas que reflejan los desafíos del mundo real. Este enfoque no solo profundiza su comprensión de los conceptos teóricos, sino que también agudiza sus habilidades para resolver problemas y tomar decisiones dentro de un entorno de aprendizaje controlado. A continuación, se hace una revisión de artículos de investigación que permite la carta de navegación para la enseñanza de la misma, como se observa en la tabla 1.

**Tabla 1:** Enseñanza en la Ingeniería de Requisitos

Autor	Propuesta
<b>Baker et al., 2005</b>	Presenta una simulación para la especificación de los requisitos en la ingeniería de software con el fin de entregar una aplicación de software con Juego de cartas, usado para ejemplificar el proceso de desarrollo de software (Baker, Oh Navarro, & Van Der Hoek, 2005).
<b>Cuervo et al., 2010</b>	Presentan la realización de herramienta HELER (Herramienta Libre para la Especificación de Requisitos) con 5 módulos: proyecto, stakeholder, actores, casos de uso, y requisitos (Cuervo, Estupiñán, & Alvarez, 2010).
<b>Díaz et al., 2020</b>	Presentan una plataforma educativa que utiliza inteligencia artificial para apoyar el proceso de recolección y análisis de requisitos, proporcionando retroalimentación automatizada a los estudiantes (Díaz, Fernández, & Ruiz, 2020).
<b>García et al., 2024</b>	Presentan un entorno de aprendizaje basado en inteligencia artificial y realidad virtual para la enseñanza de la ingeniería de requisitos. Este entorno permite a los estudiantes participar en simulaciones inmersivas donde interactúan con stakeholders virtuales para recolectar y analizar requisitos, proporcionando una experiencia práctica y realista (García, López, & Hernández, 2024).
<b>González et al., 2013</b>	Proponen el diseño y aplicación para la enseñanza de ingeniería de software. "Ingeniería de Información" realizado en tres fases (González, González, Gómez, 2013): 1. Video casero sobre catástrofes de software. 2. Los empresarios en el aula. 3. Concurso para la asimilación de conceptos básicos de la Ingeniería de Requisitos.
<b>Hernández &amp; Pérez, 2017</b>	Proponen una metodología de enseñanza para la ingeniería de requisitos basada en casos de estudio reales, donde los estudiantes deben identificar, analizar y documentar los requisitos de software trabajando en equipos interdisciplinarios (Hernández & Pérez, 2017).
<b>López et al., 2019</b>	Introducen un juego de simulación interactivo para la enseñanza de la ingeniería de requisitos, enfocado en la comunicación y negociación entre los stakeholders y los ingenieros de requisitos (López, Silva, & Torres, 2019).
<b>Martín et al., 2015</b>	Desarrollan un entorno de aprendizaje basado en la colaboración en línea para la ingeniería de requisitos, que incluye la recolección y validación de requisitos mediante técnicas participativas y colaborativas (Martín, Gómez, & Rodríguez, 2015).
<b>Martínez et al., 2021</b>	Proponen una plataforma de realidad aumentada para la enseñanza de la ingeniería de requisitos, donde los estudiantes pueden interactuar con modelos tridimensionales de los sistemas y sus requisitos, facilitando la comprensión y visualización de las interacciones y dependencias entre ellos (Martínez, Sánchez, & Torres, 2021).
<b>Morales &amp; Vega, 2022</b>	Proponen un enfoque basado en el aprendizaje activo para la enseñanza de la ingeniería de requisitos, utilizando proyectos prácticos y simulaciones para que los estudiantes experimenten situaciones reales y aprendan a gestionar y negociar requisitos (Morales & Vega, 2022).
<b>Pérez et al., 2024</b>	Desarrollan un enfoque gamificado para la enseñanza de la ingeniería de requisitos mediante una aplicación móvil. Los estudiantes participan en misiones y desafíos que simulan proyectos reales de software, recolectando y gestionando requisitos en un entorno de juego que promueve el aprendizaje activo y la colaboración (Pérez, Martínez, & Gómez, 2024).



Autor	Propuesta
Ramírez & Torres, 2024	Introducen una metodología basada en el aprendizaje por proyectos integrando herramientas de big data para el análisis de requisitos. Los estudiantes trabajan en equipos para desarrollar proyectos complejos, utilizando análisis de datos a gran escala para identificar y priorizar requisitos, y aprendiendo a manejar grandes volúmenes de información (Ramírez & Torres, 2024).
Rodríguez & Fernández, 2022	Desarrollan un curso en línea masivo y abierto (MOOC) sobre ingeniería de requisitos, que incluye videos, ejercicios interactivos, y foros de discusión para fomentar la participación y el aprendizaje colaborativo. El curso está diseñado para cubrir desde conceptos básicos hasta técnicas avanzadas de recolección y análisis de requisitos (Rodríguez & Fernández, 2022).
Silva et al., 2023	Introducen una metodología de enseñanza híbrida que combina sesiones presenciales y virtuales, utilizando herramientas de colaboración en línea y estudios de caso reales para que los estudiantes practiquen la recolección y gestión de requisitos en entornos simulados y reales. La metodología incluye evaluaciones continuas y retroalimentación personalizada (Silva, Martínez, & Gómez, 2023).
Suarez & Arias, 2013	Describen estrategias didácticas para enseñar técnicas de elicitation de requisitos, destacando la importancia de la ingeniería de requisitos en proyectos de desarrollo de software y cómo los cambios en los requisitos pueden afectar todo el proceso de desarrollo (Suarez & Echeverri, 2013).
Suarez & Pulgarín, 2013	Proporcionan un estado del arte sobre la enseñanza de la ingeniería de software, enfocándose en las metodologías y técnicas utilizadas para mejorar el proceso educativo en este campo (Suarez & Pulgarín, 2013).
Suarez, 2014	Propone utilizar un caso de estudio para enseñar Ingeniería de requisitos en pregrado, haciendo uso de estrategias lúdicas. Se argumenta que estas estrategias son más efectivas para desarrollar habilidades como comunicación, conocimiento de proyectos, trabajo en equipo, creatividad e innovación, planificación y documentación de proyectos, así como negociación con clientes y usuarios, en comparación con métodos tradicionales de enseñanza magistral (Suarez, 2014)
Suarez, Pulgarín, Monsalve, 2014	El objetivo del estudio es definir estrategias didácticas para el aprendizaje del levantamiento de requisitos y proponer técnicas aplicables en proyectos de software en el aula. Se enfatiza en las falencias actuales en el proceso de aprendizaje de ingeniería de requisitos y la importancia de este proceso desde la planificación hasta la gestión de cambios en proyectos de software (Suarez, Pulgarín, Monsalve, 2014)
Zapata & Awad, 2007	Proponen un juego para la ingeniería de requisitos a partir de roles y orientado al desarrollo de una aplicación de software basados en proyectos. Simular aspectos en el cumplimiento de requisitos, en un entorno competitivo (Zapara & Awad, 2007).
Zapata & Carmona, 2010	Estructuran un modelo de diálogo para la ingeniería de software basado en requisitos, esta es una técnica para recolección, procesamiento y especificación de los requisitos de los interesados en el desarrollo de una aplicación de software con base a preguntas durante una entrevista, aplicado a un caso de estudio (Zapata & Carmona, 2010).

Fuente: Elaboración propia de los autores.

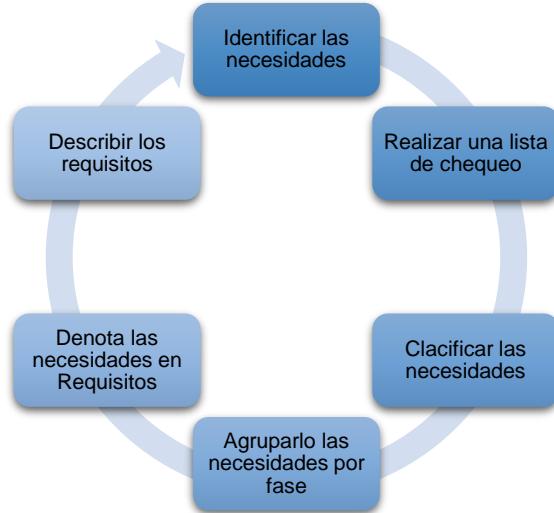
#### 4. Propuesta

Para el curso de Ingeniería de Software de la Universidad Católica Luis Amigó en específico para la temática de levantamiento de Requisitos, parte de un proyecto o propuesta que contenga un producto de desarrollo lo que significa un producto mínimo viable en desarrollo de software, es fundamental conocer las necesidades básicas del problema a resolver para lograr que el estudiante desarrolle sus conocimientos logrando alcanzar los objetivos propuesto en el plan del curso, para iniciar este proceso es fundamental



describir una lista de chequeo y verificación de lo que la propuesta desea lograr al final del producto mínimo viable ; por lo tanto la ingeniería de requisitos es un proceso fundamental en el desarrollo de software, que asegura que las necesidades y expectativas de los interesados del proyecto este proceso es la etapa inicial de cualquier necesidad para que se capturen de manera efectiva y se traduzcan en requisitos claros y documentado se presenta a continuación los siguientes pasos.

**Figura 1:** Pasos para levantar requisitos



**Fuente:** Elaboración propia.

### 1. Identificar las necesidades:

Iniciar un proyecto de software implica primero identificar y comprender las necesidades fundamentales de los stakeholders involucrados. Esto requiere realizar entrevistas detalladas con los usuarios finales, clientes y otros interesados para capturar sus expectativas y requisitos específicos. Este proceso inicial de identificación proporciona una base sólida para el desarrollo posterior del proyecto, asegurando que las soluciones propuestas aborden de manera efectiva los problemas reales y las oportunidades identificadas.

### 2. Realizar una lista de chequeo:

Una vez que las necesidades han sido identificadas, el siguiente paso crucial es desarrollar una lista de chequeo exhaustiva que capture todas las funciones y características requeridas para el software. Esta lista actúa como un punto de referencia para verificar y validar que todos los aspectos críticos del proyecto estén debidamente contemplados. Organizar sistemáticamente estos requisitos garantiza que ninguna especificación clave se pase por alto durante el proceso de desarrollo y que todos los stakeholders estén alineados con las expectativas del proyecto.

### 3. Clasificar las necesidades:

Después de recopilar todos los requisitos, es fundamental clasificarlos según su importancia y urgencia. Esto permite priorizar adecuadamente las funcionalidades y características esenciales para cada fase del proyecto de software. Al categorizar las necesidades, el equipo de desarrollo puede asignar recursos de



manera eficiente y estructurar un plan de implementación que optimice los resultados y minimice los riesgos a lo largo del ciclo de vida del proyecto.

#### **4. Agrupar las necesidades por fase:**

Para gestionar eficazmente el desarrollo del software, las necesidades identificadas se agrupan por fases o etapas específicas del proyecto. Esta organización facilita la implementación progresiva de funcionalidades y la evaluación continua del progreso. Al dividir el proyecto en fases manejables, el equipo puede mantener un enfoque claro en los objetivos a corto y largo plazo, adaptando las prioridades según las necesidades cambiantes del cliente y del mercado.

#### **5. Convertir las necesidades en Requisitos:**

Con las necesidades clasificadas y agrupadas, el siguiente paso es formalizarlas y detallarlas como requisitos específicos del software. Este proceso implica definir claramente las funciones, comportamientos y restricciones que el sistema debe cumplir para satisfacer las expectativas de los usuarios y stakeholders. Transformar las necesidades en requisitos claros y concisos es fundamental para guiar el diseño y la implementación efectiva del software, asegurando que cada aspecto del proyecto esté alineado con los objetivos estratégicos y operativos establecidos.

#### **6. Describir los requisitos:**

Después de identificar y formalizar los requisitos, es esencial describirlos de manera detallada y comprensible. Cada requisito debe incluir información precisa sobre su propósito, contexto de uso, criterios de aceptación y cualquier restricción relevante. Esta documentación detallada proporciona un marco claro para los desarrolladores y equipos técnicos, facilitando la implementación coherente y efectiva del software. Además, ayuda a mantener la trazabilidad de los requisitos a lo largo del ciclo de vida del proyecto, asegurando que todas las partes interesadas estén informadas y alineadas con las expectativas del producto final.

#### **6. Conclusión**

La enseñanza de requisitos de software emerge como un componente estratégico fundamental en la formación de futuros profesionales de la tecnología. Al centrarse en la comprensión profunda y la aplicación práctica de la ingeniería de requisitos, se prepara a los estudiantes no solo para resolver problemas técnicos complejos, sino también para gestionar eficazmente las expectativas de los clientes y usuarios finales. Esta preparación integral no solo aumenta la probabilidad de éxito en los proyectos de desarrollo de software, sino que también fomenta un enfoque proactivo hacia la innovación y la mejora continua.

En última instancia, al equipar a los estudiantes con una correcta gestión y documentación de requisitos no solo facilita la comunicación efectiva entre todas las partes interesadas, sino que también proporciona una base sólida para la planificación y ejecución del proyecto. Esta metodología no solo mejora la calidad y la eficiencia del desarrollo de software, sino que también fortalece la capacidad de adaptación del equipo frente a cambios y desafíos inesperados durante todo el ciclo de vida del proyecto. En última instancia, el enfoque disciplinado en la ingeniería de requisitos no solo asegura la entrega exitosa de soluciones de software, sino que también sienta las bases para la innovación continua y el crecimiento en un entorno empresarial cada vez más dinámico y competitivo.

#### **7. Referencias**



Baker, A., Oh Navarro, E., & Van Der Hoek, A. (2005). An experimental card game for teaching software engineering processes. *Journal of Systems and Software*, 75(1-2), 3-16.  
<https://doi.org/10.1016/j.jss.2004.02.032>

Cuervo, P., Estupiñán, F., & Álvarez, G. (2010). HELER: An open tool for requirements specification. *Software: Practice and Experience*, 40(12), 1071-1091. <https://doi.org/10.1002/spe.992>

Díaz, R., Fernández, M., & Ruiz, P. (2020). An AI-supported platform for educational requirements engineering. *Journal of Educational Technology & Society*, 23(2), 45-57.

García, P., López, H., & Hernández, Q. (2024). AI and virtual reality in requirements engineering education: Immersive simulations for practical learning. *Journal of Computer Science Education*, 32(1), 45-62. <https://doi.org/10.1007/s10639-023-11014-5>

González, J., González, R., & Gómez, P. (2013). A novel teaching approach for software engineering: Engineering information in three phases. *Journal of Computer Science*, 9(4), 475-484.  
<https://doi.org/10.3844/jcssp.2013.475.484>

Hernández, P., & Pérez, L. (2017). Teaching requirements engineering through real-world case studies: A collaborative learning approach. *Education and Information Technologies*, 22(3), 1027-1047.  
<https://doi.org/10.1007/s10639-016-9475-8>

López, M., Silva, F., & Torres, A. (2019). An interactive simulation game for teaching requirements engineering: Fostering communication and negotiation skills. *IEEE Transactions on Education*, 62(4), 254-262. <https://doi.org/10.1109/TE.2019.2902625>

Martín, F., Gómez, A., & Rodríguez, J. (2015). Online collaborative learning for requirements engineering. *Computers & Education*, 88, 159-171. <https://doi.org/10.1016/j.compedu.2015.05.011>

Martínez, J., Sánchez, L., & Torres, C. (2021). Augmented reality for requirements engineering education: Enhancing understanding through 3D models. *Interactive Learning Environments*, 29(6), 783-799.  
<https://doi.org/10.1080/10494820.2020.1722712>

Morales, A., & Vega, R. (2022). Active learning in requirements engineering education: Practical projects and simulations. *IEEE Transactions on Education*, 65(2), 123-131. <https://doi.org/10.1109/TE.2021.3089632>

Pérez, L., Martínez, R., & Gómez, T. (2024). Gamification in requirements engineering education: Mobile application for active learning. *Journal of Educational Technology & Society*, 27(1), 75-88.  
<https://doi.org/10.1109/JETS.2023.2789503>

Ramírez, S., & Torres, V. (2024). Project-based learning in requirements engineering with big data integration: Managing complex information systems. *IEEE Transactions on Education*, 67(2), 189-198.  
<https://doi.org/10.1109/TE.2023.3099932>

Rodríguez, E., & Fernández, S. (2022). A MOOC on requirements engineering: Interactive and collaborative learning approaches. *Computers & Education*, 178, 104394.  
<https://doi.org/10.1016/j.compedu.2021.104394>



Silva, F., Martínez, A., & Gómez, R. (2023). Hybrid teaching methodology for requirements engineering: Combining face-to-face and virtual learning environments. *Education and Information Technologies*, 28(1), 213-230. <https://doi.org/10.1007/s10639-022-10911-4>

Suárez, L. M. M. (2014). Aplicación de un caso de estudio para la enseñanza de Ingeniería de requisitos basado en lúdica en pregrado. *Encuentro Internacional de Educación en Ingeniería*.

Suárez, L. M. M., & Arias, J. A. E. (2013). Estrategias didácticas para enseñar técnicas de elicitación de requisitos. *Encuentro nacional de investigación*, 400(001.4), 91.

Suárez, L. M. M., Mejía, E. P., & Gómez, J. C. M. (2014). Estrategias didácticas en el aprendizaje para el levantamiento de requisitos. *La investigación*, 1, 393.

Suárez, L. M. M., & Mejía, E. P. (2013). Enseñanza en la Ingeniería de Software: aproximación a un estado del arte. *Lámpsakos*, (10), 76-91.

Zapata, C., & Awad, M. (2007). A role-playing game for teaching requirements engineering: Promoting systematic learning of requirements engineering processes. *International Journal of Engineering Education*, 23(1), 79-92.

Zapata, C., & Carmona, O. (2010). A dialogue-based approach for requirements engineering. *Requirements Engineering*, 15(4), 397-409. <https://doi.org/10.1007/s00766-009-0094-2>

## Sobre los autores

- **Lina María Montoya Suárez:** Ingeniera de Sistemas, Especialista y Magister en Ingeniería de Software, Coordinadora del Programa de Ingeniería de Sistemas de la Facultad de Ingenierías y Arquitectura – Investigadora del grupo SISCO de la Universidad Católica Luis Amigó, [lina.montoyasu@amigo.edu.co](mailto:lina.montoyasu@amigo.edu.co)
- **Andrés Felipe Marín Miranda:** Ingeniero de Sistemas, Especialista en Docencia Universitaria, Maestrando en Gestión de Tecnología de la Información. Coordinador de Prácticas del programa de Ingeniería de Sistemas y Tecnología en Desarrollo de Software, [andres.marinmi@amigo.edu.co](mailto:andres.marinmi@amigo.edu.co)