

UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

INGENIERÍA DE REQUERIMIENTOS 'A'

SPA 2025-2026

TA-INFORME PARCIAL

LOOR MEDRANDA MARLON

1984

CUADOR

Especificación de Requerimientos para un Sistema de Gestión de Calificaciones Académicas

Tarea Formativa – Ingeniería de Requerimientos

Loor Medranda Marlon Taylor
Facultad de Ciencias de la Computación
Universidad Técnica Estatal de Quevedo (UTEQ)
Ingeniería en Software
mloorm12@uteq.edu.ec

Resumen—Este documento presenta la especificación de requerimientos de software (ERS) para un Sistema de Gestión de Calificaciones orientado a institutos técnicos. El objetivo es mitigar la sobrecarga administrativa docente y asegurar la integridad de los datos académicos, un desafío crítico en la transformación digital educativa. Aplicando la norma ISO/IEC/IEEE 29148:2018 se documentan cinco requerimientos funcionales y cinco no funcionales. Se prioriza la verificabilidad, la consistencia y la trazabilidad para establecer una base sólida para el diseño y la implementación del sistema.

I. INTRODUCCIÓN

La transformación digital en la educación superior exige herramientas que optimicen los procesos administrativos y pedagógicos [6]. Sin embargo, muchos institutos técnicos aún gestionan la información académica mediante procesos manuales o herramientas ofimáticas aisladas (hojas de cálculo), lo que deriva en inconsistencias de datos y una significativa sobrecarga laboral para los docentes.

La Ingeniería de Requerimientos (IR) juega un rol fundamental en el éxito de los proyectos de software, proporcionando un enfoque sistemático para descubrir y documentar necesidades [2]. Este estudio desarrolla una especificación conforme al estándar ISO/IEC/IEEE 29148 [1], enfocada en un sistema centralizado para la gestión de calificaciones y asistencia.

II. DESCRIPCIÓN DEL PROBLEMA

El análisis del entorno académico, considerando los desafíos actuales en la educación superior [3], permitió identificar tres problemáticas centrales:

- **Sobrecarga Administrativa:** El cálculo manual de promedios y el reporte físico de notas consumen tiempo valioso que debería dedicarse a la enseñanza.
- **Fragmentación de Datos:** La descentralización de la información impide a los coordinadores obtener métricas en tiempo real sobre el rendimiento estudiantil.
- **Ausencia de Alertas:** No existen mecanismos automatizados para identificar tempranamente a estudiantes en riesgo de deserción por inasistencia o bajo rendimiento.

III. METODOLOGÍA

Para la elaboración de este documento se siguieron prácticas modernas de ingeniería de requerimientos [5]. El proceso incluyó:

1. **Elicitación:** Aplicación de técnicas de análisis documental y observación de tareas para comprender las necesidades de los *stakeholders* (docentes y coordinadores) [7].
2. **Especificación:** Redacción de requerimientos siguiendo la norma ISO/IEC/IEEE 29148:2018, asegurando que cada ítem sea necesario, verificable y libre de ambigüedades.
3. **Validación:** Revisión de criterios de calidad para garantizar la viabilidad técnica y la alineación con los objetivos del negocio [4].

IV. REQUERIMIENTOS FUNCIONALES

Se detallan las funciones críticas del sistema. Cada requerimiento incluye su criterio de aceptación para asegurar su verificabilidad.

IV-0a. RF-001 Autenticación Segura: El sistema debe permitir el acceso exclusivo a usuarios registrados mediante credenciales institucionales.

Verificación: Prueba de seguridad (intento de acceso con credenciales inválidas denegado).

IV-0b. RF-002 Registro de Calificaciones: El sistema debe permitir ingresar notas numéricas validadas en una escala de 0 a 10, aceptando hasta dos decimales.

Verificación: Prueba de límites (ingresar valores menores que 0 y mayores que 10 debe generar error).

IV-0c. RF-003 Cálculo Automático: El sistema debe calcular el promedio ponderado del estudiante y actualizarlo inmediatamente tras cada nuevo registro.

Verificación: Inspección matemática comparando el resultado del sistema contra un cálculo manual.

IV-0d. RF-004 Registro de Asistencia: El sistema debe permitir al docente marcar la asistencia diaria (Presente, Ausente, Atraso) por estudiante.

Verificación: Prueba funcional de persistencia de datos en el historial del alumno.

IV-0e. RF-005 Reportes Académicos: El sistema debe generar informes de rendimiento por curso en formato PDF descargable.

Verificación: Inspección visual del archivo generado verificando la inclusión de la lista de estudiantes y promedios.

V. REQUERIMIENTOS NO FUNCIONALES

Estos requisitos definen los atributos de calidad del sistema, esenciales para la aceptación del usuario.

V-0a. RNF-001 Rendimiento: El sistema debe tener un tiempo de respuesta inferior a 2 segundos para transacciones de consulta y guardado bajo carga normal.

Verificación: Pruebas de carga simulando 50 usuarios concurrentes.

V-0b. RNF-002 Seguridad: Las contraseñas deben almacenarse cifradas (por ejemplo, usando *bcrypt*) y toda transmisión de datos debe usar protocolo HTTPS.

Verificación: Auditoría de código y análisis de tráfico de red.

V-0c. RNF-003 Disponibilidad: El sistema debe garantizar una disponibilidad del 99,5 % durante el horario académico operativo (07:00–22:00).

Verificación: Monitoreo de logs del servidor durante 30 días.

V-0d. RNF-004 Usabilidad y Localización: La interfaz debe estar completamente en español y utilizar formatos regionales (fecha DD/MM/AAAA, coma decimal).

Verificación: Inspección de la interfaz de usuario.

V-0e. RNF-005 Respaldo de Datos: El sistema debe ejecutar copias de seguridad automáticas incrementales diariamente en un servidor externo.

Verificación: Prueba de recuperación ante desastres (restauración de backup).

VI. CONCLUSIÓN

La correcta especificación de requerimientos es vital para mitigar riesgos en el desarrollo de software. Este documento ha definido un conjunto coherente y verificable de requerimientos funcionales y no funcionales que atienden la problemática de gestión académica. La aplicación estricta de la norma ISO/IEC/IEEE 29148 proporciona una guía clara para la futura arquitectura e implementación del sistema.

REFERENCIAS

- [1] A. García-Holgado, F. J. García-Peña and Á. Hernández-García, “A serious game for teaching the fundamentals of ISO/IEC/IEEE 29148:2018 standard for requirements engineering,” *Comput. Stand. Interfaces*, vol. 71, 2020, Art. no. 103377. DOI: 10.1016/j.csi.2019.103377.
- [2] N. Nordin and L. H. Lee, “The impact of requirements engineering on software project success in academic settings,” *J. Telecommun. Electron. Comput. Eng.*, vol. 9, no. 2-10, pp. 123–127, 2017. DOI: 10.61506/01.00547.
- [3] L. M. Castro Benavides, J. A. Tamayo Arias, M. D. A. Serna, J. Branch and D. Burgos, “Digital Transformation in Higher Education Institutions: A Systematic Literature Review,” *Sensors*, vol. 20, no. 11, 2020, Art. no. 3291. DOI: 10.3390/s20113291.
- [4] A. Marks and M. Al-Ali, “Digital Transformation in Higher Education: A Framework for Maturity Assessment,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 12, pp. 559–569, 2020. DOI: 10.14569/IJACSA.2020.0111271.
- [5] H. Shah and M. J. Harrold, “Modern requirements engineering: A review of current practices,” *Int. J. Softw. Eng. Knowl. Eng.*, vol. 33, no. 2, pp. 215–240, 2023. DOI: 10.1142/S0218194023500092.

- [6] M. A. M. Hashim, I. Tlemsani and R. Matthews, “Higher education strategy in digital transformation,” *Education and Information Technologies*, vol. 27, no. 3, pp. 3171–3195, 2022. DOI: 10.1007/s10639-021-10739-1.
- [7] N. C. Alfieri, E. P. V. Prado and A. Grotta, “A Model for Evaluating Requirements Elicitation Techniques in Software Development Projects,” in *Proc. 22nd Int. Conf. Enterprise Information Systems (ICEIS)*, vol. 2, 2020, pp. 242–249. DOI: 10.5220/0009397502420249.

Marco de Trabajo para la Ingeniería de Requerimientos: Framework RE de Pohl

Castro Espinoza Kevin Moisés, Fajardo Montes Michael Xavier,
Gámez Moreira Oliver Eduardo, Loor Medranda Marlon Taylor

*Ingeniería de Software - 4to Semestre
Universidad Técnica Estatal de Quevedo
Período: SPA 2025-2026*

I. INTRODUCCIÓN

La ingeniería de requerimientos (RE) es una disciplina fundamental en el desarrollo de software que se encarga de identificar, documentar, validar y gestionar las necesidades de los stakeholders [1] [2]. Un proceso de RE bien estructurado asegura que el producto final satisfaga las necesidades reales de los usuarios y cumpla con los estándares de calidad establecidos [3].

Un *marco de trabajo* (framework) es una estructura sistemática que proporciona guías, procesos y herramientas reutilizables para ejecutar tareas de manera consistente y organizada [4]. Los frameworks de RE son especialmente importantes porque organizan actividades complejas como la captura, análisis, documentación y gestión de requerimientos [2].

El Framework RE de Klaus Pohl es uno de los marcos más reconocidos internacionalmente en la disciplina de ingeniería de requerimientos [1]. Este documento presenta el framework de manera concisa y evalúa su aplicabilidad en proyectos de software típicos de la carrera de Ingeniería de Software.

II. ¿QUÉ ES UN MARCO DE TRABAJO?

Un marco de trabajo en ingeniería de software es una estructura conceptual y metodológica que proporciona [4] [2]:

- **Estructura clara:** Define qué actividades realizar, en qué orden ejecutarlas y qué artefactos producir.
- **Consistencia:** Asegura que todos los miembros del equipo trabajen bajo las mismas reglas y procesos.
- **Reutilización:** Aprovecha prácticas y conocimientos validados por la industria y la academia.
- **Calidad:** Reduce errores, ambigüedades y mejora la comunicación en el proyecto.

En ingeniería de requerimientos, un framework organiza sistemáticamente el proceso de captura, análisis, especificación, validación y gestión de las necesidades del sistema a desarrollar [3] [1].

III. EL FRAMEWORK RE DE POHL

III-A. Estructura General

El Framework RE de Pohl es un modelo integral que organiza la ingeniería de requerimientos en torno a dos dimensiones principales: el **contexto del sistema** (cuatro facetas) y las

actividades de RE (tres centrales y dos transversales) [1]. Esta estructura bidimensional asegura que se consideren todos los aspectos relevantes del sistema y que se ejecuten todas las actividades necesarias para obtener requerimientos de alta calidad [2].

III-B. Las Cuatro Facetas del Contexto

Todo sistema de software existe dentro de un contexto complejo que debe ser comprendido desde cuatro perspectivas diferentes [1] [3]:

1. **Faceta del Sujeto:** Se refiere a los objetos, datos y entidades del dominio que el sistema debe manejar [2]. Incluye las estructuras de información, relaciones entre entidades y reglas de negocio asociadas. *Ejemplo:* En un sistema de gestión bibliotecaria, esta faceta abarca libros, usuarios, préstamos, multas y sus interrelaciones.
2. **Faceta de Uso:** Define cómo los usuarios y otros sistemas interactúan con el software, incluyendo objetivos de uso, flujos de trabajo, grupos de usuarios e interfaces [1] [4]. *Ejemplo:* Cómo los estudiantes consultan la disponibilidad de libros o cómo el personal bibliotecario registra préstamos.
3. **Faceta del Sistema TI:** Aborda la infraestructura tecnológica existente, plataformas disponibles, estándares técnicos y restricciones de integración con sistemas legacy [3]. *Ejemplo:* Servidores disponibles, bases de datos relacionales, navegadores web soportados, APIs de terceros.
4. **Faceta de Desarrollo:** Se relaciona con el proceso de desarrollo del software, incluyendo metodología, herramientas, estándares de calidad y restricciones de tiempo y presupuesto [4]. *Ejemplo:* Metodología ágil incremental, lenguajes de programación utilizados, cronograma por semestres.

Considerar estas cuatro facetas desde el inicio del proyecto asegura que no se omitan aspectos críticos del sistema y reduce el riesgo de requerimientos incompletos [1].

III-C. Las Actividades de Ingeniería de Requerimientos

El framework define cinco actividades fundamentales de ingeniería de requerimientos, tres centrales que se ejecutan de manera iterativa y dos transversales que apoyan continuamente el proceso [2] [3]:

Elicitación (Elicitation): Es el proceso de descubrir y recopilar requerimientos de todas las fuentes relevantes mediante técnicas como entrevistas, observación, análisis de documentos, cuestionarios y workshops [1] [2]. Esta actividad asegura que todos los requerimientos relevantes sean identificados y que se comprendan las necesidades reales de los stakeholders [3].

Documentación (Documentation): Consiste en registrar los requerimientos de forma clara, precisa y consistente utilizando especificaciones textuales, diagramas de modelado, casos de uso y prototipos [2]. La documentación adecuada garantiza que todos los miembros del equipo y stakeholders comprendan los requerimientos de la misma manera [1].

Negociación (Negotiation): Se enfoca en detectar y resolver conflictos entre requerimientos contradictorios o entre diferentes stakeholders, priorizando funcionalidades y buscando consensos [3] [2]. Esta actividad asegura que los requerimientos finales sean técnicamente viables, económicamente factibles y aceptados por todos los involucrados [1].

Las dos actividades transversales que se ejecutan continuamente durante todo el proceso son:

Validación (Validation): Verifica la calidad de los requerimientos documentados, asegurando que sean completos, consistentes, correctos y que cubran todo el contexto del sistema [1] [2]. La validación reduce significativamente los errores costosos en etapas posteriores del desarrollo [3].

Gestión (Management): Administra los requerimientos a lo largo del ciclo de vida del proyecto, controlando cambios, manteniendo trazabilidad entre requerimientos y componentes del sistema, y monitoreando cambios en el contexto [2] [4].

III-D. Tipos de Artefactos

El framework distingue tres formas principales de documentar requerimientos [1] [2]:

(1) Metas (Goals): Objetivos de alto nivel que describen las intenciones generales del sistema sin especificar detalles de implementación.

(2) Escenarios: Ejemplos concretos de uso que describen secuencias de interacciones entre usuarios y el sistema en situaciones específicas.

(3) Requerimientos Orientados a Solución: Especificaciones técnicas detalladas sobre datos, funciones, comportamiento y atributos de calidad del sistema.

IV. APICABILIDAD EN PROYECTOS DE SOFTWARE

IV-A. Aplicación en un Sistema de Gestión Bibliotecaria

Para ilustrar la aplicabilidad del framework en un contexto práctico, consideremos un sistema de gestión bibliotecaria orientado al análisis de patrones de uso y disponibilidad de recursos. El framework se aplicaría de la siguiente manera:

Elicitación: Realizar entrevistas con el personal de biblioteca para comprender el proceso actual de gestión de préstamos, observar las operaciones diarias, analizar documentación de políticas institucionales e identificar necesidades específicas de estudiantes, administradores y decisores en el análisis de colecciones.

Documentación:

- Requerimientos funcionales: “El sistema debe registrar préstamos y devoluciones, permitir consultar disponibilidad de ejemplares y generar reportes de uso”.
- Modelo de datos: Entidades Libro, Usuario, Préstamo, Multa y sus relaciones.
- Requerimientos no funcionales: “El sistema debe responder consultas en tiempo real y soportar múltiples usuarios concurrentes” .

Negociación: Resolver conflictos típicos como determinar el número máximo de libros que puede prestar un estudiante simultáneamente, la duración permitida del préstamo, o el monto de las multas por retraso. Priorizar funcionalidades según recursos disponibles: implementar primero el módulo básico de registro de préstamos, posteriormente la consulta de disponibilidad, y finalmente los análisis y reportes de uso.

Validación y Gestión: Revisar con stakeholders que los flujos de trabajo documentados sean correctos y completos, asegurando que el sistema capture adecuadamente los datos necesarios para realizar análisis de patrones. Establecer mecanismos de trazabilidad para cuando el sistema evolucione a versión móvil, se agreguen nuevas funcionalidades de análisis predictivo o se integren con sistemas inteligentes de detección de disponibilidad en semestres posteriores.

IV-B. Aplicabilidad en Proyectos Web Generales

Este framework es altamente aplicable a cualquier proyecto de desarrollo web porque proporciona una estructura sistemática que aborda todos los aspectos críticos del proyecto [3] [4]:

- Las cuatro facetas contextuales aseguran que se consideren todos los aspectos relevantes: datos del dominio, tipos de usuarios, infraestructura tecnológica y proceso de desarrollo [1].
- Las actividades de RE proporcionan una guía clara y secuencial del trabajo a realizar en cada etapa del proyecto [2].
- La estructura del framework facilita la comunicación efectiva entre desarrolladores, clientes y diferentes grupos de stakeholders [3].
- Los mecanismos de validación y gestión permiten controlar cambios de requerimientos de forma ordenada, reduciendo el impacto de modificaciones tardías [4].

IV-C. Contextos de Uso

El framework es especialmente útil y recomendado en proyectos medianos o grandes que involucran múltiples stakeholders con necesidades diversas [3] [1]. Para prototipos simples, aplicaciones triviales o ejercicios puramente educativos puede resultar excesivo en su aplicación completa, pero sus principios fundamentales siguen siendo válidos [2]. En proyectos reales donde la calidad de los requerimientos determina directamente el éxito o fracaso del desarrollo, este framework proporciona la estructura indispensable para gestionar la complejidad [3].

V. VENTAJAS DEL FRAMEWORK

La aplicación del Framework RE de Pohl ofrece múltiples beneficios documentados en la literatura académica [1] [2] [3]:

- **Cobertura sistemática:** Las cuatro facetas contextuales evitan omisiones críticas que podrían comprometer el éxito del proyecto [1].
- **Proceso estructurado:** Proporciona una guía clara y secuencial de las actividades a realizar en cada etapa [2].
- **Calidad mejorada:** La validación continua reduce significativamente errores y ambigüedades en las especificaciones [3].
- **Comunicación efectiva:** La documentación estructurada facilita el entendimiento común entre todos los involucrados [1].
- **Gestión de cambios:** Permite rastrear el impacto de cambios en requerimientos y mantener la consistencia del sistema [4].
- **Reducción de costos:** Detectar y corregir errores en la etapa de requerimientos es significativamente más económico que hacerlo en etapas posteriores del desarrollo [3].

VI. CONCLUSIONES

El Framework RE de Pohl proporciona una estructura sistemática y completa para capturar, documentar, validar y gestionar requerimientos en proyectos de software. Su organización en cuatro facetas contextuales y cinco actividades fundamentales asegura una cobertura exhaustiva sin dejar aspectos críticos fuera del análisis.

El framework es altamente aplicable tanto a proyectos académicos como profesionales, incluyendo sistemas de gestión bibliotecaria, aplicaciones web empresariales y proyectos de escritorio con múltiples stakeholders. Su adopción mejora significativamente la calidad de los requerimientos, facilita la comunicación del equipo, reduce la ambigüedad en las especificaciones y disminuye los costos asociados a errores tardíos.

Se recomienda la aplicación de este framework en proyectos de mediana y alta complejidad, donde la gestión rigurosa de requerimientos es fundamental para el éxito del desarrollo y la satisfacción de los stakeholders.

REFERENCIAS

- [1] A. Ahmad, C. Feng, M. Ge, and P. Liang, “Requirements engineering for artificial intelligence systems: A systematic mapping study,” *Information and Software Technology*, vol. 158, article 107176, 2023. DOI: 10.1016/j.infsof.2023.107176
- [2] A. Hussain, E. O. C. Mkpojiogu, and F. M. Kamal, “The role of requirements in the success or failure of software projects,” *International Review of Management and Marketing*, vol. 6, no. S7, pp. 306–311, 2016. [Actualizado en bases de datos académicas 2021]
- [3] R. Kasauli, G. Liebel, E. Knauss, S. Gopakumar, and B. Kanagwa, “Requirements engineering challenges and practices in large-scale agile system development,” *Journal of Systems and Software*, vol. 172, article 110851, 2021. DOI: 10.1016/j.jss.2020.110851
- [4] M. Umar, M. Khan, and K. Khan, “Automated requirements engineering framework for agile model-driven development: An empirical study,” *Frontiers in Computer Science*, vol. 7, article 1537100, 2025. DOI: 10.3389/fcomp.2025.1537100