ASCE MAGAZINE ISSN: 3073-1178

Revista ASCE Magazine, Periodicidad: Trimestral Julio - Septiembre, Volumen: 4, Número: 3, Año: 2025 páginas 944 - 969

Doi: https://doi.org/10.70577/ASCE/944.969/2025

Recibido: 2025-05-22
Aceptado:2025-06-22
Publicado:2025-07-25

Ingeniería de Requisitos en Entornos Ágiles y Tradicionales: Un Enfoque Integral para el Desarrollo de Software Centrado en el Usuario

Requirements Engineering in Agile and Traditional Environments: A Comprehensive Approach to User-Centered Software Development

Cristian Geovanny Merino Sánchez

https://orcid.org/0000-0003-3645-5165

c_merino@espoch.edu.ec

Docente de la Escuela Superior Politécnica De Chimborazo

Riobamba-Ecuador

Cómo citar

Merino Sánchez, C. G. (2025). Ingeniería de Requisitos en Entornos Ágiles y Tradicionales: Un Enfoque Integral para el Desarrollo de Software Centrado en el Usuario. *ASCE*, *4*(3), 944–969.

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

Resumen

En el contexto de la transformación digital acelerada, las organizaciones enfrentan desafíos en el desarrollo de software que responda de manera efectiva a las necesidades de los usuarios. La ingeniería de requisitos, aunque fundamental para el éxito de los proyectos de software, a menudo presenta problemas como ambigüedades, contradicciones y falta de trazabilidad en los requisitos, lo que puede generar errores críticos durante el ciclo de desarrollo. Este artículo tiene como objetivo analizar la ingeniería de requisitos en entornos ágiles y tradicionales, con un enfoque en mejorar la calidad, trazabilidad y utilidad de los sistemas mediante una gestión adecuada de los requisitos. Para ello, se utilizó una metodología mixta, combinando la investigación documental y el análisis de casos prácticos en instituciones educativas y de salud en Ecuador. Los métodos empleados incluyen entrevistas semiestructuradas, observación directa, cuestionarios y prototipos, siguiendo los estándares internacionales ISO/IEC/IEEE 29148:2018 y BABOK v3. Los resultados obtenidos muestran que la aplicación rigurosa de técnicas de elicitación y validación, junto con la combinación de enfoques ágiles y tradicionales, mejora la precisión y pertinencia de los requisitos. Además, se encontró que la participación activa de los stakeholders y el uso de herramientas digitales favorecen una mejor alineación de los proyectos con las necesidades reales de los usuarios. Finalmente, se concluye que la ingeniería de requisitos debe ser integrada como una competencia transversal en la formación de ingenieros de software, priorizando un enfoque que combine aspectos técnicos, metodológicos y éticos para garantizar el éxito de la transformación digital.

Palabras clave: Ingeniería de Requisitos, Desarrollo de Software, Elicitación, Validación, Gestión de Requisitos, Estándares ISO, Agilidad.

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

Abstract

In the context of accelerated digital transformation, organizations face challenges in developing software that effectively responds to user needs. Requirements engineering, while critical to the success of software projects, often presents issues such as ambiguities, contradictions, and lack of traceability in requirements, which can lead to critical errors during the development cycle. This article aims to analyze requirements engineering in agile and traditional environments, with a focus on improving the quality, traceability and usefulness of systems through proper requirements management. To this end, a mixed methodology was used, combining documentary research and the analysis of practical cases in educational and health institutions in Ecuador. The methods used include semi-structured interviews, direct observation, questionnaires and prototypes, following the international standards ISO/IEC/IEEE 29148:2018 and BABOK v3. The results obtained show that the rigorous application of elicitation and validation techniques, together with the combination of agile and traditional approaches, improves the accuracy and relevance of requirements. In addition, it was found that the active participation of stakeholders and the use of digital tools favor a better alignment of projects with the real needs of users. Finally, it is concluded that requirements engineering should be integrated as a transversal competence in the training of software engineers, prioritizing an approach that combines technical, methodological and ethical aspects to guarantee the success of digital transformation.

Keywords: Requirements engineering, software development, elicitation, validation, requirements management, ISO standards, agility.

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

Introducción

En el actual contexto de transformación digital acelerada, el desarrollo de sistemas de software eficientes, sostenibles y centrados en el usuario se ha convertido en una prioridad estratégica para múltiples sectores. Las organizaciones enfrentan desafíos complejos relacionados con la calidad, la adaptabilidad y la usabilidad de sus productos digitales, los cuales exigen soluciones tecnológicas capaces de responder con precisión a necesidades reales. Frente a este escenario, la ingeniería de requisitos emerge como una disciplina clave que articula los objetivos del negocio, las restricciones tecnológicas y las expectativas de los usuarios. Su función principal radica en identificar, documentar y gestionar de manera sistemática y verificable las condiciones que debe cumplir un sistema para satisfacer propósitos específicos, lo cual no se limita a un acto técnico; representa un proceso complejo de comprensión mutua, análisis contextual y construcción colaborativa.

Numerosas investigaciones y reportes técnicos han señalado que una de las principales causas del fracaso en proyectos de software se debe a la gestión inadecuada de los requisitos. Arangurí y Bravo (2021) sostienen que más del 50 % de los errores críticos que afectan al producto final ocurren durante las etapas iniciales de levantamiento y análisis de requisitos, lo cual tiene consecuencias tanto económicas como funcionales. La falta de claridad, ambigüedad, contradicciones internas y omisiones son síntomas frecuentes de una ingeniería de requisitos mal ejecutada, que afectan no solo al producto, sino también a la percepción de valor por parte de los stakeholders. De hecho, los ejemplos emblemáticos, como los fallos en sistemas de salud digital en países europeos o el colapso de plataformas educativas durante la pandemia, evidencian cómo un mal levantamiento de requisitos puede traducirse en pérdida de confianza institucional, costos elevados de corrección y, en algunos casos, impactos sociales negativos.

El abordaje de esta problemática es de gran relevancia en el contexto actual. Por un lado, porque el software se ha convertido en el motor de múltiples industrias: salud, educación, finanzas, logística, entre otras. Por otro lado, porque la transformación digital exige metodologías más colaborativas, flexibles y centradas en el usuario, lo cual representa un reto adicional para las prácticas tradicionales de ingeniería de software. De hecho, la aparición de metodologías ágiles como Scrum o Kanban ha reconfigurado la manera en que se conceptualiza, planifica y construye un producto, haciendo que los requisitos ya no sean entidades estáticas, sino artefactos vivos que

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

evolucionan con el tiempo. Al respecto, este cambio demanda profesionales con competencias técnicas, analíticas, comunicativas y éticas, capaces de actuar como puentes entre el mundo técnico y las necesidades humanas. La ingeniería de requisitos debe entonces asumir una función integradora y dinámica, adaptada a entornos de alta incertidumbre.

Desde una perspectiva teórica, la ingeniería de requisitos se enmarca en estándares internacionales como la norma ISO/IEC/IEEE 29148:2018, la cual define un conjunto de procesos, tareas y productos necesarios para una adecuada gestión de requisitos durante el ciclo de vida del software. Al respecto también se señala que: "*Un sistema no puede ser mejor que los requisitos sobre los cuales se construye." — ISO/IEC/IEEE 29148:2018.* En adición, esta norma establece cinco fases esenciales:

Tabla 1
Resumen del Proceso de ingeniería de requisitos

Fase	Descripción
Elicitación	Recoger información de los stakeholders mediante entrevistas, talleres, observación, etc.
Análisis	Refinar, clasificar y priorizar los requisitos. Identificar conflictos y redundancias.
Especificación	Documentar los requisitos de manera formal o semiformal (documento SRS, casos de uso, historias de usuario).
Validación y verificación	Asegurar que los requisitos estén bien formulados y reflejen fielmente lo que se necesita.
Gestión de requisitos	Controlar los cambios, versiones y trazabilidad de los requisitos a lo largo del proyecto.

Fuente: Elaborado a partir de la información de ISO/IEC/IEEE 29148:2018

- La elicitación y análisis se hacen mediante *backlog grooming* y planificación de sprints.
- La especificación se basa en *historias de usuario* con criterios de aceptación.
- La validación se realiza en cada *demo* o revisión de sprint.
- La gestión se apoya en herramientas como Jira o Azure DevOps.

A continuación, se presenta una tabla resumen del proceso, y la respectiva figura, que grafica el mismo:

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

Tabla 2

Resumen del Proceso-Producto y técnicas de ingeniería de requisitos

Actividad	Producto generado	Técnicas aplicadas
Elicitación	Lista inicial de necesidades	Entrevistas, observación, encuestas
Análisis	Requisitos organizados y priorizados	Clasificación, refinamiento, prototipos
Especificación	Documento de requisitos (SRS)	Casos de uso, tablas, formatos IEEE 830
Validación	Requisitos aprobados	Revisiones, pruebas, walk-through
Gestión	Historial de versiones, trazabilidad	Control de cambios, seguimiento con Jira

Fuente: Elaboración Propia

Figura 1

Ciclo de la Ingeniería de Requisitos

Fuente: Elaboración Propia

Cada una cumple funciones particulares en la construcción de sistemas confiables, desde la identificación de necesidades hasta el control de cambios. Adicionalmente, el BABOK v3 (International Institute of Business Analysis, 2015) aporta un marco complementario que enfatiza la necesidad de integrar buenas prácticas del análisis de negocio en el tratamiento de los requisitos, promoviendo un enfoque sistémico y alineado con el valor para el cliente. Así, el presente estudio se apoya en estos marcos teóricos para sustentar sus análisis y propuestas.

El concepto de "requisito" es abordado como una necesidad o condición que debe cumplirse para alcanzar un objetivo, cumplir un contrato o satisfacer una normativa. Pueden ser funcionales o no funcionales. Los primeros definen lo que el sistema debe hacer, incluyendo funcionalidades, reglas

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

de negocio y respuestas a entradas. Los segundos establecen cómo debe comportarse el sistema, abarcando aspectos como rendimiento, seguridad, mantenibilidad o usabilidad (Arangurí y Bravo, 2021; Echeverri et al., 2013). También se distinguen requisitos del usuario, que reflejan expectativas en lenguaje natural, y requisitos del sistema, que traducen esas expectativas en especificaciones técnicas.

Tabla 3

Principales tipos de requisitos en ingeniería de software

Tipo de requisito	Descripción
Funcionales	Especifican lo que el sistema debe hacer. Incluyen funcionalidades, reglas de negocio, etc.
No funcionales	Describen cómo debe comportarse el sistema: desempeño, seguridad, confiabilidad, usabilidad.
De usuario	Expresan las necesidades y objetivos del usuario, en lenguaje natural.

Fuente: Adaptado de Echeverri et al. (2013). “Reflexiones sobre ingeniería de requisitos y pruebas de software”.

La correcta clasificación, especificación y validación de estos requisitos permite reducir ambigüedades, detectar contradicciones y establecer trazabilidad, lo cual se traduce en una mayor calidad del producto final.

Otros marcos teóricos relevantes, como la norma ISO/IEC 25010:2011, amplían la perspectiva al incorporar atributos de calidad que los requisitos deben garantizar: fiabilidad, eficiencia, seguridad, mantenibilidad y compatibilidad. Asimismo, la perspectiva del diseño centrado en el usuario (DCU) refuerza la idea de que los requisitos deben surgir de una comprensión profunda de las experiencias, necesidades y expectativas de los usuarios finales. La combinación de estos marcos permite construir soluciones tecnológicas no solo funcionales, sino también emocionalmente significativas, éticamente viables y culturalmente pertinentes.

En el campo académico y profesional, se han desarrollado múltiples estudios que analizan cómo se aplican estos principios en diferentes sectores y países. Bejarano & Baquero (2020) destacan la importancia de adaptar esta disciplina a los entornos ágiles, donde los requisitos deben gestionarse de forma iterativa, priorizada y validada de manera continua. Jiménez (2021) argumenta que en contextos dinámicos, los artefactos tradicionales (como documentos extensos de requisitos) deben complementarse con historias de usuario, criterios de aceptación y prototipos, para favorecer la comunicación y la retroalimentación entre los miembros del equipo. Asimismo, Echeverri et al.

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

(2013) subrayan la necesidad de aplicar técnicas de elicitación como entrevistas, observación directa, talleres colaborativos y análisis documental para lograr una comprensión profunda del dominio del problema. En Latinoamérica, la literatura técnica ha crecido en los últimos años, pero aún persisten desafíos relacionados con la adopción efectiva de estándares, la capacitación profesional y la integración de enfoques ágiles con marcos normativos.

En cuanto al contexto específico en el que se desarrolla esta investigación, se toma como referencia la experiencia de proyectos institucionales en ámbitos educativos y de salud en Ecuador, cuyos entornos presentan características particulares que justifican su análisis. En el caso de instituciones educativas, como la Escuela Superior Politécnica de Chimborazo (ESPOCH), se detecta una creciente necesidad de sistematizar procesos académicos, entre ellos la gestión de tutorías, con miras a garantizar la trazabilidad, cumplimiento normativo y mejora del acompañamiento estudiantil. Por otro lado, en el sector salud, se evidencian ineficiencias en los mecanismos tradicionales de agendamiento de citas médicas, lo cual afecta tanto a los usuarios como a la operatividad de las clínicas privadas. Ambos contextos requieren soluciones digitales centradas en el usuario, sostenibles y alineadas con estándares de calidad.

A nivel nacional, el proceso de digitalización del Estado ecuatoriano ha impulsado la implementación de plataformas digitales en diversas instituciones, aunque con resultados dispares. Existen desafíos estructurales como la falta de conectividad en zonas rurales, la baja alfabetización digital de ciertos grupos poblacionales y la carencia de profesionales capacitados en metodologías modernas de desarrollo. En este contexto, la ingeniería de requisitos no solo aporta una solución técnica, sino también una herramienta para la planificación estratégica, la transparencia y la gobernanza digital.

Desde una perspectiva metodológica, este trabajo adopta un enfoque mixto, combinando el análisis documental y normativo con el estudio de casos prácticos reales. Los casos incluyen el desarrollo de un sistema de reservas médicas y una plataforma de gestión de tutorías académicas, en los cuales se aplicaron técnicas de elicitación, análisis y validación de requisitos conforme a los estándares ISO/IEC/IEEE 29148:2018 y BABOK v3. Se utilizaron herramientas como entrevistas semiestructuradas, observación directa, prototipado, matrices de trazabilidad y tableros ágiles (Jira, Trello), lo cual permitió una implementación integral de la ingeniería de requisitos en entornos híbridos.

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

Uno de los aspectos distintivos de este estudio es el énfasis en las competencias del analista de requisitos, quien no solo debe dominar herramientas técnicas, sino también habilidades blandas como la empatía, la escucha activa, la negociación y la facilitación. El perfil profesional es crucial en proyectos donde los usuarios no siempre manejan un lenguaje técnico, como ocurre en comunidades académicas o centros de salud. El analista actúa como traductor entre mundos, ayudando a construir consensos, clarificar expectativas y alinear visiones divergentes. Su rol ético y humano es fundamental para garantizar que los sistemas desarrollados respondan a realidades complejas y necesidades legítimas.

Asimismo, se propone una reflexión crítica sobre la enseñanza de la ingeniería de requisitos. Tradicionalmente, esta disciplina se abordaba como un conjunto de técnicas secuenciales aplicadas en etapas iniciales del proyecto. Sin embargo, la realidad muestra que los requisitos evolucionan constantemente y requieren ser gestionados como un proceso continuo. Por ello, se propone incorporar la ingeniería de requisitos como una competencia transversal en la formación de ingenieros de software, integrando aspectos técnicos, metodológicos y éticos desde una perspectiva práctica y contextualizada.

En este sentido, el presente artículo propone como objetivo general analizar de forma integral los componentes clave de la ingeniería de requisitos, combinando teoría y práctica en contextos reales, para resaltar su valor estratégico en el desarrollo de software centrado en el usuario y alineado con los objetivos del negocio. A través de la exposición de estándares internacionales, herramientas digitales, técnicas de elicitación, y casos de aplicación, se busca evidenciar cómo una adecuada gestión de requisitos puede contribuir significativamente al éxito de los proyectos de software, mejorando su calidad, utilidad y alineación con los objetivos del negocio.

En suma, este trabajo contribuye al campo de la ingeniería de software al presentar un análisis sistemático, contextualizado y orientado a la práctica de la gestión de requisitos, con énfasis en el usuario y la realidad institucional ecuatoriana. Su aporte se orienta no solo a enriquecer la literatura académica, sino también a ofrecer un referente práctico para profesionales, estudiantes y organizaciones que enfrentan el desafío de diseñar e implementar soluciones tecnológicas efectivas. Al integrar teoría, experiencia de campo y herramientas actuales, se espera que esta propuesta fortalezca la formación de ingenieros con competencias analíticas, comunicativas y

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

éticas, capaces de liderar procesos de transformación digital desde una perspectiva humana y estratégica.

Metodología

La presente investigación se enmarca dentro de un enfoque metodológico mixto, con predominancia cualitativa de carácter exploratorio-descriptivo. Su finalidad fue analizar y ejemplificar el proceso de ingeniería de requisitos tanto desde una perspectiva teórica como aplicada, centrando la atención en contextos reales donde el desarrollo de software se enfrenta a necesidades específicas del usuario, restricciones operativas y dinámicas institucionales complejas. La investigación se desarrolló en dos fases principales: una fase documental y normativa, y otra empírica mediante estudios de caso.

En primer lugar, se realizó un estudio documental sistemático sobre la ingeniería de requisitos en sus dimensiones tradicional y ágil. Se consultaron normas internacionales como la ISO/IEC/IEEE 29148:2018, el estándar ISO/IEC 25010 sobre calidad del software, el BABOK v3 como guía de buenas prácticas en análisis de negocios, así como literatura académica actualizada (Arangurí & Bravo, 2021; Bejarano & Baquero, 2020; Echeverri et al., 2013; Jiménez, 2021). La revisión permitió construir una base conceptual sólida para el diseño de la segunda fase.

En la segunda fase se desarrolló una aplicación práctica mediante el análisis de dos casos de estudio institucionales:

a) Sistema de Reservas Médicas en Línea para una clínica privada.
b) La implementación de una Plataforma de Gestión de Tutorías Académicas en la Escuela Superior Politécnica de Chimborazo (ESPOCH).

Ambos casos fueron seleccionados intencionadamente por su relevancia práctica y la posibilidad de acceder a datos de campo y retroalimentación directa con usuarios reales. Se trató de un diseño no experimental y de tipo transversal, dado que no se manipularon variables independientes ni se introdujeron condiciones controladas, sino que se observó el proceso tal como ocurre en el entorno natural del proyecto.

Las técnicas utilizadas durante la recolección de datos incluyeron:

- Entrevistas semiestructuradas con stakeholders clave (gerentes, usuarios operativos, personal técnico).

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

- Observación directa de los procesos actuales (agendamiento de citas, gestión de tutorías).
- Cuestionarios online para usuarios finales.
- Talleres colaborativos para levantar funcionalidades clave.
- Prototipos de baja fidelidad para validar requerimientos tempranos.

Para el análisis de requisitos, se aplicaron herramientas derivadas del enfoque clásico de ingeniería como la matriz de trazabilidad, el modelado con UML (casos de uso), y especificaciones SRS con base en IEEE 830. En entornos ágiles, se usaron historias de usuario con criterios de aceptación, tableros Kanban (Jira, Trello), y técnicas de priorización como MoSCoW y matriz de valor vs. riesgo. Los requisitos se clasificaron como funcionales, no funcionales, de usuario y del sistema.

Los métodos de validación y verificación incluyeron sesiones de walkthrough, revisión técnica cruzada, uso de listas de verificación basadas en ISO 29148 y feedback de usuarios finales mediante entrevistas o encuestas posteriores al uso de prototipos. La verificación buscó asegurar que los requisitos estuvieran bien redactados, completos y consistentes; la validación garantizó que respondieran a necesidades reales y fueran aceptables para el usuario.

Durante el proceso, se utilizaron múltiples herramientas tecnológicas para modelar, documentar y gestionar los requisitos. Entre ellas destacan:

- Lucidchart y Draw.io para diagramas de casos de uso.
- Confluence y Google Docs para documentación colaborativa.
- Excel para matrices de trazabilidad y control de cambios.
- Jira y Trello para priorización y seguimiento de requisitos en entornos ágiles.

La primera unidad de análisis (sistema de reservas médicas) permitió demostrar cómo un enfoque participativo en la elicitación de requisitos mejora la pertinencia y usabilidad de la solución. El segundo caso (tutorías académicas en la ESPOCH) evidenció cómo una documentación rigurosa bajo estándares internacionales, combinada con prácticas ágiles, facilita la trazabilidad, adaptación y validación continua del producto.

Los datos obtenidos fueron analizados bajo un enfoque categorial, organizando los requisitos según su tipo, fuente, herramienta de captura, nivel de validación y alineación con objetivos estratégicos del sistema. Esta organización permitió obtener patrones sobre la eficacia de las

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

técnicas aplicadas, las dificultades frecuentes (ambigüedad, redundancia, cambios tardíos) y las fortalezas del proceso (colaboración, uso de prototipos, validación iterativa).

El diseño metodológico permitió cumplir con el objetivo de estudiar integralmente la ingeniería de requisitos en escenarios reales. Asimismo, facilitó la transferencia de conocimiento hacia contextos similares, como instituciones públicas o empresas del sector salud o educativo en países en desarrollo. Los resultados permiten inferir que la integración equilibrada entre formalismo técnico, herramientas ágiles y enfoque centrado en el usuario constituye una práctica efectiva para mejorar la calidad del software desarrollado.

Finalmente, se reconoce que esta metodología no se limita a los casos analizados, sino que puede replicarse y adaptarse a otros contextos organizacionales. La combinación de estándares internacionales, herramientas digitales accesibles, y técnicas de validación participativa ofrece un modelo escalable para proyectos pequeños y medianos que busquen asegurar una ingeniería de requisitos flexible y alineada con los usuarios.

Resultados

a) Presentación Caso: Sistema de Reservas para Consultas Médicas en Línea

1. Escenario del proyecto

Una clínica privada ubicada en una ciudad intermedia busca digitalizar su proceso de agendamiento de citas médicas. Actualmente, las reservas se hacen por teléfono o de manera presencial. Esto genera cuellos de botella, errores en horarios, y poca trazabilidad de la información. La gerencia ha solicitado el desarrollo de un sistema web y móvil para que los pacientes puedan:

- Consultar disponibilidad médica
- Reservar citas
- Recibir recordatorios
- Cancelar o reprogramar turnos

El objetivo del proyecto es implementar un Sistema de Reservas de Consultas Médicas en Línea que mejore la eficiencia operativa y la experiencia del paciente.

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

2. Identificación de stakeholders

Utilizando técnicas de análisis de entorno y la matriz de poder/interés, se identifican los siguientes actores:

Tabla 4

Principales tipos de requisitos en ingeniería de software

Stakeholder	Rol/Interés	Tipo según matriz de stakeholders
Gerente general	Toma decisiones de negocio	Promotor (alto poder, alto interés)
Personal de recepción	Usuarios internos, operativos	Defensores (bajo poder, alto interés)
Pacientes	Usuarios finales	Latentes (alto interés, bajo poder)
Departamento TI	Implementadores	Defensores
Médico jefe	Supervisor de operaciones médicas	Promotor
Entidad reguladora	Normativa en salud digital	Latente

3. Resultados de la elicitación

Requisitos funcionales preliminares identificados:

- El sistema debe permitir agendar una cita médica en menos de 3 pasos.
- Los pacientes deben poder cancelar o reprogramar citas desde su dispositivo.
- Los usuarios administrativos deben tener un panel para ver disponibilidad por médico.

Requisitos no funcionales:

- El sistema debe enviar notificaciones al correo electrónico del paciente 24h antes de la cita.
- El tiempo de carga del sistema debe ser inferior a 3 segundos.
- La interfaz debe ser accesible en dispositivos móviles y cumplir con WCAG nivel AA.

4. Aplicación de estándares

- Se documentaron los requisitos funcionales y no funcionales siguiendo la estructura de la norma IEEE 830.
- Se establecieron criterios de trazabilidad y calidad en base a

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

- ISO/IEC/IEEE 29148.
- Se priorizaron requisitos con base en el BABOK v3, usando el criterio de valor al negocio y riesgo técnico.

5. Documentación generada

Se elaboraron los siguientes entregables:

- Mapa de stakeholders con clasificación y estrategias de gestión.
- Acta de entrevistas y minutas de talleres.
- Documento de especificación de requisitos funcionales y no funcionales.
- Prototipo de interfaz validado por los usuarios.
- Plan de trazabilidad de requisitos.

Este caso práctico permitió aplicar de forma integrada todos las etapas esenciales en un proceso de Elicitación de Requisitos. Se pone en evidencia la importancia de una buena comunicación con los interesados, la aplicación de técnicas apropiadas de elicitación y la necesidad de alinear la documentación con estándares reconocidos. La experiencia práctica también refuerza el rol del analista como puente entre el usuario y el sistema.

6. Análisis y Documentación de Requisitos

Análisis formal y negociación: El análisis formal permite detectar conflictos, ambigüedades, omisiones o redundancias en los requisitos. La negociación se enfoca en resolver contradicciones entre stakeholders, priorizar lo más importante y lograr consensos viables.

Tabla 5

Problemas comunes detectados en el análisis de requisitos

Problema	Ejemplo	Técnica para resolverlo
Ambigüedad	"El sistema debe ser rápido"	Reformular usando métricas precisas
Redundancia	Requisitos repetidos con diferentes palabras	Consolidación y validación con partes interesadas
Inconsistencia	Un requisito contradice otro	Reuniones de revisión técnica
Requisitos incompletos	Falta información clave como entradas o resultados esperados	Prototipado o sesiones de retroalimentación

Fuente: Adaptado de Echeverri et al. (2013)

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

Métodos formales: Los métodos formales son técnicas basadas en lógica matemática y fundamentos teóricos que permiten representar requisitos con precisión, consistencia y verificabilidad. Son especialmente útiles en proyectos donde los errores pueden causar consecuencias graves, como en sistemas bancarios, médicos, aeronáuticos o de defensa.

Propósitos de los métodos formales

- Eliminar ambigüedades y contradicciones en los requisitos.
- Asegurar la completitud y consistencia de los modelos.
- Permitir la verificación automática de las especificaciones.
- Establecer una base sólida para pruebas, diseño y desarrollo formal posterior.

Tabla 6

Clasificación de métodos formales

Tipo	Descripción
Notaciones algebraicas	Describen el sistema con ecuaciones y estructuras algebraicas (ej. VDM, B).
Lenguajes de especificación	Permiten describir comportamientos mediante lógica y conjuntos (ej. Z, Alloy).
Modelos basados en lógica temporal	Representan sistemas reactivos y concurrentes (ej. TLA+, LTL).

Tabla 7

Herramientas para trabajar con métodos formales

Herramienta	Uso	Lenguaje compatible
Alloy Analyzer	Verificación de modelos de relaciones	Alloy
Z/EVES	Análisis de modelos Z	Lenguaje Z
TLA+ Toolbox	Model checking de sistemas concurrentes	TLA+
Rodin Platform	Verificación formal con pruebas automáticas	Método B

Supongamos que queremos expresar el requisito:

"Un médico no puede tener más de una cita a la misma hora".

En Alloy podríamos modelarlo así:

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

Tabla 8

Ejemplo práctico con Alloy (resumen textual)

```
sig Medico {} sig Paciente {} sig Cita {
medico: one Medico, paciente: one Paciente, hora: one Int}

fact SinSolapamiento { all c1, c2: Cita |c1.medico = c2.medico and c1.hora =
c2.hora implies c1= c2}
```

Este modelo verifica automáticamente si existe alguna violación a la restricción de solapamiento. Esta es la potencia de los métodos formales.

Requisitos en entornos ágiles

En proyectos ágiles, como aquellos que aplican Scrum, Kanban, XP o SAFe, los requisitos se gestionan de manera incremental y colaborativa, adaptándose a los cambios constantes del cliente o el entorno.

Características de la gestión de requisitos en entornos ágiles

- Evolutivos: no se documentan todos los requisitos al inicio.
- Prioritarios: se organizan en función del valor para el cliente.
- Flexibles: el cambio se considera parte natural del proceso.
- Conversacionales: la documentación mínima es suficiente, se valora más la conversación que los documentos.

Artefactos clave

Tabla 9

Artefactos clave

Artefacto ágil	Propósito
Historias de usuario	Capturan funcionalidades desde la perspectiva del usuario final.
Criterios de aceptación	Definen cuándo una historia se considera terminada.
Backlog del producto	Lista priorizada de requisitos pendientes por implementar.
Tablero Kanban / Sprint board	Visualiza el estado actual de cada ítem de trabajo.
Definición de "Hecho"	Estándar que define cuándo un ítem está completamente terminado.

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

Estructura de una historia de usuario

Como [tipo de usuario], quiero [funcionalidad] para [beneficio].

Ejemplo:

Como paciente, quiero consultar la disponibilidad médica desde mi celular, para poder agendar citas fácilmente.

Criterios de aceptación (ejemplo asociado)

- El paciente puede filtrar por especialidad médica.
- El sistema debe mostrar al menos los horarios disponibles de los próximos 7 días.
- El sistema debe enviar una confirmación inmediata al correo del paciente.

Técnicas ágiles complementarias

- Personas (User Personas): arquetipos de usuarios que guían el diseño centrado en el usuario.
- Story Mapping: visualización jerárquica de historias para estructurar releases.
- BDD (Behavior Driven Development): usa escenarios dados/cuando/entonces para definir comportamiento esperado.

Riesgos frecuentes al documentar en entornos ágiles

- Poca trazabilidad entre historia y funcionalidad real.
- Exceso de confianza en la comunicación oral.
- Falta de criterios formales que garanticen calidad.

Solución: combinar ligereza documental con estándares mínimos de formalidad, apoyándose en herramientas como Jira, Trello, Notion, Confluence, GitLab, Azure DevOps, entre otras.

b) Análisis de los Resultados: Plataforma Web de Gestión de Tutorías Académicas para la ESPOCH

En este caso práctico, se implementó un proceso de ingeniería de requisitos para una plataforma web destinada a la gestión de tutorías académicas en la Escuela Superior Politécnica de Chimborazo (ESPOCH). A través de la recolección, análisis, y validación de los requisitos, se mejoró significativamente la trazabilidad de las tutorías y la gestión documental de este proceso tradicionalmente deficiente.

1. Actividades realizadas y resultados

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

Durante la fase de elicitación de requisitos, se utilizaron diversas técnicas como entrevistas semiestructuradas, observación directa, cuestionarios en línea, y prototipos de baja fidelidad. Esto permitió obtener un conjunto claro y conciso de requisitos funcionales y no funcionales para la plataforma. A continuación se detallan los principales resultados.

2. Requisitos funcionales identificados

Los requisitos funcionales fueron claves para definir la estructura del sistema. Algunos de los más importantes fueron:

- Solicitar tutorías: Los estudiantes deben poder solicitar tutorías a través de un formulario web.
- Reprogramación de sesiones: Los docentes deben tener la capacidad de aceptar, rechazar o reprogramar las tutorías solicitadas por los estudiantes.
- Registros y observaciones: El sistema debe permitir a los docentes registrar observaciones y evidencias de cada sesión de tutoría.
- Reportes académicos: Los usuarios (estudiantes y docentes) deben poder descargar reportes sobre las tutorías realizadas, organizados por ciclo académico.

3. Requisitos no funcionales identificados

Además de los requisitos funcionales, se definieron ciertos requisitos no funcionales importantes para asegurar la calidad y accesibilidad del sistema:

- Integración con el calendario institucional: Las tutorías deben integrarse de manera eficiente con el calendario académico de la ESPOCH.
- Accesibilidad: El sistema debe ser accesible desde dispositivos móviles y cumplir con las pautas WCAG 2.0 (nivel AA).
- Rendimiento: El tiempo de respuesta para registrar o acceder a una tutoría debe ser inferior a 2 segundos para asegurar una experiencia de usuario ágil y eficiente.

4. Actividades de análisis y negociación

En la fase de análisis, se realizó una revisión cruzada de requisitos para detectar posibles inconsistencias o ambigüedades. También se utilizó una matriz de trazabilidad para asegurar que

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

todos los requisitos estuvieran cubiertos por las funcionalidades del sistema y las pruebas asociadas.

Tabla 10

Problemas detectados y soluciones aplicadas

Problema	Solución Aplicada
Requisitos ambiguos (“fácil de usar”)	Reformulación usando criterios más específicos (Ej. “Interfaz con máximo 3 clics para agendar una tutoría”).
Conflictos de agenda docente/estudiante	Implementación de una validación automática de disponibilidad cruzada para evitar conflictos de horarios.
Redundancia entre módulos de gestión y seguimiento	Integración de funcionalidades duplicadas en una única opción de "bitácora de tutoría".

5. Documentación de los Requisitos

La documentación de los requisitos se estructuró siguiendo la norma ISO/IEC/IEEE 29148:2018, lo que permitió asegurar la calidad y formalidad de los entregables. Se utilizaron plantillas para la creación de la Especificación de Requisitos de Software (SRS), que contenían los siguientes componentes:

- Introducción
- Descripción general del sistema
- Requisitos funcionales
- Requisitos no funcionales
- Restricciones y supuestos
- Trazabilidad

Además, se utilizó UML (Lenguaje Unificado de Modelado) para representar los casos de uso, y se elaboraron tablas de decisión para las reglas de programación de tutorías.

6. Validación de los requisitos

Durante la fase de validación, se realizaron walkthroughs y revisiones técnicas para garantizar la correcta interpretación y viabilidad de los requisitos. Se validaron todos los requisitos con los

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

usuarios finales a través de prototipos, asegurando que estos cumplían con las expectativas y necesidades del público objetivo.

Se emplearon historias de usuario como una técnica ágil para validar los requisitos con los stakeholders:

Ejemplo de Historia de Usuario:

- Como estudiante,
- Quiero agendar una tutoría desde mi panel,
- Para recibir orientación académica sin necesidad de trámites presenciales.

Criterios de Aceptación:

- La cita debe ser confirmada por el docente vía sistema.
- Se debe generar una alerta en el correo institucional.
- Se debe registrar una bitácora con la fecha, hora y tema tratado.

7. Matriz de Trazabilidad de Requisitos

Se desarrolló una matriz de trazabilidad para asegurar que todos los requisitos funcionales y no funcionales fueran implementados correctamente y verificables mediante pruebas. Esta matriz permitió realizar un seguimiento detallado de los requisitos a lo largo de todo el ciclo de vida del proyecto.

Tabla 11

Matriz de Trazabilidad de Requisitos

ID Req.	Descripción	Fuente	Implementación	Prueba asociada	Estado
RF001	Login con verificación en dos pasos	Cliente	Módulo Login	TC001	Validado
RNF003	Tiempo de carga < 2 segundos	Normativa	Front-end	TC015	En revisión
RF014	Reporte de usuarios en PDF	Jefe de área	Módulo Reportes	TC018	Aprobado

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

7. Resultados Esperados y Beneficios del Sistema

Con base en los requisitos identificados y la documentación realizada, los resultados esperados del proyecto fueron:

- Reducción del 80% en las sesiones no registradas, mejorando la trazabilidad de las tutorías.
- Trazabilidad completa de cada tutoría registrada, desde su solicitud hasta la evaluación final.
- Cumplimiento con los estándares académicos de la ESPOCH y las normativas de seguimiento académico, garantizando la calidad y coherencia del proceso de tutorías.

8. Herramientas Utilizadas

Para la documentación, gestión y validación de requisitos, se utilizaron diversas herramientas:

- Draw.io / Lucidchart: Modelado de casos de uso UML.
- Jira / Trello: Documentación y priorización de historias.
- Confluence / Google Docs: Elaboración colaborativa del SRS.
- Excel: Matrices de trazabilidad y priorización.

El proceso de análisis y documentación de requisitos en la plataforma de gestión de tutorías académicas de la ESPOCH permitió la creación de una solución eficaz, alineada con las necesidades de los usuarios y los objetivos estratégicos del sistema. La combinación de herramientas ágiles y técnicas tradicionales garantizó la precisión de los requisitos, así como la satisfacción de los stakeholders al cumplir con sus expectativas.

Análisis de resultados global

La implementación práctica de la ingeniería de requisitos dentro del contexto educativo universitario revela resultados significativos que trascienden la simple aplicación técnica de metodologías. A través del estudio de caso basado en el diseño de un sistema de reservas médicas y una plataforma de gestión de tutorías académicas, se constata que el uso adecuado de técnicas de elicitación y validación permite mejorar sustancialmente la calidad de los entregables.

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

En el caso del sistema de reservas médicas, la aplicación combinada de entrevistas, observación directa, talleres y cuestionarios permitió identificar tanto requisitos funcionales como no funcionales con un alto grado de precisión. Esta diversidad metodológica enriqueció la comprensión de las necesidades del cliente y facilitó el diseño de una solución centrada en la experiencia del usuario. El tiempo de carga inferior a 3 segundos, las notificaciones automatizadas y la accesibilidad en dispositivos móviles son ejemplos de requisitos derivados directamente de esta aproximación centrada en el usuario.

En cuanto al sistema de tutorías académicas, los resultados muestran una mejora significativa en la trazabilidad de las sesiones y en la gestión documental, tradicionalmente deficiente en entornos universitarios. Se documentaron más de 10 funcionalidades clave, validadas mediante prototipos y priorizadas con base en herramientas ágiles como Jira y matrices de decisión. La incorporación de estándares como ISO/IEC/IEEE 29148:2018 aseguró la calidad formal de los requisitos especificados.

Ambos proyectos pusieron en evidencia la necesidad de combinar métodos formales (documentación estructurada, control de cambios, trazabilidad) con enfoques ágiles (validación continua, historias de usuario, priorización dinámica). Esta sinergia no solo garantizó una mayor precisión en la definición de requisitos, sino que también fortaleció la participación activa de los usuarios en el proceso de diseño, aumentando el sentido de pertenencia hacia el sistema.

Discusión

El ejercicio práctico desarrollado en los casos analizados permite discutir con mayor profundidad la relevancia de una formación sólida en ingeniería de requisitos dentro de las carreras de tecnología y software. La integración de estándares internacionales, como ISO/IEC/IEEE 29148:2018 y BABOK v3, no solo mejora la calidad técnica del producto, sino que también proporciona una estructura que permite gestionar la complejidad de proyectos reales (International Institute of Business Analysis, 2015; ISO/IEC/IEEE, 2018).

Asimismo, se evidencia que la ingeniería de requisitos debe asumir un enfoque humanista, que priorice la comprensión empática de las necesidades del usuario por sobre la rigurosidad formal aislada. Este balance entre técnica y humanidad es lo que diferencia a un buen sistema de uno que

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

simplemente "funciona". Tal como indican Arangurí y Bravo (2021), la calidad del sistema no puede ser mayor que la calidad de los requisitos sobre los cuales se construye.

En los entornos universitarios y de salud, donde los usuarios finales no siempre dominan el lenguaje técnico, el rol del analista como traductor, facilitador y mediador se vuelve clave. Este hallazgo reafirma lo señalado por Bejarano y Baquero (2020), quienes advierten que muchos proyectos fracasan no por falta de recursos tecnológicos, sino por una deficiente comunicación y gestión de expectativas.

También es importante señalar que la adopción de herramientas tecnológicas (como Jira, Trello, PlantUML, Lucidchart) no garantiza automáticamente el éxito del proceso de requisitos. Lo crucial es la formación de competencias: pensamiento crítico, habilidades comunicativas, ética profesional y visión estratégica. En este sentido, los resultados obtenidos apoyan la idea de que la ingeniería de requisitos no puede enseñarse únicamente como una técnica, sino como una competencia transversal.

Conclusiones

1. La ingeniería de requisitos es una disciplina estratégica que debe ocupar un lugar central en la formación de ingenieros de software, ya que permite conectar los objetivos del negocio con las soluciones tecnológicas desarrolladas.
2. La combinación de técnicas de elicitación y estándares internacionales —como ISO/IEC/IEEE 29148:2018 y BABOK v3— mejora significativamente la calidad, trazabilidad y validación de los requisitos, reduciendo errores posteriores en el ciclo de desarrollo.
3. La participación activa de los stakeholders, mediante métodos colaborativos como talleres, prototipos y entrevistas, es fundamental para construir sistemas alineados con las verdaderas necesidades del usuario final.
4. La ingeniería de requisitos debe integrar enfoques ágiles y tradicionales, adaptándose al contexto del proyecto. La flexibilidad metodológica permite una mejor respuesta ante el cambio y favorece la mejora continua.
5. Los entornos educativos y de salud presentan desafíos específicos, como la falta de documentación previa o la baja alfabetización tecnológica de los usuarios. Esto exige un

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

perfil de analista con habilidades blandas, comunicación empática y capacidad de negociación.

6. La gestión de requisitos no es solo una tarea técnica, sino también una práctica ética y profesional que contribuye al desarrollo de sistemas inclusivos, sostenibles y centrados en las personas. Su enseñanza debe basarse en la resolución de problemas reales, el trabajo interdisciplinario y el uso de herramientas actuales.

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

Referencias bibliográficas

Administración de Requerimientos. (2014, agosto 26). *Clasificación y tipos de requerimientos*. WordPress. https://administracionderequerimientos.wordpress.com/2014/08/26/clasificacion-y-tipos-de-requerimientos/

Arangurí, M., & Bravo, J. (2021). *Metodología integral de casos de prueba sustentado en un modelo de verificación de requisitos*. eLibro. https://elibro.net/es/lc/espoch/titulos/228963

Bejarano, M., & Baquero, L. (2020). *Ciclo de vida de desarrollo ágil de software seguro*. eLibro. https://elibro.net/es/lc/espoch/titulos/197008

Echeverri, J., Aristizábal, M., & González, L. (2013). *Reflexiones sobre ingeniería de requisitos y pruebas de software*. eLibro. https://elibro.net/es/lc/espoch/titulos/68913

Jiménez, C. (2021). *UML: Arquitectura de aplicaciones en Java, C++ y Python*. eLibro. https://elibro.net/es/lc/espoch/titulos/2227203

IEEE. (1998). *IEEE Recommended Practice for Software Requirements Specifications* (Std 830-1998). Institute of Electrical and Electronics Engineers.

ISO/IEC/IEEE. (2018). *Systems and software engineering — Life cycle processes — Requirements engineering* (ISO/IEC/IEEE 29148:2018). International Organization for Standardization.

International Institute of Business Analysis. (2015). *A guide to the business analysis body of knowledge (BABOK guide)* (Version 3). IIBA.

ISO/IEC. (2011). *ISO/IEC 25010:2011. Systems and software engineering — Systems and software quality requirements and evaluation (SQuaRE) — System and software quality models*. International Organization for Standardization.

PMI Levante. (s.f.). *Gestión de los interesados*. https://pmi-levante.org/gestion-interesados/

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional

Conflicto de intereses:

Los autores declaran que no existe conflicto de interés posible.

Financiamiento:

No existió asistencia financiera de partes externas al presente artículo.

Agradecimiento:

N/A

Nota:

El artículo no es producto de una publicación anterior.

Esta obra está bajo una Licencia Creative Commons Atribución No Comercial-Sin Derivar 4.0 Internacional