

Libro de Actas

JCSA 2025

Jornadas de Calidad de Software y Agilidad

27 y 28 de noviembre de 2025
Facultad Regional Resistencia



UNIVERSIDAD
TECNOLÓGICA NACIONAL
FACULTAD REGIONAL
RESISTENCIA



Universidad Nacional del Nordeste
Facultad de Ciencias
Exactas y Naturales y Agrimensura



Universidad Nacional de Misiones
Facultad de Ciencias
Exactas, Químicas y Naturales

Jornadas de Calidad de Software y Agilidad : JCSA 2025 / Mateo Beneyto ... [et al.] ;
Compilación de Verónica A. Bollati ; Noelia S. Pinto ; César Acuña ; Coordinación
general

de Verónica A. Bollati. - 1a ed. - Ciudad Autónoma de Buenos Aires : Universidad
Tecnológica Nacional, 2026.

Libro digital, PDF

Archivo Digital: descarga y online

ISBN 978-950-42-0266-0

1. Ingeniería de Software. 2. Software. I. Beneyto, Mateo II. Bollati, Verónica A.,
comp. III. Pinto, Noelia S., comp. IV. Acuña, César, comp. V. Bollati, Verónica A., coord.
CDD 005.30688

Libro de Actas

Jornadas de Calidad de Software y Agilidad

Fecha del Congreso: 27 y 28 de noviembre de 2025

Lugar de las Jornadas: Universidad Tecnológica Nacional, Facultad Regional Resistencia, Chaco, Argentina.

DOI: <https://doi.org/10.33414/ajea.1999.2026>

ISBN: 978-950-42-0266-0

Fecha de Publicación: 04/2026

ISBN 978-950-42-0266-0



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional.



Jornadas de Calidad de Software y Agilidad

Jornadas de Calidad de Software y Agilidad (UTN-UNaM-UNNE)

27 y 28 de noviembre de 2025

Facultad Regional Resistencia

Actas

Resumen

Las Jornadas de Calidad de Software y Agilidad (JCSA) inician en el año 2017, bajo el nombre original de Jornadas de Calidad de Software, con la intención de difundir avances significativos en el campo de la Ingeniería del Software y, en particular, conceptos, métodos y herramientas que contribuyan a la Calidad de Software. Brindando un foro común donde la industria del software y las universidades puedan intercambiar experiencias, exponer necesidades y fomentar acciones conjuntas en el área.

A partir del año 2021, las jornadas amplían su espectro, incorporando temas relacionados a la agilidad, e incluyendo la recepción de trabajos científicos y experiencias en la industria. En esta IX edición, las jornadas se vuelven a desarrollar en la Provincia del Chaco, puntualmente en las instalaciones de la Facultad Regional Resistencia de la Universidad Tecnológica Nacional. En ellas, tendrá lugar la exposición de artículos y experiencias emergentes de investigación, talleres, conferencias y experiencias de la industria con la participación de integrantes de la Universidad, Industria y Estado.

Las actividades de las jornadas se realizan de manera híbrida, combinando actividades en forma presencial en la ciudad de Resistencia con otras que se llevarán adelante en cada una de las sedes que participan en la organización. El evento está destinado a investigadores, profesionales, docentes y estudiantes en las áreas de Ingeniería y Licenciatura en Sistemas o carreras afines. Invitamos especialmente a profesionales y empresarios del sector del software, así como también público con interés en el tema.

Comité Organizador

FRRe – UTN

Dr. César J. Acuña
Dra. Verónica Bollati
Dra. Noelia Pinto



FCEQyN UNaM

Mgter. Alice Rambo
Dr. Eduardo Zamudio
Mgter. Sergio Caballero
Mgter. Martín Rey
Ing. Selva Ivaniszyn
Sra. Laura Burzminski



FaCENA – UNNE

Mg. Gladys Noemí Dapozo
Dr. Emanuel Irrazábal
Lic. Ma. de los Ángeles Ferraro
Dr. Rubén Bernal



Comité Académico

Mag. Liliana Cuenca Pletsch

(CINApTIC UTN FRRe)

Dra. Verónica Bollati

(CINApTIC UTN FRRe / CONICET)

Dr. César J. Acuña

(CINApTIC UTN FRRe)

Dra. Noelia Pinto

(CINApTIC UTN FRRe)

Esp. Gabriela Tomaselli

(CINApTIC UTN FRRe)

Ing. Nicolas Tortosa

(CINApTIC UTN FRRe)

Dra. Valeria Sandobal Verón

(GIESINUTN FRRe)

Ing. Germán Gaona

(CINApTIC UTN FRRe)

Dr. Horacio Leone

(INGAR UTN FRSF)

Dr. Silvio Gonnet

(INGAR UTN FRSF)

Dr. Gustavo Rossi

(LIFIA UNLP)

Dra. Alejandra Garrido

(LIFIA UNLP)

Dr. Andrés Rodríguez

(LIFIA UNLP)

Dr. Marcelo Estayno

(UNSAM)

Dr. Luis Olsina

(GIDIS UNLPam)

Dra. Luciana Ballejos

(INGAR UTN FRSF)

Mgter. Gladys Dapozo

(UNNE)

Dr. Emanuel Irrazabal

(UNNE)

Dr. Rubén Bernal

(UNNE)

Dr. David la Red Martínez

(UNNE)

Dra. Sonia Mariño

(UNNE)

Mgter. Mónica Tugnarelli

(UNER)

Dra. Gabriela Arévalo

(UNQui)

Dra. Luciana Roldan

(INGAR UTN FRSF)

Dra. Milagros Gutierrez

(UTN FRSF)

Dra. Mariel Alejandra Ale

(UTN FRSF)

Dra. Elsa Estevez

(UNSur/CONICET)

Dra. Alicia Mon

(UNLaM)

Dra. Nancy Ganz

(IIDII-FCEQyN-UNaM)

Mgter. Ing. Alice Rambo

(IIDII-FCEQyN-UNaM)

Ing. Selva Nieves Ivaniszyn

(FCEQyN-UNaM)

Mgter. Lic. Sergio Caballero

(FCEQyN-UNaM)

Mgter. Lic. Martín Rey

(IIDII-FCEQyN-UNaM)

Dr. Eduardo Zamudio

(IIDII-FCEQyN-UNaM)

Dr. Horacio Kuna

(IIDII-FCEQyN-UNaM)

Ing. Rubén Castaño

(FCEQyN-UNaM)

Dr. Diego Godoy

(UGD)

Ing. Edgardo Belloni

(UGD)

Mgter. Cristina Greiner

(UNNE)

Dra. María Fernanda Golobisky

(UTN-FRSF)

Ms. Ariel Passini

(UNLP)

Mgter. Pablo Thomas

(UNLP)

Dr. Fernando Emmanuel Frati

(UNDEC)

Dra. Marcela Genero Bocco

(UCM)

Dr. Jorge Andrés Díaz Pace

(SISTAN)

Dr. Nazareno Aguirre

(UNRC)

Temas de Interés

- Calidad del Producto y Proceso de Software
- Calidad Interna y Externa del Software
- Testing de Software
- Verificación y Validación
- Modelos y Métricas de Calidad de Software
- Calidad de Datos
- User Experience Diseño Centrado en el Usuario
- Herramientas, Aplicaciones Industriales y Estudios Empíricos
- Gestión de la Calidad de Software
- Enseñanza de Calidad de Software
- Métodos y Prácticas Ágiles
- Ingeniería del Software Continúa (Ejemplo: DevOps, Agile, etc)
- Ingeniería de Requisitos en Métodos Ágiles
- Peopleware y Management 3.0
- Estimación y Planificación Ágil
- Anomalías de Código – Refactoring

Índice

Modelos y Métricas para la Gestión de Calidad de Software: Adaptación Metodológica y Mejora Continua en el Contexto Organizacional	Pag.9
María de los Ángeles Ferraro; Sonia I. Mariño	
Desafíos técnicos y la aplicación de soluciones tecnológicas en el proceso de medición de granos de arroz	Pag.13
Mateo Beneyto; Mauricio Nicolás Castro; Mario Sergio Cleva	
SOFT. Modelo de proceso para la construcción y actualización de muestras de proyectos Software	Pag.23
Juan Andrés Carruthers; Emanuel Irrazabal	
Construyendo Alianza: una práctica ágil para fortalecer la retención estudiantil en Algoritmos y Estructuras de Datos	Pag.31
Noelia Pinto; Diego Bolatti; Nicolás Tortosa; María José Ojeda	
IRIR: Índice de Riesgo de Incendios Rurales Alternativo al Índice Meteorológico de Peligro de Incendios Forestales (FWI)	Pag.40
Carlos Brys	
NaVi-Q como soporte a la evaluación de visualizaciones narrativas: un estudio de caso.....	Pag.50
Andrea Lezcano Airaldi; Melisa Lezcano Airaldi; Emanuel Irrazábal	
MIIPP: Un enfoque datadriven para la mejora continua de la estimación de esfuerzo ágil	Pag.61
Nicolás Tortosa; César J. Acuña; Noelia Pinto; Gabriela Tomaselli; Yoel Marain; Florencia Gonzalez; Ezequiel Machuanin; Chiara Diez	
De la mejora continua a la formación por competencias: Evolución en la enseñanza de Ingeniería del Software (2015–2025)	Pag.72
Maria de los Angeles Ferraro; Laura Gómez Solís1	
Pasantías Profesionales y Calidad de Software en la Administración Pública: El Rol Interdisciplinario.....	Pag.82
María de los Angeles Ferraro; Sonia I. Mariño	
Uso de Inteligencia Artificial para la generación de código de calidad para optimizar el tiempo priorizando la lógica de solución en tesis de posgrado	Pag.91
Gerzel Stella Maris; La Rede Martínez David Luis; Agostini Federico	
De métodos prescriptivos a la cultura ágil basada en valores y principios.....	Pag.99
María José Ojeda; Verónica A. Bollati	
Enfoque de arquitectura orientada a agentes para la mejora de calidad de requerimientos ágiles.....	Pag.110
Noelia Pinto; César J. Acuña; Gabriela Tomaselli; Nicolás Tortosa; Denise Martinez; Jerónimo Zapata; Martín Lopez Soto; Ariano Miranda	



Trabajos



Modelos y Métricas para la Gestión de Calidad de Software: Adaptación Metodológica y Mejora Continua en el Contexto Organizacional

María de los Ángeles Ferraro

Contaduría General de la Provincia de Corrientes. Ministerio de Hacienda y Finanzas, Argentina.
ferrarom@cgpcorrientes.gob.ar

Sonia I. Mariño

Facultad de Ciencias Exactas y Naturales y Agrimensura Universidad Nacional del Nordeste, Argentina.
ferrarom@cgpcorrientes.gob.ar, msonia@exa.unne.edu.ar

Resumen

Este artículo presenta una experiencia real de la industria del software en el sector público de la provincia de Corrientes, detallando la adaptación metodológica de una Dirección de Sistemas que administra un sistema crítico para el funcionamiento contable-financiero de los tres poderes del estado provincial, implementado en la Contaduría General de la Provincia de Corrientes (CGPC), el Sistema Integrado de Información Financiera (S.I.I.F.). La experiencia abarca una trayectoria de diez años de mejora continua. Inicialmente, la Dirección de Sistemas enfrentaba un esquema de gestión informal, basado en la comunicación verbal y con ausencia de métricas de calidad, lo que dificulta la planificación del mantenimiento correctivo, preventivo, adaptativo y evolutivo de todo sistema y en particular de un sistema crítico.

Se describe un modelo de gestión flexible y contextualizado que se implementó para superar este desafío: el diseño de un modelo híbrido que tomó la estructura y control de ITIL para la gestión del ciclo de vida del software, integrándose con la flexibilidad y respuesta rápida de las metodologías Ágiles. Este enfoque, centrado en la adaptación al contexto de recursos acotados y alta responsabilidad, se consolidó mediante la implementación de ejes centrales de calidad: Trazabilidad de Requisitos, Control de Versiones y la Definición de Métricas para medir el progreso y el impacto de la gestión.

Los resultados demuestran la eficacia y sostenibilidad del modelo. A lo largo de una década, la mejora continua no solo optimizó el uso del tiempo y la disponibilidad del sistema, sino que logró institucionalizar la calidad como una política de gestión, validada a nivel directivo y reconocida en eventos federales y nacionales. Este estudio es una experiencia industrial relevante que valida la importancia de la adaptación metodológica y la persistencia en la gestión de calidad para sistemas críticos.

Palabras Claves: Calidad de Software, Modelos de Calidad, Métricas, ITIL, Metodologías Ágiles, Mejora Continua, Sector Público.

1. Contexto y Desafío: El Problema a Resolver

El objeto de estudio es el Sistema Integrado de Información Financiera (S.I.I.F.), un sistema crítico establecido por Ley de Administración Financiera y de los Sistemas de control, contrataciones y administración de bienes del sector público” N° 5.571/2004, por su alto riesgo era prioritario para considerar una iniciativa de calidad.

El punto de partida era una Dirección de Sistemas con:

1. Un esquema de gestión no formal para el desarrollo y mantenimiento del software, basado en la comunicación verbal.
2. Ausencia de indicadores y métricas formales que permitieran medir el progreso, la eficiencia y la calidad del desarrollo y el proceso de atención a las solicitudes de usuarios.

El desafío fundamental fue establecer un Modelo de Gestión de Ciclo de Vida del Software que permitiera la mejora continua y la reducción del riesgo operativo en un entorno de recursos humanos y tiempo acotados (en el tiempo, un equipo entre 12 y 15 integrantes trabajando en jornada matutina).

2 Solución Metodológica: El Modelo Híbrido y sus Pilares

La solución adoptada fue el diseño e implementación de un Modelo Híbrido y Flexible, reconociendo que “ningún modelo es universal” y que el éxito radica en la adaptación.

2.1 Adaptación Metodológica (ITIL/Ágil)

Se optó por una fusión metodológica que combina la estabilidad y el control de ITIL para el Gobierno de TI con la agilidad, visibilidad y flexibilidad en la ejecución (Scrum, Kanban):

Modelo Base (ITIL): Proporcionó la estructura y el control necesarios para la gestión del ciclo de vida del software, crucial para la estabilidad de un sistema administrativo-financiero regulado por ley.

Integración Ágil: Se incorporaron prácticas de metodologías Ágiles para la planificación, comunicación y respuesta rápida a las demandas y requerimientos de cambio, permitiendo reaccionar eficientemente a las necesidades operativas.

Adaptación Clave: Se enfatizó la toma de lo necesario de cada marco, sin caer en rigideces, para integrar la cultura organizacional, las necesidades propias de una Dirección de Sistemas y el contexto estatal.

2.2 Ejes Centrales de la Calidad (Modelos y Métricas)

La implementación se sostuvo en pilares concretos y medibles, que impactaron directamente en la calidad del producto y del proceso:

1. **Trazabilidad de Requisitos:** Establecida como el eje principal para garantizar que cada cambio se alinearía con la necesidades del usuario, asegurando la correspondencia del producto con las especificaciones.

2. **Control de versiones:** Formalización del proceso para manejar los cambios críticos del S.I.I.F., incluyendo políticas de despliegue y circuito de cambios.

3. **Definición y Verificación de Métricas:** Se establecieron indicadores para la toma de decisiones y la mejora. Estos midieron el impacto de la gestión en:

- Número de implementaciones exitosas (Disminución de vuelta atrás de implementaciones)
- Tiempo de respuesta a demandas.
- Eficiencia de la trazabilidad.
- Número de incidentes resueltos.

3 Resultados y Sostenibilidad a Largo Plazo (10 Años)

La experiencia de la mejora continua sostenida ha generado resultados concretos y validados a lo largo de diez años, transformando el enfoque de la calidad de una tarea técnica a una forma de trabajo.

3.1. Impacto y Eficiencia Comprobada

La aplicación consistente del modelo híbrido y las métricas se tradujo en una mejora consolidada y verificable:

- **Optimización de Recursos:** El equipo de Sistemas (promedio en el tiempo entre 12 y 15 integrantes) logró atender las demandas diarias y alcanzar los objetivos planificados en una jornada matutina. Esta optimización de recursos es una métrica concreta de eficiencia en el contexto público.
- **Métricas Validadas:** Los procesos de verificación y validación de las métricas propuestas en el Subsistema de Requerimientos arrojaron resultados muy satisfactorios en Noviembre de 2024, proporcionando un dato temporal y concreto de la medición de calidad.
- **Disponibilidad y Uso:** La continuidad y disponibilidad ininterrumpida del S.I.I.F. es una evidencia tangible del impacto positivo en el servicio, respaldado por la alta respuesta de los usuarios.

3.2. Institucionalización y Persistencia

La principal lectura de esta trayectoria de diez años es que la persistencia, el enfoque a generación de valor y la mejora continua en la gestión es la métrica clave. El modelo se ha consolidado y trascendido la Dirección de Sistemas:

Modelo Institucionalizado: El sistema de gestión se ha posicionado como una herramienta de gestión institucional, cuyos resultados se evalúan y se publican oficialmente en la Cuenta de Inversión del Ejercicio desde el año 2019.

Colaboración Académica: La colaboración con la Universidad ha sido un motor de innovación, generando lazos que permitan mejorar continuamente la formación de futuros profesionales y también las estrategias a implementar en la industria. Se concretó una presentación conjunta (CGPC/UNNE/UNLP) en las Jornadas Argentinas de Informática (JAIIO) 2025, en el simposio de Informática en el Estado.

Reconocimiento Externo: Los resultados de esta experiencia se compartieron y validaron en el 3er Encuentro de la Red Federal de Economía del Conocimiento ('Innovar para Transformar') en abril de 2025.

4 Conclusiones.

En conclusión, este caso demuestra que la adaptación al contexto organizacional de modelos (ITIL y Ágil) y el esfuerzo en una cultura de calidad son esenciales para el éxito a largo plazo en sistemas de alto impacto y responsabilidad, validando la calidad a través de la eficiencia operativa y el valor público, incluso en ausencia de certificación formal.

Referencias

1. CHEN, C., et al. ¿Cómo dañan los defectos las cualidades? un estudio empírico sobre la caracterización de una ontología de mantenibilidad de software en software de código abierto. En 2018 IEEE International Conference on Software Quality, Reliability and Security (QRS) . IEEE, 2018. 226-237.
2. PÉREZ, B., et al. Cuáles son las prácticas utilizadas por los profesionales de software en el pago de la deuda técnica: resultados de una familia internacional de encuestas. En Actas de la 3ra Conferencia Internacional sobre Deuda Técnica. 2020, 103-112.
3. LAZO, A. T.; VALENCIA, L., PELÁEZ, E. Ingeniería de Requisitos: de la especificación de requisitos de software al aseguramiento de la calidad. Cómo lo hacen las Mipymes desarrolladoras de software de la ciudad de Pereira. Entre Ciencia e Ingeniería, 2016, vol. 10, no 20, p. 117-123.
4. VILLAFAÑE, A., et al. Herramienta de gestión de trazabilidad de requerimientos en proyectos de software. En XVIII Congreso Argentino de Ciencias de la Computación. 2013.
5. FERRARO, M. de los A., et al. Especificación y trazabilidad de requerimientos en el desarrollo de aplicaciones web. 2012.
6. TABARES, M., ARANGO F., ANAYA, R. Una Revisión de Modelos y Semánticas Para la Trazabilidad de Requisitos. Revista EIA, ISSN 1794 -1237 Número 6, 33, <https://thedigitalprojectmanager.com/es/tools/herramientas-gestion-requisitos/>; Consultado al 29/04/2022
7. CRIADO, J.I.GIL-GARCÍA, J. R. Creación de valor público a través de tecnologías y estrategias inteligentes: desde servicios digitales hasta inteligencia artificial y más allá. Revista Internacional de Gestión del Sector Público, 2019.
8. Panorama de la gestión pública en América Latina y el Caribe: un gobierno abierto centrado en el ciudadano, 2018, p. 75.Cepal.org
9. AXELOS (2019). ITIL® Foundation. ITIL 4 Edition. Londres: The Stationery Office. <https://www.axelos.com>
10. Jefatura de Gabinete de Ministros, Elaboración de Estándares y Dictámenes Tecnológicos, <https://www.argentina.gob.ar/jefatura/innovacion-ciencia-y-tecnologia/onti/estandares-tecnologicos>, sitio consultado al 28-02-2025.
11. Contaduría General de la Provincia de Corrientes (C.G.P.C.). Publicación en YouTube institucional CGPC, Lista: Publicaciones y Congresos: (https://www.youtube.com/watch?v=wnjucudSJrKo&list=PLoDvrzdWwtCSBpiu_5aILifRSSaShvgBt&index=1).
12. <https://openaccess.uoc.edu/bitstream/10609/149948/1/ITIL4UnaIntroduccion.pdf>, consultado al 20/01/2025.

13. Información disponible en: www.cgpcorrientes.gob.ar

14. Videos disponibles en Youtube institucional https://www.youtube.com/channel/UCTueiog8oONwUT9kBMk_9Bg

15. Nuevo Régimen de Promoción de la Economía del Conocimiento, https://www.argentina.gob.ar/sites/default/files/2022/02/otg_informe_economia_del_conocimiento.pdf

Desafíos técnicos y la aplicación de soluciones tecnológicas en el proceso de medición de granos de arroz

Mateo Beneyto

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación (CInApTIC), Argentina.

Mauricio Nicolás Castro

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación (CInApTIC), Argentina.

Mario Sergio Cleva

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación (CInApTIC), Argentina.

Resumen

El control morfológico del arroz enfrenta el dilema entre precisión y costo, se presenta una solución de bajo costo para medir la morfología de granos de arroz combinando una cinta transportadora y una pipeline modular. Se estandariza la captura con fondo azul para maximizar contraste y reproducibilidad. Utiliza un fondo azul para optimizar la segmentación y la calibración se realiza con una moneda como patrón, corrigiendo la distorsión vertical y vinculando el proceso a SQuaRE: mejora fiabilidad (consistencia), usabilidad (procedimiento simple) y funcional suitability (exactitud). Para rendimiento, la arquitectura soporta CPU/OpenCL/CUDA con fallback automático. El seguimiento emplea Filtro de Kalman + Algoritmo Húngaro, manteniendo identidades ante oclusiones. Se reporta validación interna como indicador preliminar de satisfacción. La propuesta enfatiza calidad de producto/proceso, métricas y modelos SQuaRE, calidad de datos y métodos ágiles, con base sólida para transferencia y validaciones en campo.

Palabras clave: Morfología, Visión artificial, Otsu, Watershed, Filtro de Kalman, Algoritmo Húngaro, Fallback

1 Introducción

El mayor desafío presentado durante la elaboración de esta solución, incluso antes de indagar en los desafíos técnicos que se tuvieron que enfrentar, fue la propia determinación de la calidad morfológica de los granos de arroz. Pese a ser un proceso crítico en la industria agroalimentaria, presenta una actualidad donde los métodos empleados tienen limitaciones significativas. Por un lado, las técnicas manuales, si bien son extendidas, implican un considerable consumo de tiempo y esfuerzo. Además, dependen de la subjetividad y el conocimiento de personal experto, introduciendo una potencial variabilidad inter e intraoperador. Por otro lado, las soluciones automatizadas basadas en maquinaria especializada, aunque precisas, conllevan elevados costos de adquisición y mantenimiento, así como una notable complejidad de uso, lo que restringe su adopción en contextos productivos con recursos limitados. El desafío central, por tanto, es superar esta dicotomía mediante el desarrollo de un sistema que ofrezca la precisión de la automatización a un costo accesible, democratizando el acceso a tecnologías avanzadas de análisis.

Precisamente, el presente trabajo se inscribe en este marco, formando parte de un proyecto de investigación científica desarrollado en el Centro de Investigación Aplicada en Tecnologías de la Información y la Comunicación (CInApTIC) de la Universidad Tecnológica Nacional (UTN), Facultad Regional Resistencia, Argentina. Esta iniciativa busca generar alternativas tecnológicas de bajo costo para optimizar los procesos de control de calidad en el sector agrícola, contando con la colaboración del Instituto Nacional de Tecnología Agropecuaria (INTA) para la realización de los estudios empíricos. El enfoque del proyecto es la aplicación de técnicas de procesamiento digital de imágenes mediante el uso de software de libre distribución y hardware accesible para el modelado

morfológico, reconocimiento y clasificación de granos de arroz en la etapa de postcosecha.

Para abordar esta problemática y materializar el enfoque de bajo costo, se ha concebido un sistema integral que combina un dispositivo físico con una solución de software avanzada. El componente de hardware consiste en una cinta transportadora de escritorio, diseñada para disponer los granos de arroz de manera controlada, evitando su solapamiento y garantizando condiciones óptimas para la captura de imágenes mediante una cámara web o un dispositivo móvil. A fin de adaptarse al funcionamiento de esta cinta, se desarrolló una solución de software que constituye el núcleo del sistema: una aplicación multiplataforma implementada con una arquitectura multicapa que integra módulos de detección, seguimiento y medición.

2 Calidad de software y procesos orientados a atributos de calidad

El diseño empleado se apoya en el modelo de calidad de producto de la familia SQuaRE (ISO/IEC 25010), priorizando rendimiento, fiabilidad, mantenibilidad y portabilidad. La pipeline modular —calibración y estandarización de captura (fondo azul), segmentación (Otsu), separación (Watershed), seguimiento (Kalman + Húngaro) y medición— disminuye el acoplamiento y permite cambios localizados sin efectos colaterales, lo que refuerza la mantenibilidad en sus subcaracterísticas de modularity y modifiability previstas por el modelo.

Las condiciones de entrada estandarizadas y la calibración con objeto patrón mejoran la consistencia de los datos, fortaleciendo la funcional suitability del sistema. A su vez, la selección dinámica de backends (CPU/OpenCL/CUDA) con mecanismo de fallback equilibra latencia y continuidad en entornos heterogéneos, alineándose con performance efficiency (time behavior, resource utilization) y reliability (availability, fault tolerance) dentro del mismo modelo de calidad.

Finalmente, estas decisiones se describen y justifican conforme a ISO/IEC/IEEE 42010, explicitando stakeholders, concerns, views y decisiones de arquitectura, lo que asegura la trazabilidad entre necesidades y atributos de calidad definidos por SQuaRE.

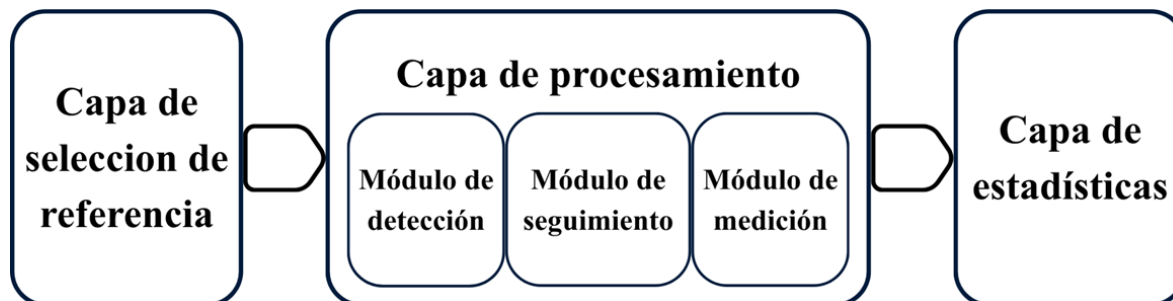


Fig. 1. Arquitectura de la aplicación

3 Toma de la imagen y calibración

Un desafío técnico fundamental en la implementación de sistemas de visión por computadora de bajo costo es la distorsión radial introducida por los lentes de cámaras web y dispositivos móviles. Esta distorsión provoca que los objetos cercanos a los bordes de la imagen parezcan distorsionados o curvados, lo que afecta significativamente la precisión de las mediciones morfológicas, especialmente en aplicaciones donde se requiere exactitud dimensional.

Dado que el diseño del sistema incluye una cinta transportadora que posiciona los granos de arroz en una trayectoria lineal y controlada, la corrección geométrica se puede simplificar al enfocarse únicamente en la distorsión vertical (eje Y). Al garantizar que todas las mediciones se realicen dentro de la franja central horizontal de la imagen, se elimina la necesidad de corregir la distorsión horizontal (eje X), optimizando considerablemente el proceso de calibración y reduciendo la complejidad computacional del sistema.

3.1 Método de Calibración Basado en Objeto de Referencia Estándar

Para abordar esta problemática, se ha desarrollado un método de calibración empírica que utiliza un objeto de referencia con dimensiones conocidas y estandarizadas: una moneda. La selección de este objeto responde a criterios de accesibilidad y estandarización, ya que en ausencia de instrumentos de medición precisos, sus dimensiones pueden ser consultadas en fuentes públicas confiables, garantizando un punto de partida reproducible para la calibración.

El procedimiento de calibración se basa en tomar múltiples mediciones del diámetro de la moneda a lo largo del eje vertical de la imagen. La medición de referencia se realiza colocando la moneda en la posición central de la imagen, donde la distorsión es mínima, y se mide su diámetro vertical en píxeles. El usuario debe ingresar la dimensión real del diámetro de la moneda, medida con un instrumento de precisión como calibre o regla, para establecer la relación inicial píxel por milímetro en el centro de la imagen.

Posteriormente, se toman mediciones de la moneda en posiciones extremas superior e inferior del eje vertical, donde la distorsión es más pronunciada. En estas zonas, el diámetro medido en píxeles será visualmente menor debido a la curvatura del lente, proporcionando información cuantitativa sobre la magnitud de la distorsión radial en diferentes posiciones verticales.

Para cada posición vertical muestreada, se calcula un Factor de Corrección. En los bordes, como la medida real en píxeles es menor, el Factor de Corrección será un valor mayor a 1. Este valor representa cuánto debe “estirarse” o escalar la medición en esa altura para ser correcta. Esta metodología asegura que la corrección de la distorsión vertical dependa explícitamente de la distancia al centro óptico de la imagen, mejorando la precisión a lo largo de toda la trayectoria de la cinta.

La precisión de esta corrección puede mejorarse significativamente aumentando el número de muestras de monedas y la cantidad de alturas de medición, lo que permitiría construir una curva de corrección polinomial más robusta para el eje vertical, adaptándose mejor al patrón de distorsión real del lente utilizado.

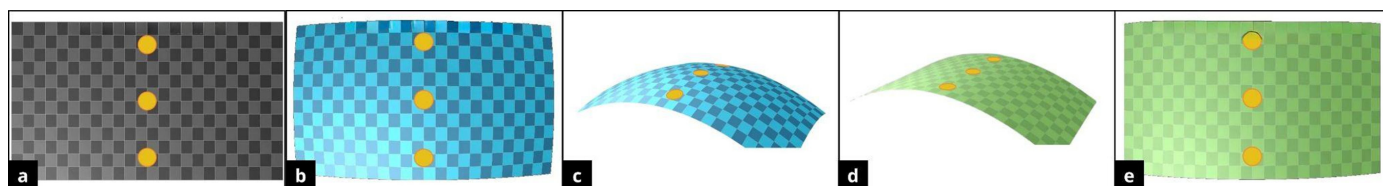


Fig. 2. Proceso de corrección de la distorsión radial del lente en el eje vertical. (a) Imagen real vista fuera de la cámara (b) Modelo que representa la curvatura del lente a partir del lente de la cámara, donde las monedas aparentan ser más pequeñas en los extremos. (c) Vista tridimensional de la superficie de distorsión. (d) Aplicación del factor de corrección a lo largo de la trayectoria vertical de las monedas. (e) Vista superior de la imagen corregida, demostrando que los objetos de referencia han recuperado una dimensión uniforme y consistente.

La calibración con objeto de referencia estandarizado contribuye directamente a la fiabilidad (consistencia de las mediciones en distintas posiciones del eje vertical) y a la usabilidad (procedimiento simple y repetible por personal técnico). En esta etapa del proyecto, y conforme al marco SQuaRE, definimos y aplicamos medidas cuantitativas de exactitud dimensional con referencia conocida, documentamos sistemáticamente los ajustes y factores de corrección para asegurar trazabilidad y repetibilidad del proceso de medición. Para etapas subsiguientes, se deja explicitado como trabajo futuro la incorporación de análisis estático de código y pruebas automatizadas sobre los módulos de medición, a fin de fortalecer mantenibilidad y robustez.

Dado que en esta instancia no hubo usuarios finales, se realizó una validación interna con usuarios técnicos del equipo y en contextos laborales reales controlados. Se registraron impresiones sobre claridad del procedimiento, tiempo de calibración y confianza en las mediciones. Este insumo se reporta como indicador preliminar de satisfacción del cliente interno, dejando explícito que la medición con usuarios finales se programará en fases posteriores de transferencia.

4 Pipeline de procesamiento de imágenes

La implementación de la solución de software requiere de una arquitectura de procesamiento de imágenes robusta y eficiente, capaz de manejar las particularidades inherentes al análisis morfológico de granos de arroz.

En este contexto, se ha desarrollado una pipeline de procesamiento digital de imágenes que constituye el núcleo computacional del sistema, diseñada específicamente para optimizar la detección, segmentación y medición de granos sobre fondos controlados, para nuestro contexto, se utilizó un fondo azul.

4.1 Elección de fondo azul

La elección de un fondo azul no es arbitraria: maximiza el contraste espectral con granos claros y estabiliza la segmentación. En trabajos previos se documenta, con representación visual del flujo de procesamiento, que el uso de fondo azul favorece la separación fondo-objeto y la puesta a punto del umbral de binarización en pipelines reproducibles (por ejemplo, separación por canales RGB, selección del canal y umbral) [4]. Complementariamente, en un estudio específico de comparación entre canales (R, G, B) se muestra que el canal azul ofrece el mejor comportamiento para distinguir clases y reducir solapamientos, mejorando la detectabilidad del canal rojo (que ofrece mayor contraste en un fondo azul con los granos) sobre los canales azul y verde [5]. En conjunto, estas evidencias sustentan el uso de fondo azul para simplificar la segmentación y disminuir errores de extracción de contornos en granos de arroz.

Desde una perspectiva técnica, el fondo azul ofrece ventajas adicionales en términos de estabilidad de iluminación. A diferencia de fondos blancos o grises, que pueden verse afectados por variaciones en la temperatura de color de las fuentes lumínicas, el azul mantiene características espectrales más consistentes bajo diferentes condiciones de iluminación artificial, reduciendo así la variabilidad en las mediciones y mejorando la reproducibilidad de los resultados.

4.2 Arquitectura de la pipeline

Aislamiento Cromático (Extracción de Canal): La primera acción consiste en extraer el canal rojo de la imagen capturada. Esta elección se fundamenta en el alto contraste natural entre el arroz (con componentes rojos, amarillentos o blancos) y el fondo azul, que posee una mínima contribución a este canal. Al aislar el canal rojo, se maximiza la diferencia de intensidad entre el grano y el fondo, lo cual es crucial para las fases posteriores.

Reducción de Ruido (Filtrado Gaussiano): Seguidamente, se aplica un filtrado gaussiano para suavizar la imagen. El objetivo es reducir el ruido de alta frecuencia y los artefactos de iluminación, preservando al mismo tiempo las formas morfológicas esenciales de los granos.

Segmentación Automática (Thresholding de Otsu): La tercera fase realiza una segmentación binaria utilizando el algoritmo de thresholding de Otsu. Este método es fundamental porque selecciona el umbral de manera automática basándose en el análisis del histograma de la imagen, maximizando la distinción entre el objeto y el fondo. Su principal beneficio es la adaptabilidad a las variaciones de iluminación y contraste, eliminando la necesidad de ajuste manual.

Refinamiento Morfológico (Operaciones de Apertura y Cierre): A continuación, se aplican operaciones morfológicas de apertura y cierre. La apertura está diseñada para eliminar el ruido de píxeles aislados y desconectar granos levemente solapados, mientras que el cierre busca rellenar pequeños huecos dentro de los objetos y conectar regiones cercanas que pertenecen al mismo grano. Estas operaciones son esenciales para obtener máscaras binarias que representen fielmente la morfología del arroz.

Separación de granos contiguos (Segmentación por Watershed): Pese a la optimización lograda mediante la segmentación cromática y el refinamiento morfológico, persiste un desafío crítico en la precisión de la medición: la separación inequívoca de granos contiguos o ligeramente superpuestos. Para mitigar este problema, se ha integrado este algoritmo: el centro de cada grano individual (obtenido por la Transformada de Distancia) se corresponde con un pico de alta intensidad, los cuales se usan como semillas para el algoritmo, dividiendo de forma correcta los granos de arroz que se encuentran en contacto.

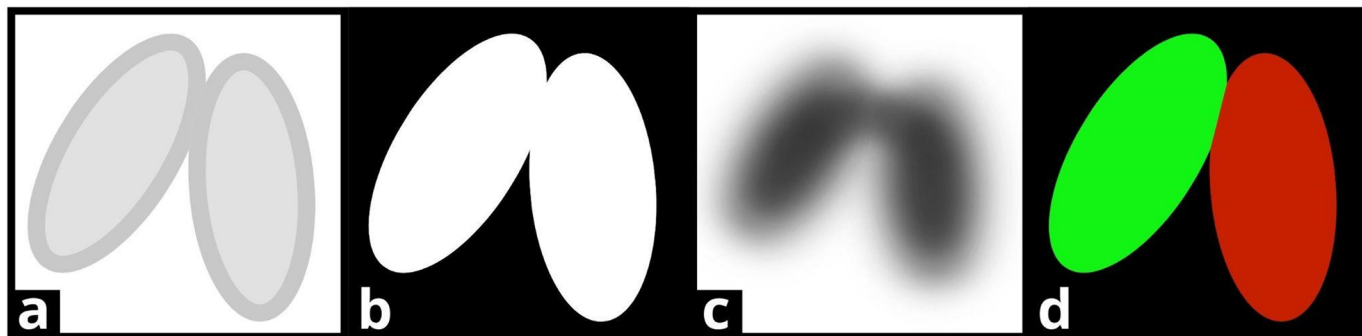


Fig. 3. Proceso de separación de granos de arroz contiguos mediante el algoritmo Watershed, donde (a) se presenta la imagen original tras el preprocesamiento cromático, (b) la máscara binaria obtenida en la segmentación inicial, (c) el mapa de distancia euclidiana que destaca los máximos locales correspondientes a los centros de cada grano y (d) el resultado final de la segmentación.

Detección de Contornos (Algoritmo de Canny): La sexta etapa utiliza el algoritmo de Canny para la detección de bordes. Este método emplea el análisis de gradientes para generar contornos precisos y continuos de los granos, que son la base para las mediciones morfológicas subsiguientes. La configuración de umbrales se optimiza para lograr un equilibrio ideal entre la detección de bordes y la minimización de detecciones erróneas.

Validación Visual (Generación de Overlay): Finalmente, se crea un overlay visual donde los contornos detectados se superponen a la imagen original. El uso de un color de alto contraste maximiza la visibilidad, lo que permite una validación cualitativa inmediata y facilita la identificación de cualquier posible anomalía en el procesamiento.

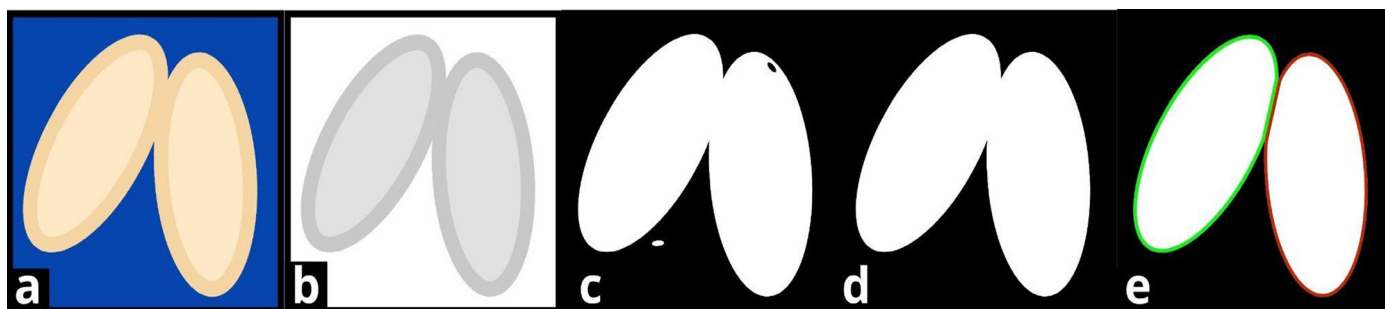


Fig. 4. Etapas de la pipeline: (a) la imagen original, (b) imagen después del aislamiento cromático, donde se extrae el canal rojo para mejorar el contraste entre los granos y el fondo; (c) la máscara binaria obtenida mediante segmentación automática con thresholding de Otsu; (d) la máscara refinada con operaciones morfológicas y (e) la validación visual, que utiliza un overlay para comprobar la precisión de los contornos detectados.

5 Sistema de seguimiento

Una vez que los granos son detectados en cada fotograma capturado, el sistema enfrenta un desafío fundamental: mantener la identidad única de cada grano a medida que se desplaza por la cinta transportadora. Sin un mecanismo de seguimiento robusto, el sistema trataría cada fotograma de manera aislada, lo que resultaría en conteos erróneos, mediciones duplicadas y la imposibilidad de realizar un análisis individualizado de calidad. Para superar este obstáculo, se ha implementado una arquitectura de seguimiento que combina tres componentes clave: el Filtro de Kalman para la predicción del movimiento, el Algoritmo Húngaro para la asignación óptima de detecciones, y un sistema de gestión del ciclo de vida de tracks.

5.1 Modelo de Estado y Predicción con Filtro de Kalman

El Filtro de Kalman es un estimador recursivo óptimo en el sentido de mínimo error cuadrático medio, diseñado para inferir el estado interno de un sistema dinámico lineal a partir de una serie de mediciones incompletas o ruidosas. En el contexto del seguimiento de granos de arroz, su función es estimar de manera robusta la

trayectoria de cada grano, modelando su estado cinemático —posición y velocidad— a partir de las detecciones visuales obtenidas en cada fotograma, las cuales son consideradas observaciones inherentemente ruidosas. El filtro opera como un predictor y corrector estocástico, manteniendo una representación probabilística del estado de cada grano y actualizándose secuencialmente a medida que se adquieren nuevas observaciones.

Principio Fundamental y Formulación Bayesiana. En su núcleo, el Filtro de Kalman es una implementación de inferencia bayesiana. El algoritmo mantiene una “creencia” (una distribución de probabilidad) sobre el estado actual del sistema. Esta creencia se modela asumiendo que toda la incertidumbre, tanto en la dinámica del sistema como en el proceso de medición, puede ser descrita mediante distribuciones Gaussianas. El proceso operativo se descompone en un ciclo de dos fases:

Predicción (a priori): Utilizando el modelo dinámico del sistema, el filtro predice el estado y la incertidumbre asociada para el siguiente instante de tiempo.

Corrección (a posteriori): Cuando se recibe una nueva medición, esta se utiliza para corregir la predicción, resultando en una nueva estimación del estado con una incertidumbre reducida.

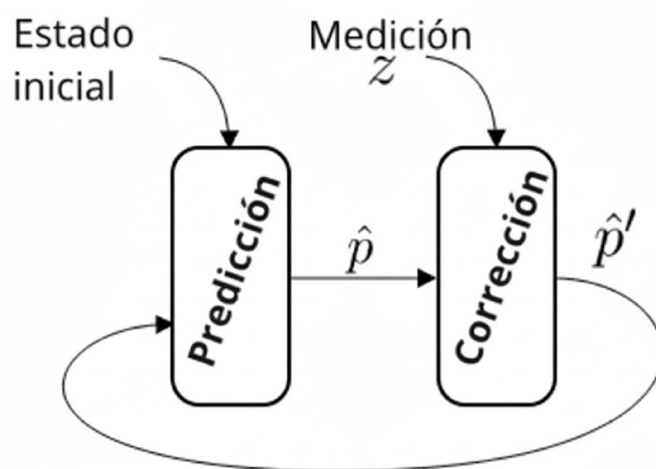


Fig. 5. Esquema del ciclo de funcionamiento del Filtro de Kalman. El proceso alterna entre una fase de predicción del estado, basada en el modelo dinámico, y una fase de corrección, donde se ajusta la estimación según las mediciones observadas. Este ciclo iterativo permite mantener una estimación precisa y actualizada del sistema

5.2 Implementación Algorítmica en el Flujo de Procesamiento

La implementación del Filtro de Kalman se ejecuta como un proceso cíclico sincronizado con la adquisición de fotogramas. Antes de comenzar el seguimiento, el sistema se inicializa configurando los parámetros que definen el comportamiento esperado de los objetos. Esto incluye establecer un modelo de movimiento (cómo se espera que los granos se desplacen) y definir numéricamente los niveles de incertidumbre: el ruido del proceso, que representa la impredecibilidad del movimiento, y el ruido de la medición, que cuantifica la imprecisión de las detecciones de la cámara. Estos valores de ruido son calibrados para ajustar la respuesta del filtro, equilibrando la confianza entre la predicción interna del modelo y la evidencia de las mediciones reales.

Una vez inicializado, el ciclo operacional comienza con la captura de un fotograma y la posterior detección de objetos, lo que genera una lista de mediciones con las posiciones de los granos. Para cada objeto ya bajo seguimiento, el filtro primero ejecuta una fase de predicción, utilizando el estado del fotograma anterior para estimar su posición actual. A continuación, un mecanismo de asociación de datos, basado en el Algoritmo Húngaro, establece la correspondencia más probable entre las posiciones predichas y las mediciones reales, resolviendo el problema de asignación.

Con las asociaciones establecidas, se ejecuta la fase de corrección. En este paso, se calcula un factor de ponderación óptimo (la Ganancia de Kalman) que equilibra la incertidumbre de la predicción frente a la de la medición. El filtro utiliza este factor para fusionar ambas fuentes de información, produciendo una estimación de estado final que es más precisa y robusta que cualquiera de las dos por separado. Este ciclo se complementa con una lógica de gestión de trazas que crea nuevos seguidores para detecciones no asociadas y elimina aquellos que se

pierden de vista, asegurando la eficiencia del sistema. La estimación corregida de cada fotograma sirve como base para la predicción en el siguiente, perpetuando el ciclo recursivo.

5.3 Algoritmo Húngaro para Asignación Óptima de Detecciones

Una vez que el Filtro de Kalman ha generado las predicciones de estado, el sistema debe resolver el problema de asociación de datos: emparejar de manera inequívoca cada predicción con su correspondiente detección en el nuevo fotograma. Este es un problema de optimización combinatoria, donde el objetivo es encontrar la asignación uno a uno entre el conjunto de predicciones y el de detecciones que minimice un costo total acumulado. Para ello, se emplea el Algoritmo Húngaro, un método que garantiza encontrar la solución globalmente óptima.

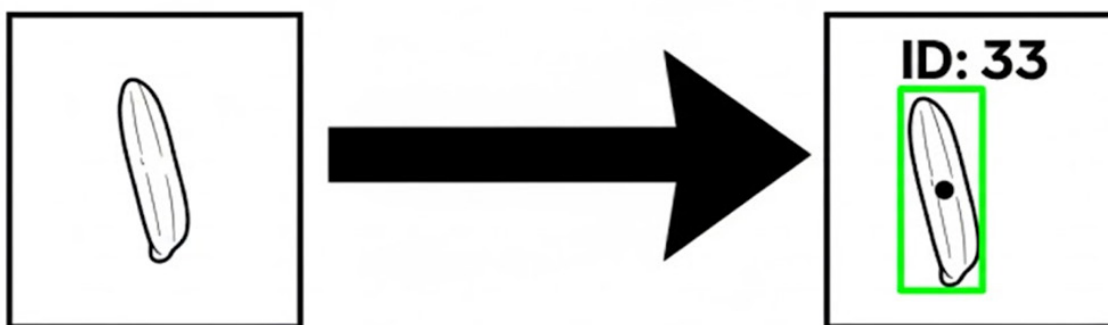


Fig. 6. Resultado del proceso de detección y seguimiento. Un objeto (izquierda) es exitosamente identificado en un fotograma posterior (derecha). El sistema lo enmarca en un rectángulo delimitador (*bounding box*) y le asigna un ID único, confirmando que la asociación entre la predicción y la nueva detección fue resuelta de manera óptima.

5.4 Medición de Compatibilidad

Para que el algoritmo pueda tomar una decisión, primero se debe cuantificar qué tan “compatible” es cada posible par de predicción-detección. Esto se logra mediante una función de costo híbrida que combina dos métricas complementarias. La primera es la distancia euclidiana entre la posición predicha y la detectada, que evalúa la continuidad cinemática. La segunda es el Intersection over Union (IoU), que mide la superposición espacial de sus cajas delimitadoras, capturando así la consistencia morfológica. Estas dos métricas se ponderan y combinan en un único valor de costo, donde un costo menor indica una mayor compatibilidad.

5.5 Optimización del Proceso

Antes de ejecutar el algoritmo principal, se aplica un sistema de gating dual para reducir el espacio de búsqueda. Este mecanismo actúa como un filtro de viabilidad, descartando de antemano las asignaciones que son físicamente inverosímiles. Un par solo se considera candidato si la distancia entre ellos es inferior a un umbral o si su superposición (IoU) supera un mínimo requerido. Esto previene que el algoritmo invierta tiempo en evaluar emparejamientos claramente incorrectos.

Con la matriz de costos ya calculada y filtrada, se ejecuta el algoritmo de Kuhn-Munkres. Este método iterativo manipula la matriz de costos para encontrar un conjunto de asignaciones donde el costo total sea el mínimo posible, garantizando así la solución más coherente a nivel global en lugar de simplemente emparejar los más cercanos de forma individual.

5.6 Integración y Resultados

El resultado del algoritmo clasifica cada predicción y detección en una de tres categorías.

Asignados: Las predicciones emparejadas con detecciones se utilizan para ejecutar la fase de corrección del Filtro de Kalman, actualizando y refinando la trayectoria del objeto.

No Asignados (Predicciones): Las predicciones que no encuentran pareja se consideran temporalmente ocluidas o perdidas. El sistema mantiene su seguimiento basándose únicamente en el modelo de movimiento del filtro.

No Asignados (Detecciones): Las detecciones que no corresponden a ninguna predicción existente se identifican

como nuevos objetos que ingresan al campo de visión, lo que desencadena la creación de nuevos trackers para ellos.

5.7 Manejo de Oclusiones y Reparición de Granos

El sistema de seguimiento está diseñado para manejar robustamente las oclusiones, un desafío común en entornos industriales donde los granos pueden ocultarse temporalmente. Cuando un grano desaparece de la vista, su traza no se elimina de inmediato. En cambio, el Filtro de Kalman continúa operando en modo de predicción, estimando la posición probable del grano basándose en su última velocidad y dirección conocidas. Esta capacidad de “imaginar” la trayectoria del objeto durante un número predefinido de fotogramas es crucial, ya que permite que el sistema anticipe una posible reaparición sin perder la identidad del grano.

Si el grano vuelve a ser visible, el algoritmo Húngaro entra en juego para gestionar la reaparición. Compara la nueva detección con todas las predicciones activas, incluidas las de los granos que estaban temporalmente perdidos. Si la nueva detección se encuentra lo suficientemente cerca de la posición predicha de un grano oculto, el sistema reasocia la detección con su traza original, restaurando su identidad de forma transparente. Esta reasociación inteligente asegura que cada grano sea seguido de manera continua a lo largo de su recorrido, garantizando la integridad de las mediciones y los conteos incluso cuando se producen amontonamientos o sombreados momentáneos.

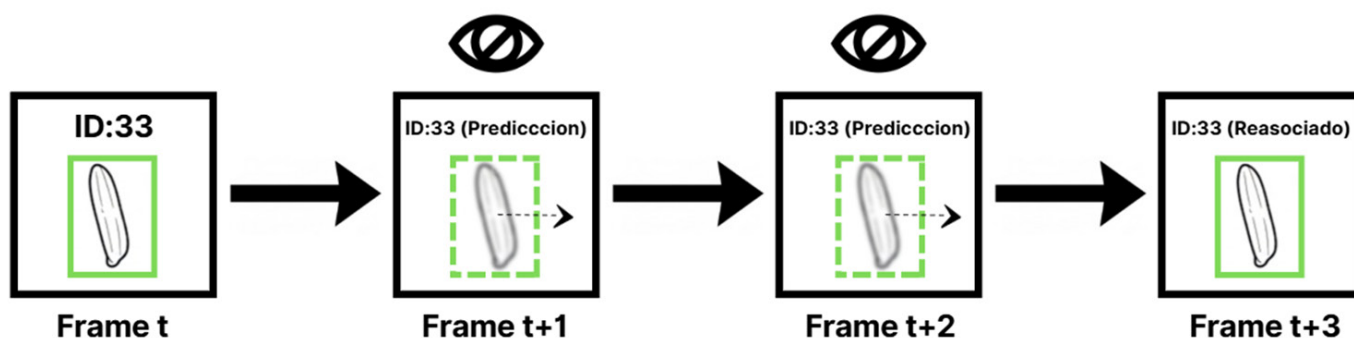


Fig. 7. Esquema del proceso de seguimiento robusto ante una oclusión, mostrando cómo la identidad de un objeto (ID: 33) se preserva a través de la predicción de su trayectoria mientras está oculto (t+1, t+2) hasta su exitosa reasociación (t+3).

6 Optimización de rendimiento

El análisis industrial de granos de arroz en tiempo real impone una carga computacional extremadamente alta. El sistema debe procesar un flujo de video de alta frecuencia donde cada fotograma individual se somete a una secuencia de operaciones complejas, desde el preprocesamiento hasta la detección y el seguimiento. Dado que el objetivo principal del proyecto es ofrecer una solución accesible que funcione en hardware de bajo costo y dispositivos con poder computacional limitado (como computadoras antiguas o móviles), un enfoque que dependa exclusivamente de una CPU estándar corre el riesgo de convertirse en un cuello de botella crítico, incapaz de mantener el ritmo del flujo de datos. Para superar esta limitación fundamental, se optó por una solución que no sólo acelerará el procesamiento, sino que también garantizara la flexibilidad y la fiabilidad. La estrategia elegida fue la implementación de una arquitectura de backends múltiples, un enfoque híbrido que permite al sistema aprovechar la potencia de procesamiento paralelo de las GPUs cuando estén disponibles, manteniendo al mismo tiempo una alta eficiencia en CPUs estándar.

6.1 Optimización de Rendimiento y Arquitectura de Backends

Para cumplir con estos exigentes requisitos, el sistema se ha diseñado sobre una arquitectura de backends múltiples implementada en C++ con la librería OpenCV. Esta elección tecnológica es fundamental, ya que C++ proporciona el control de bajo nivel sobre la gestión de memoria y los recursos del sistema, algo crítico para exprimir el máximo rendimiento en hardware limitado. OpenCV, por su parte, ofrece un conjunto de algoritmos de visión por computadora altamente optimizados, con soporte nativo para múltiples motores de aceleración. La

arquitectura soporta el procesamiento estándar a través de la CPU, así como la aceleración por hardware mediante las tecnologías de GPU OpenCL y CUDA. En el momento de la inicialización, el sistema realiza una comprobación automática para detectar qué recursos están presentes y son funcionales, seleccionando dinámicamente el backend más performante para la tarea.

6.2 Sistema de Conmutación por Error para Máxima Fiabilidad

Un pilar fundamental de la arquitectura es su sistema de conmutación por error (fallback) robusto. La fiabilidad en un entorno industrial o de investigación es a menudo más crítica que la velocidad máxima. Por ello, si el backend de aceleración seleccionado (ya sea CUDA u OpenCL) falla durante el inicio o la ejecución, el sistema está programado para conmutar automáticamente al backend de CPU. Este mecanismo de seguridad, implementado mediante el manejo de excepciones en C++, garantiza la continuidad operativa ininterrumpida. Esto permite que el análisis de granos continúe incluso en caso de un fallo en el hardware de la GPU o problemas con sus controladores, asegurando que la aplicación nunca deje de funcionar por completo.

6.3 Optimizaciones Específicas y Gestión de Memoria

El rendimiento no solo se logra eligiendo el hardware correcto, sino utilizándolo de la manera más eficiente posible. Cada backend cuenta con implementaciones optimizadas para las operaciones de visión por computadora más críticas de la pipeline, como el filtrado de imágenes y la detección de bordes. Estas rutinas están diseñadas para explotar las fortalezas de cada arquitectura, ya sea el paralelismo masivo de los núcleos CUDA o las capacidades de vectorización de las CPU modernas.

Además, se ha implementado una estrategia de gestión de memoria avanzada para minimizar la latencia asociada a las transferencias de datos entre la CPU y la GPU, un conocido cuello de botella. El sistema utiliza tipos de datos específicos de OpenCV para priorizar mantener los datos de imagen en la memoria de la GPU durante la mayor cantidad de pasos del procesamiento posible. Esto reduce la necesidad de copiar datos de un lado a otro, agilizando significativamente el flujo de trabajo. Adicionalmente, elementos computacionalmente costosos como los kernels de filtrado se almacenan en caché, evitando el gasto de tiempo que supondría su recreación en cada fotograma.

7 Conclusiones

El desarrollo de este sistema integral representa la materialización exitosa de un enfoque científico y tecnológico de bajo costo para la medición morfológica de granos de arroz, superando la dicotomía tradicional entre precisión y accesibilidad económica en el sector agroalimentario. La iniciativa ha logrado democratizar el acceso a tecnologías de análisis avanzado mediante la combinación de hardware accesible y software especializado de código abierto.

El sistema desarrollado presenta una arquitectura modular para el análisis de granos de arroz, integrando procesamiento digital de imágenes con una segmentación optimizada mediante el uso estratégico de un fondo azul y una pipeline eficiente. Se implementa una calibración geométrica empírica con objetos estándar, simplificada a una corrección unidimensional en el eje vertical para adaptarse a la cinta transportadora. El seguimiento robusto de granos se logra a través de la combinación del Filtro de Kalman y el Algoritmo Húngaro, garantizando la identidad única y la precisión de las mediciones en condiciones de oclusión. Finalmente, la flexibilidad de hardware se asegura mediante una arquitectura de backends múltiples (CPU, OpenCL, CUDA) que incluye un sistema de conmutación por error (failover) para la fiabilidad operativa industrial.

Finalmente, desde la perspectiva de la agilidad, el trabajo se gestionó mediante historias de usuario y ceremonias regulares: daily para sincronizar avances y bloqueos, Sprint Review para inspeccionar el incremento y recoger feedback, y Retrospectiva para impulsar mejoras continuas del proceso. Mantuvimos una colaboración presencial sostenida, con comunicación frecuente de progresos y dificultades, lo que facilitó una adquisición osmótica de conocimiento dentro del equipo y aceleró la resolución de problemas sin sacrificar calidad.

Este trabajo ha logrado materializar una solución tecnológica innovadora que combina precisión científica con accesibilidad económica. La metodología desarrollada, los algoritmos implementados y la arquitectura propuesta constituyen una base sólida para futuras investigaciones y aplicaciones en el campo de la visión por computadora aplicada a la agricultura, contribuyendo al avance del conocimiento científico y al desarrollo tecnológico del país.

Referencias

1. Birla, R., Chauhan, A.P.S.: An efficient method for quality analysis of rice using machine vision system. *Journal of Advances in Information Technology* (2015).
2. Canny, J.: A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1986).
3. Castro, M.N., Cleva, M.S.: Herramienta móvil de bajo costo para la caracterización morfológica de granos de arroz utilizando procesamiento de imágenes digitales. In: *II Congreso Paraguayo de Informática, Asunción, Paraguay* (2024).
4. Castro, M.N., Beneyto, M., Cleva, M.S.: Reconocimiento morfológico de granos de arroz mediante procesamiento de imágenes con tecnologías accesibles. In: *Simposio Latinoamericano sobre Educación, Tecnología y Ciencia* (2025).
5. Cleva, M.S., Liska, D.O., et al.: Validación del método de la elipse que mejor ajusta para la determinación de las dimensiones de granos y semillas con ImageJ (CLICAP 2022) (2022a)
6. Cleva, M.S., Fontana, M.L., et al.: Reconocimiento de granos vanos/llenos de arroz mediante Procesamiento Digital de Imágenes con ImageJ (2022b).
7. Cleva, M.S., et al.: Técnicas de procesamiento digital de imágenes para la estimación del peso de granos de arroz cáscara. In: *XXVIII Reunión de Comunicaciones Científicas, Técnicas y de Extensión, UTN Facultad Regional Resistencia* (2023).
8. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*, 3rd edn. Prentice Hall, Upper Saddle River (2008).
9. ISO/IEC/IEEE 42010: Software, systems and enterprise — Architecture description (2022).
10. ISO/IEC 25000: Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE (2014).
11. ISO/IEC 25010: Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models (2011).
12. ISO/IEC 25040: Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Evaluation process (2024).
13. Kalman, R.E.: A new approach to linear filtering and prediction problems. *ASME Journal of Basic Engineering* (1960)
14. Kuhn, H.W.: The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* (1955)
15. Otsu, N.: A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics* (1979).
16. Sahbani, B., Adiprawita, W.: Kalman filter and Iterative-Hungarian Algorithm implementation for low complexity point tracking as part of fast multiple object tracking system. In: *International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)* (2016)
17. Vincent, L., Soille, P.: Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1991)

SOFT. Modelo de proceso para la construcción y actualización de muestras de proyectos Software

Juan Andrés Carruthers

Grupo de Investigación en Calidad de Software, Facultad de Ciencias Exactas y Naturales y Agrimensura, Universidad Nacional del Nordeste, Argentina.
jacarruthers@exa.unne.edu.ar

Emanuel Irrazabal

Grupo de Investigación en Calidad de Software, Facultad de Ciencias Exactas y Naturales y Agrimensura, Universidad Nacional del Nordeste, Argentina.
eirrazabal@exa.unne.edu.ar

Resumen

Las muestras de proyectos software son una de las principales fuentes de información utilizadas en la Ingeniería del Software Empírica. El desarrollo de software de fuente abierta y su difusión a través de plataformas públicas de código han facilitado el acceso a millones de proyectos, sin embargo, la falta de controles en la creación de proyectos implica que algunos pueden contener información irrelevante. Otras fuentes son las muestras compartidas por la comunidad científica, aunque, en ocasiones, se conforman siguiendo procedimientos arbitrarios, con escasa documentación y cuestionables desde el punto de vista metodológico. En este sentido, se propuso la implementación del modelo de proceso SUM4SOFT para la construcción, actualización y curaduría de colecciones de proyectos software. El modelo se probó en un estudio de Ingeniería del Software y se refinó por medio de un estudio de caso. La aplicación del modelo de proceso resultó en un instrumento de recolección de datos y colecciones de proyectos software que se validaron en un estudio de minería de repositorios. Esto demostró que tanto SUM4SOFT como las herramientas implementadas constituyen soluciones viables para generar resultados confiables en la Ingeniería del Software Empírica.

Palabras Clave: Muestra de proyectos de software, Actualización de muestras, Software de fuente abierta, Ingeniería del Software Empírica.

1 Introducción

La Ingeniería del Software Empírica (ISE) es una disciplina enfocada en recolectar y analizar datos obtenidos del proceso de desarrollo y mantenimiento del software para evaluar herramientas, prácticas o procedimientos que permitan, por ejemplo: corregir errores, detectar vulnerabilidades, estimar el tiempo de desarrollo, con el objetivo de integrar los hallazgos como evidencia científica [1].

Una de las principales fuentes de datos empleadas para conducir estudios empíricos es el código fuente del software [2]. En este sentido, plataformas públicas para compartir código de fuente abierta como GitHub, Gitlab o Bitbucket ofrecen acceso a millones de proyectos software para el desarrollo de estudios empíricos, los cuales deben ser seleccionados de manera cuidadosa porque muchos de ellos no son aptos para la investigación en ISE [3, 4]. Dado el esfuerzo que significa la recolección de datos, algunos estudios comparten a la comunidad científica las colecciones de proyectos software utilizadas en la investigación.

No obstante, estudios secundarios [5, 6] revelaron ciertas prácticas encontradas en la literatura que comprometen la validez de los resultados obtenidos a partir de estas colecciones. Por un lado, problemas metodológicos en la conformación de las muestras: número reducido de proyectos, sin criterios explícitos de evaluación de calidad y técnicas de muestreo no probabilísticas. Por otro lado, cuestiones sobre la difusión de los instrumentos de la investigación, como instrucciones incompletas, acceso limitado a los conjuntos de datos, ausencia de paquetes de replicación, entre otras.

Para abordar esta necesidad se propuso la construcción del modelo de proceso SUM4SOFT. El modelo detalla los pasos y reglas para construir colecciones, y además ofrece plantillas para estandarizar la generación de los

instrumentos de recolección de datos facilitando la replicación de los estudios.

Las secciones de este trabajo se organizan de la siguiente manera. En la sección 2 se describen los fundamentos conceptuales y estudios previos que motivan esta investigación. En la sección 3 se expone el enfoque metodológico adoptado para la construcción y validación del modelo. En la sección 4 se detalla la propuesta SUM4SOFT, sus actividades, roles y artefactos. En la sección 5 resume los estudios empíricos realizados para probar el modelo y el instrumento JavaQ. Finalmente, En la sección 6 se presentan los aportes principales, las implicancias del trabajo y las futuras líneas de investigación.

2 Antecedentes

Para conocer el estado de arte de la construcción de colecciones orientadas a estudios en ISE se condujeron dos mapeos sistemáticos [7]. Inicialmente, se analizaron estudios primarios que emplearon colecciones de proyectos software publicados en revistas y congresos especializados en ISE [8]. Durante la revisión, se detectó que el 28% de los artículos emplearon colecciones de referencia. Por ello, se decidió ampliar la revisión en un segundo mapeo sistemático enfocado en estas colecciones consumidas de manera frecuente por la comunidad científica para complementar los hallazgos iniciales [9]. En ambos casos se estudiaron los criterios de selección, la caracterización de los proyectos software, los metadatos extraídos y el uso de herramientas.

En la mayoría de los estudios relevados se incluyeron criterios de selección específicos con el caso de estudio, demostrando la falta de estrategias de selección estandarizadas. Casi en su totalidad no consideraron utilizar la información más reciente para conducir los estudios, tanto es así que la mayoría nunca actualizaron las muestras (ver Fig. 1). Adicionalmente, se recopilieron los siguientes criterios para identificar proyectos aptos para la investigación en ISE: la disponibilidad del código fuente, la utilización de herramientas para dar soporte a procesos, el tamaño, la historia de desarrollo, la popularidad, la colaboración del equipo de desarrollo y la actividad reciente. Con respecto a los metadatos, se extrajeron métricas de código, pruebas unitarias, datos de repositorio, entre otras.

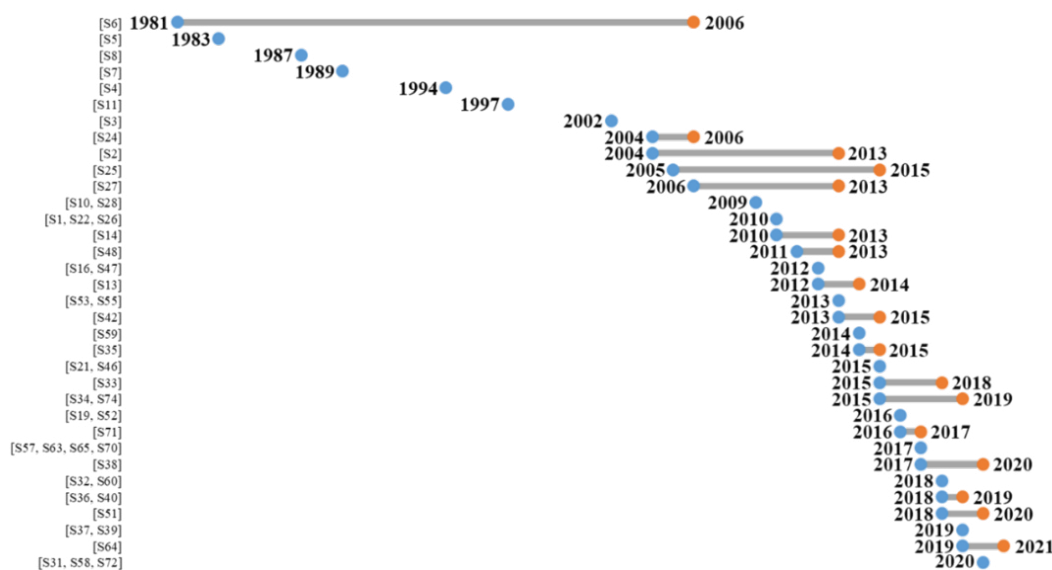


Fig. 1. Fecha de creación y última actualización de las colecciones relevadas. Fuente [9].

Al igual que otros estudios secundarios se evidenció la dificultad de acceso a los datos, reporte de herramientas y paquetes de replicación limitado, desestimación de la vigencia de las muestras y falta de estrategias para actualizar los datos. Por otra parte, la Fig. 1 refleja como la mayoría de las colecciones de referencia nunca han sido actualizadas, con últimas versiones publicadas desde 1983 hasta 2016. Esto podría presentar inconvenientes debido al proceso dinámico de evolución continua del software [10]. Además de la implementación de nuevas funcionalidades, corrección de errores, comprobación de funcionalidades, documentación, etc.; las tecnologías, entornos y bibliotecas utilizados para construir, probar y desplegar las soluciones informáticas también evolucionan con el tiempo, provocando a su vez la actualización del propio sistema para mantenerse operativo.

La vigencia refiere a la medida en que una muestra representativa recopilada en el pasado continúa representando a la misma población en el presente, y cobra notoriedad en aquellos dominios donde los datos sufren

cambios frecuentes [11]. Por ello, es necesario monitorear los datos recopilados en la colección de proyectos software para asegurar que siga representando a la población objetivo.

Para investigar la relevancia de la vigencia en colecciones de la ISE se analizó la evolución de una población de proyectos software durante un período de seis años (desde julio del 2017 hasta junio del 2023) a través de un estudio longitudinal [12]. El estudio abarcó siete métricas de repositorio: número de confirmaciones, colaboradores, PR cerradas, PR combinadas, incidentes resueltos, estrellas y bifurcaciones.

El análisis estadístico realizado con la prueba de KruskalWallis H [13] reveló diferencias significativas en todas las métricas en el período comprendido. En la Fig. 2 se graficaron las curvas de supervivencia de las métricas mostrando que, en la mayoría de ellas, la probabilidad de que una muestra con estos datos continúe siendo representativa por seis años o más es de 0%. Aunque la evolución no se manifiesta de manera uniforme en todas ellas, emplear conjuntos de datos creados hace varios años no parece ser una estrategia viable si el objetivo es producir resultados generalizables.

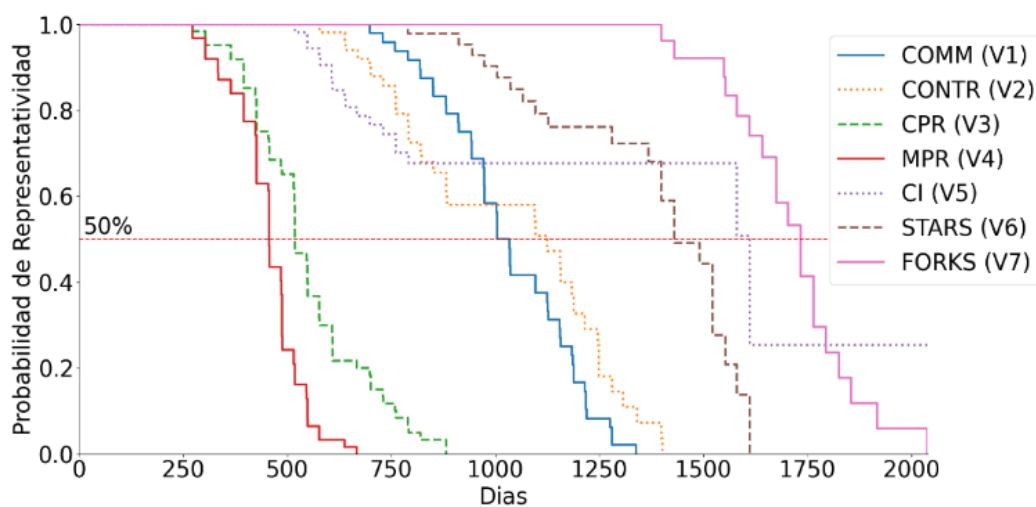


Fig. 2. Análisis de supervivencia de la representatividad por métrica. Fuente [12]

3 Metodología

Para diseñar, construir y validar SUM4SOFT se empleó la metodología ciencia de diseño [14] compuesta de cinco actividades: explicar el problema, definir objetivo y requisitos, diseñar y desarrollar el artefacto, demostrar el artefacto y evaluar el artefacto. La Fig. 3 muestra las tareas planteadas para alcanzar el objetivo de la investigación, destacando cinco estudios: dos mapeos sistemáticos de la literatura, un estudio longitudinal, un estudio de minería de repositorios de software y un estudio de caso.

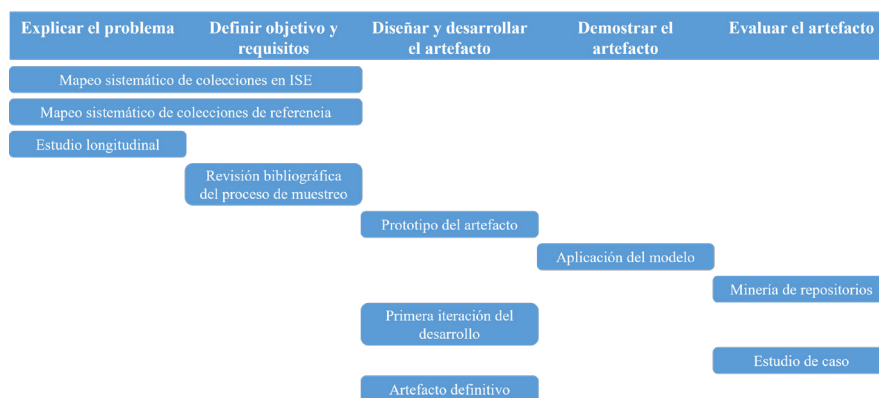


Fig. 3. Cronología de ejecución de las tareas de ciencia de diseño

4 Desarrollo

Según Marbán et al. [15] un modelo de proceso es un conjunto de pasos o actividades para llegar a un objetivo determinado. Estas actividades reciben las entradas necesarias para su ejecución y producen salidas. Por lo tanto, es necesario detallar cuáles son las actividades involucradas en los procesos modelados, lo que implica la especificación de sus entradas, salidas, orden de ejecución, y actores, entre otras cuestiones. Para la definición de las actividades se estudiaron las guías de muestreo en Ingeniería del Software relevadas por Baltes y Ralph [5]. SUM4SOFT (por sus siglas en inglés, Sampling and Update Model for Software datasets) es el modelo de proceso propuesto para abordar las problemáticas reportadas de los estudios en la ISE. El diseño del modelo abarca las cuatro perspectivas propuestas por Curtis et al. [16] comportamiento, funcional, organizacional e informacional; y se diagramaron con el lenguaje de modelado SPEM 2.0. En la Fig. 4 se muestra la perspectiva de comportamiento del modelo completo donde se distinguen las dos actividades centrales: construir el instrumento de recolección de datos (A1) y ejecutar el instrumento de recolección de datos (A2).

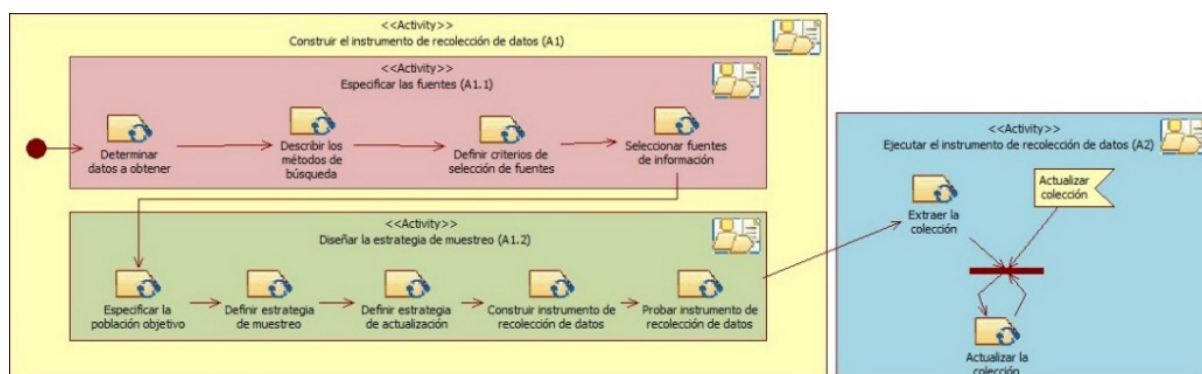


Fig. 4. Perspectiva de comportamiento de SUM4SOFT

La actividad A1 establece las tareas necesarias para generar el instrumento con el cual construir y actualizar las muestras. Las salidas de la actividad son un instrumento de recolección de datos acompañado de la documentación de diseño y las instrucciones de uso. La misma comprende las subactividades: especificar las fuentes (A1.1) y diseñar la estrategia de muestreo (A1.2); e implica tareas como buscar los recursos que proporcionarán los datos, caracterizar la población objetivo, diseñar la estrategia de muestreo, entre otras. El objetivo de la subactividad A1.1 es obtener una lista con las fuentes de los datos a utilizar y una guía simple de acceso a las mismas. Por su parte, el objetivo de la subactividad A1.2 es diseñar las reglas y estrategias para construir y conservar la colección curada de proyectos software partiendo de la guía de acceso a fuentes de la subactividad anterior.

La actividad A2, a partir del instrumento de recolección de datos creado en la A1, tiene como objetivo crear una colección de proyectos software y mantenerla vigente pese a los cambios producidos por el paso del tiempo.

Adicionalmente, se definieron tres roles de acuerdo con las competencias técnicas necesarias para desempeñar las actividades y tareas del modelo propuesto. Primero, el responsable de la colección: experto del dominio responsable de dirigir y supervisar todas las etapas del proceso, quien conoce en profundidad el tema de estudio, permitiéndole satisfacer los objetivos de la investigación. Segundo, el diseñador de los instrumentos de recolección de datos: encargado de las tareas de la subactividad A1.1, y además de construir y probar el instrumento de recolección de datos. Tercero, el analista de la muestra: cuenta con conocimientos en estadística aplicada necesarios para determinar las estrategias de muestreo y actualización, y verificar la composición de las colecciones.

5 Validación

Con el objetivo de demostrar la utilidad de SUM4SOFT en un contexto real, se siguió el marco propuesto para construir el instrumento de recolección de datos JavaQ. En primer lugar, se especificaron los datos requeridos y dónde obtenerlos. A partir de los resultados de los mapeos sistemáticos, se identificaron los metadatos que cuantifican los criterios relevados y se obtuvieron dos posibles plataformas que brindaban acceso a estos datos: Github y Gitlab. Debido a las limitaciones de Gitlab para el filtrado de proyectos por lenguaje de programación, solo se recurrió a Github como fuente.

En segundo lugar, se definieron la población objetivo y las estrategias de muestreo y actualización. La población

objetivo se conformó con aquellos proyectos que cumplieran con los umbrales de la Tabla 1, los cuales operacionalizan los criterios de selección. La lista de proyectos extraída sirvió de marco muestral para recolectar los proyectos aptos para la investigación.

Umbral	Criterio
Lenguaje Java	Proyectos Java
Repositorios públicos	Metadatos disponibles
Repositorios GIT	Uso de herramientas para soporte a procesos
50 incidentes o más	
Repositorio no copiado	Proyectos originales
10000 KB o más	Tamaño
3 o más colaboradores	Colaboración
50 PR o más	
1 año desde creación o más	Historia
1000 confirmaciones o más	
10 bifurcaciones o más	Popularidad
10 estrellas o más	
1 confirmación/PR o más en el último mes	Actividad reciente
1 confirmación o más por mes en el último año	
Repositorio no archivado	

Tabla 1. Umbrales para caracterizar la población objetivo.

En cuanto a la estrategia de muestreo, se empleó muestreo aleatorio estratificado para seleccionar los sujetos en el marco muestral al igual que estudios relacionados [17]. La implementación del procedimiento de selección involucró la generación de los estratos a partir del tamaño del proyecto utilizando el algoritmo para agrupamiento k-means [18]. Antes de aplicar el algoritmo, se separaron los proyectos con valores extremos de tamaño y se aplicó kmeans con cinco grupos. Una vez obtenidos los grupos, se procedió a seleccionar aleatoriamente los proyectos de cada grupo proporcionalmente de acuerdo con el tamaño del estrato y del marco muestral. La Fig. 5 ilustra gráficamente la estrategia.

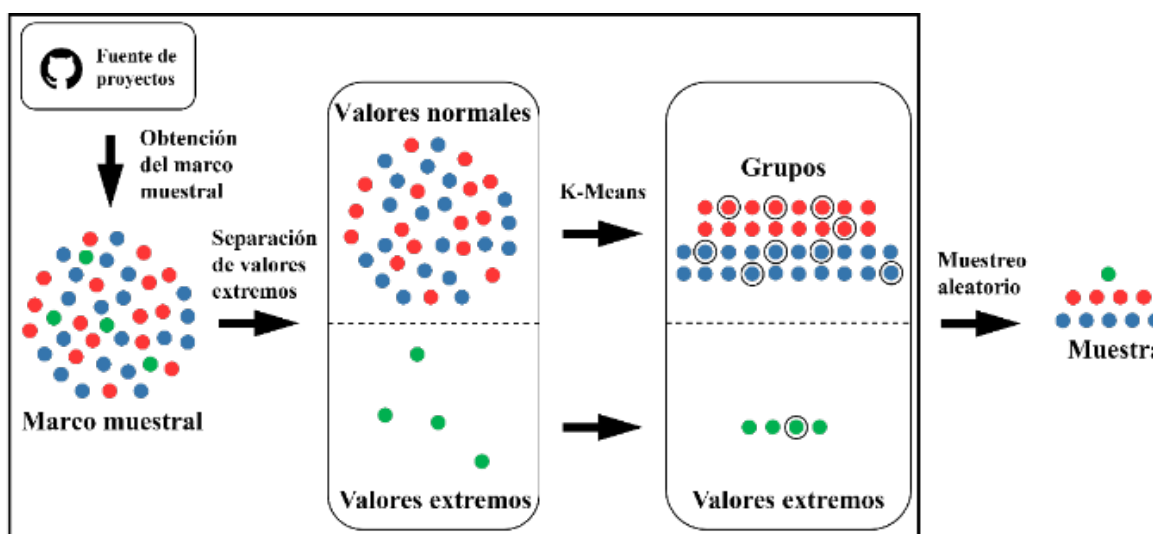


Fig. 5. Esquema de la estrategia de muestreo

Con respecto a la estrategia de actualización, el proceso inicia generando un marco muestral nuevo a partir de los umbrales definidos en la Tabla 1. Si los elementos de la muestra original tienen actividad reciente se procede a

actualizar los proyectos en la muestra por las nuevas versiones disponibles. En cambio, los proyectos sin actualizaciones se descartan de la muestra actualizada. Los proyectos descartados son reemplazados a partir de un proceso similar al muestreo estratificado aleatorio. Primero, se generan los estratos a partir marco muestral nuevo de la misma manera que la estrategia de muestreo. Segundo, se registra la cantidad de proyectos de cada estrato que existen en la muestra actualizada (sin los proyectos descartados) y se obtiene la diferencia con la cantidad proporcional de elementos correspondientes al estrato equivalente en el marco muestral. Tercero, se extraen los proyectos en el marco muestral de acuerdo con las diferencias registradas por grupo en el paso previo.

Esta herramienta se empleó en dos estudios empíricos: un estudio de minería de repositorios y un estudio de caso. El objetivo del estudio de minería de repositorios [19] fue evaluar la efectividad de JavaQ para construir y actualizar colecciones de proyectos. Para ello, se generaron los umbrales de calidad de tres métricas de código: complejidad ciclomática (McCC), cantidad de líneas de código (LOC) y peso de métodos por clase (WMC). Los umbrales de métricas delimitan puntos de referencia aceptables y anormales en los componentes de software, proporcionando un indicador para evaluar el estado del sistema [20]. Para generar los umbrales de las métricas se implementó el enfoque estadístico basado en datos planteado por Alves et al. [21]. Esta técnica establece tres valores de referencia a través de la función de distribución acumulada de la métrica: límite inferior (70%), medio (80%) y superior (90%), delimitando cuatro niveles de riesgo de un componente de sistema: riesgo bajo ($\leq 70\%$), riesgo moderado (70% 80%), riesgo alto (80% 90%) y riesgo muy alto ($> 90\%$).

Se utilizaron tres colecciones para la evaluación: una muestra desactualizada (Qualitas Corpus), una muestra con datos recientes (Colección Vigente) y una versión nueva de la muestra desactualizada (Qualitas Actualizado). La muestra desactualizada corresponde a la versión publicada en 2013 de la colección Qualitas Corpus [22], la colección vigente se obtuvo de aplicar la estrategia de muestreo estratificado, y la tercera es una nueva versión del Qualitas Corpus con la estrategia de actualización.

Para el análisis estadístico se utilizó la prueba de hipótesis KolmogorovSmirnov de dos muestras independientes [23], y esta reveló diferencias significativas entre la colección vigente y el Qualitas Corpus en la distribución de probabilidades de las métricas LOC, McCC y WMC sembrando dudas acerca de la validez del Qualitas Corpus para conducir investigaciones actuales centradas en estas propiedades. En cambio, la comparación de los umbrales obtenidos de las colecciones recolectadas con las estrategias de muestreo y actualización implementadas en JavaQ presentaron diferencias marginales. En la Fig. 6 se trazaron los umbrales de las tres muestras analizadas en el estudio, donde las curvas que corresponden a las colecciones obtenidas de JavaQ (azul y naranja) prácticamente se superponen de principio a fin, y la verde (Qualitas Corpus) tiene una distancia considerable de las demás.

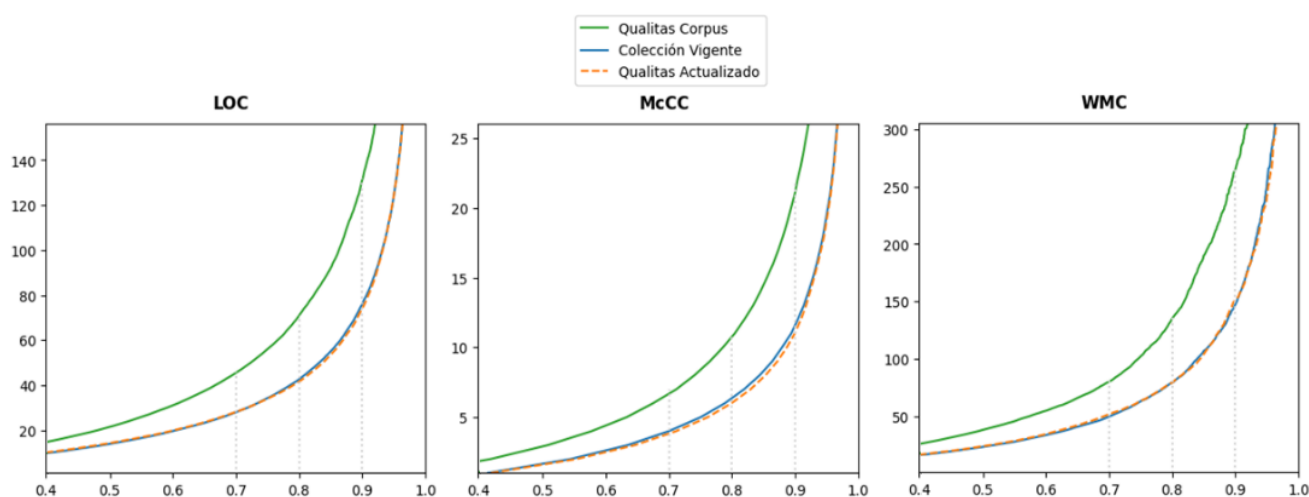


Fig. 6. Funciones de distribución acumulada de cada colección

Para validar los resultados obtenidos en el estudio de minería de repositorios y recibir mejoras para el modelo de proceso se realizó un estudio de caso siguiendo los lineamientos establecidos por Runeson et al. [24]. El estudio se desarrolló en la Universidad de Hamburgo en el marco de una estancia de investigación doctoral.

En la primera etapa, se condujo una replicación del estudio de minería de repositorios enmarcado en el estudio de caso aplicando las estrategias de muestreo implementadas en JavaQ. Al igual que en el estudio original, los investigadores externos extrajeron dos colecciones con proyectos actuales de Github: una colección vigente

¹Las Becas ALEARG son un convenio entre el Ministerio de Educación de la Argentina y el Servicio Alemán de Intercambio Académico para realizar estancias de investigación en universidades alemanas en el marco de estudios doctorales en universidades argentinas

y una versión actualizada del Qualitas Corpus. A partir de ellos se derivaron los umbrales de las métricas LOC, McCC y WMC, y se contrastaron con los obtenidos en el estudio original. El análisis estadístico de los umbrales de referencia obtenidos en la replicación y el estudio original mostraron resultados consistentes para las métricas LOC, McCC, y WMC. Se observaron diferencias mínimas entre las colecciones junto con pvalores mayores a 0.85 en la prueba KolmogorovSmirnov, sugiriendo que las distribuciones de los datos en ambos estudios provienen de la misma población.

El estudio de caso también tuvo como objetivo capturar sugerencias del grupo de investigación externo para refinar las tareas y roles del modelo de proceso propuesto. Las respuestas se analizaron cuantitativa y cualitativa-mente, y se sintetizaron en la Tabla 2. Esto permitió operacionalizar la mejora de las tareas y roles de SUM4SOFT, los cuales fueron incluidos en la versión definitiva del artefacto.

Act.	Tarea	Motivación
A1.1	Describir los métodos de búsqueda	Agregar métodos de búsqueda
A1.2	Especificar la población objetivo	Instrumentar criterios de selección
	Definir estrategia de muestreo	Informar sobre el tamaño de la muestra
	Definir estrategia de actualización	Estandarizar estrategia de actualización
	Probar instrumento de recolección	Verificar el instrumento
A2	Actualizar la colección	Comprobar desviaciones

Tabla 2. Modificaciones de tareas incluidas en el modelo

6 Conclusión y futuros trabajos

La contribución principal de esta investigación fue el desarrollo de herramientas para el soporte de estudios en la ISE. Por un lado, el modelo de proceso SUM4SOFT, y por otro, las salidas del modelo obtenidos en el estudio de minería de repositorios: una colección de proyectos software y el instrumento de recolección de datos JavaQ.

También se profundizó acerca del estado actual de las colecciones de proyectos software utilizadas en la ISE a través de tres estudios empíricos: dos mapeos sistemáticos y un estudio longitudinal. Las revisiones de la literatura confirmaron algunas problemáticas mencionadas por otros autores como: la falta de estrategias de selección de proyectos estandarizadas, la omisión de la vigencia como criterio de selección, la ausencia del muestreo probabilístico y las restricciones de acceso, tanto a las colecciones como a las herramientas empleadas en la experimentación con ellas. A su vez se identificaron las características deseables en los proyectos software para construir una colección curada. Por su parte, el estudio longitudinal midió el impacto del tiempo en una población de proyectos demostrando la importancia de la vigencia en contextos donde la población evoluciona de manera constante.

Como trabajos futuros, se propone validar SUM4SOFT en otros contextos. Si bien el modelo de proceso se validó inicialmente en un estudio de minería y posteriormente en un estudio de caso utilizando el instrumento de recolección de datos JavaQ, probarlo en un contexto de aplicación diferente al de métricas del código fuente le proporcionaría mayor robustez. El objetivo consistirá en monitorear la utilización de SUM4SOFT en un nuevo estudio empírico diseñado por investigadores externos.

Por otra parte, dado que la pérdida de la vigencia de las colecciones es un problema que afecta no solo a las métricas de código fuente y de repositorio, resulta lógico extender el estudio de la vigencia a otros tópicos de la ISE. Por ejemplo, las pruebas del software, la seguridad, refactorización del código, entre otras.

Referencias

- Kitchenham, B., Dyba, T. & Jorgensen, M. Evidence-based software engineering. in Proceedings. 26th International Conference on Software Engineering 273–281 (2004).
- Lehman, M. M. Laws of software evolution revisited. in Lecture Notes in Computer Science. vol. 1149 108–124 (1996).

3. Munaiah, N., Kroh, S., Cabrey, C. & Nagappan, M. Curating GitHub for engineered software projects. *Empir Softw Eng* 22, 3219–3253 (2017).
4. Kalliamvakou, E. et al. The promises and perils of mining GitHub. in *Proceedings of the 11th Working Conference on Mining Software Repositories* 92–101 (2014).
5. Baltes, S. & Ralph, P. Sampling in software engineering research: a critical review and guidelines. *Empir Softw Eng* 27, 94 (2022).
6. Cosentino, V., Luis, J. & Cabot, J. Findings from GitHub. in *Proceedings of the 13th International Conference on Mining Software Repositories* 137–141 (2016).
7. Petersen, K., Feldt, R., Mujtaba, S. & Mattsson, M. Systematic Mapping Studies in Software Engineering. in *Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering* 68–77 (2008). doi:10.14236/ewic/EASE2008.8.
8. Carruthers, J. A., Diaz-Pace, J. A. & Irrazabal, E. A. A Systematic Mapping Study of Empirical Studies Performed with Collections of Software Projects. *Computación y Sistemas* 26, (2022).
9. Carruthers, J. A., Diaz-Pace, J. A. & Irrazabal, E. A. How are software datasets constructed in Empirical Software Engineering studies? A systematic mapping study. in *48th Euromicro Conference on Software Engineering and Advance Applications* 442–450 (2022).
10. Lehman, M. M. Programs, life cycles, and laws of software evolution. *Proceedings of the IEEE* 68, 1060–1076 (1980).
11. Munger, K. The Limited Value of Non-Replicable Field Experiments in Contexts With Low Temporal Validity. *Soc Media Soc* 5, 205630511985929 (2019).
12. Carruthers, J. A., Diaz-Pace, J. A. & Irrazabal, E. A longitudinal study on the temporal validity of software samples. *Inf Softw Technol* 168, 107404 (2024).
13. Kruskal, W. H. & Wallis, W. A. Use of Ranks in One-Criterion Variance Analysis. *J Am Stat Assoc* 47, 583–621 (1952).
14. Johannesson, P. & Perjons, E. An Introduction to Design Science. *An Introduction to Design Science* vol. 9783319106 (2014).
15. Marbán, Ó., Mariscal, G., Menasalvas, E. & Segovia, J. An engineering approach to data mining projects. *Lecture Notes in Computer Science* 4881 LNCS, 578–588 (2007).
16. Curtis, B., Kellner, M. I. & Over, J. Process modeling. *Commun ACM* 35, 75–90 (1992).
17. Gorostidi, J., Ait, A., Cabot, J. & Canovas Izquierdo, J. L. On the Creation of Representative Samples of Software Repositories. in *Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* 434–439 (2024).
18. Bugayenko, Y. et al. Qualitative Clustering of Software Repositories Based on Software Metrics. *IEEE Access* 11, 14716–14727 (2023).
19. Vidoni, M. A systematic process for Mining Software Repositories: Results from a systematic literature review. *Inf Softw Technol* 144, 106791 (2022).
20. Lanza, M. & Marinescu, R. *Object-Oriented Metrics in Practice*. (2006).
21. Alves, T. L., Ypma, C. & Visser, J. Deriving metric thresholds from benchmark data. in *2010 IEEE International Conference on Software Maintenance* 1–10 (2010).
22. Tempero, E. et al. The Qualitas Corpus: A Curated Collection of Java Code for Empirical Studies. in *2010 Asia Pacific Software Engineering Conference* 336–345 (2010).
23. Sheskin, D. J. *Parametric and Nonparametric Statistical Procedures*. (2000).
24. Runeson, P., Höst, M., Rainer, A. & Regnell, B. *Case Study Research in Software Engineering*. (2012). doi:10.1002/9781118181034.

Construyendo Alianza: una práctica ágil para fortalecer la retención estudiantil en Algoritmos y Estructuras de Datos

Noelia Pinto

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Algoritmos y Estructuras de Datos, Ingeniería en Sistemas de Información, Argentina.
ns.pinto@gmail.com

Diego Bolatti

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Algoritmos y Estructuras de Datos, Ingeniería en Sistemas de Información, Argentina.
diegobolatti@gmail.com

Nicolás Tortosa

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Algoritmos y Estructuras de Datos, Ingeniería en Sistemas de Información, Argentina.
nicotortosa@gmail.com

María José Ojeda

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Algoritmos y Estructuras de Datos, Ingeniería en Sistemas de Información, Argentina.
ojedamariaj1569@gmail.com

Resumen

La deserción estudiantil en asignaturas fundamentales de Ingeniería, como Algoritmos y Estructuras de Datos (AED), constituye un desafío persistente en la educación superior. Este trabajo presenta la implementación de “Alianza”, una práctica pedagógica inspirada en el framework Agile Driven Education (ADE), orientada a fortalecer la retención y el compromiso estudiantil. La experiencia, desarrollada en la UTN FRRe, articula valores ágiles —colaboración, iteración y mejora continua— para fomentar un entorno educativo más flexible y participativo. Las estrategias incluyeron acuerdos colaborativos, comunicación continua, ajustes curriculares, talleres de resolución de problemas y retrospectivas ágiles. Los resultados preliminares muestran un incremento del 20% en la asistencia promedio y del 15% en la tasa de aprobación. Se concluye que la integración de prácticas ágiles en la enseñanza de ingeniería promueve una cultura de confianza, responsabilidad compartida y aprendizaje activo, aportando un modelo replicable en otras asignaturas críticas.

Palabras clave: prácticas ágiles, retención estudiantil, compromiso docenteestudiante, retroalimentación continua.

1. Introducción

La deserción estudiantil en los programas de ingeniería, en particular en las asignaturas fundamentales como Algoritmos y Estructuras de Datos (AED), constituye un desafío recurrente en instituciones de educación superior de todo el mundo. Diversos estudios señalan que el abandono temprano no solo afecta la trayectoria académica de los estudiantes, sino que también impacta en el rendimiento institucional y en las tasas de egreso [1,2]. En el caso específico de las disciplinas informáticas, AED se considera una materia bisagra que exige altos niveles de abstracción, razonamiento lógico y competencias de resolución de problemas, lo cual se traduce en elevados índices de desaprobación y abandono.

En la Universidad Tecnológica Nacional, Facultad Regional Resistencia (UTN FRRe), la asignatura AED convoca aproximadamente a 450 estudiantes de primer año durante el ciclo lectivo 2025, distribuidos en cuatro comisiones.

Esta materia ha sido sistemáticamente identificada como una de las más complejas dentro del plan de estudios, registrando los picos de abandono durante el primer cuatrimestre, período crítico en el que los estudiantes deben adaptarse a las exigencias de la vida universitaria. Entre los factores que contribuyen a esta problemática se encuentran la sobrecarga académica, la falta de conexión percibida con los contenidos, la baja motivación y la superposición de evaluaciones en distintas asignaturas [3].

Frente a este escenario, surge la necesidad de implementar enfoques pedagógicos innovadores que trasciendan las metodologías tradicionales. En los últimos años, la incorporación de prácticas ágiles al ámbito educativo ha ganado terreno como una estrategia para promover la adaptabilidad, la colaboración y la mejora continua [4,5]. Principios como la retroalimentación iterativa, la planificación adaptativa y la colaboración entre actores han demostrado su potencial para incrementar el compromiso y la permanencia estudiantil cuando se trasladan del ámbito del desarrollo de software al aula [6,7].

En este marco, el enfoque Agile Driven Education (ADE) ha sido recientemente propuesto como un modelo que articula los valores y principios ágiles con la enseñanza universitaria, planteando al aula como un ecosistema ágil donde la iteración, la comunicación y la cocreación son fundamentales para favorecer el aprendizaje y la retención [8]. Este marco conceptual aporta un fundamento sólido para diseñar e implementar prácticas educativas innovadoras, entre las que se encuentra “Alianza”, una iniciativa que busca transformar la dinámica pedagógica en la asignatura AED.

“Alianza” se materializa en un acuerdo colaborativo entre docentes y estudiantes para cocrear condiciones de aprendizaje que favorezcan la participación activa, la permanencia y el rendimiento académico. A diferencia de intervenciones centradas únicamente en ajustes curriculares, “Alianza” pone el acento en la confianza, la responsabilidad compartida y la comunicación como pilares para mitigar el abandono temprano. Además, la experiencia incorpora retrospectivas periódicas, una práctica característica de la agilidad, que promueven la reflexión conjunta, la evaluación crítica y la mejora continua [9], permitiendo ajustar la propuesta de manera iterativa en función del feedback del propio grupo de aprendizaje.

La contribución de este trabajo es doble: por un lado, presentar el diseño e implementación de “Alianza” como práctica inspirada en la agilidad en la asignatura AED; y por otro, analizar su impacto preliminar en la retención estudiantil, destacando el rol de la retrospectiva como motor de innovación pedagógica sostenible.

Este artículo se organiza del siguiente modo: la Sección 2 revisa antecedentes teóricos y experiencias previas; la Sección 3 describe la metodología y las estrategias de implementación de la práctica “Alianza”; la Sección 4 presenta los resultados preliminares; la Sección 5 aborda la retrospectiva del proceso; finalmente, las Secciones 6 y 7 exponen conclusiones y líneas futuras de trabajo.

2. Trabajos relacionados

El estudio de la deserción estudiantil en la educación superior ha sido ampliamente abordado desde diferentes perspectivas. Tinto [1] propuso un modelo de integración académica y social como factor determinante en la permanencia universitaria, mientras que Kuh et al. [2] identificaron la importancia de factores institucionales como la calidad de la enseñanza, el acompañamiento docente y la construcción de comunidades de aprendizaje. Más recientemente, Braxton [10] ha revisitado estos modelos para destacar la necesidad de estrategias innovadoras y centradas en el estudiante, especialmente en asignaturas con altos índices de abandono.

En el ámbito de las ciencias de la computación, la problemática de la retención cobra mayor relevancia en las asignaturas introductorias de programación y estructuras de datos, donde los índices de desaprobación y abandono suelen superar el promedio de otras disciplinas [11,12]. Estos trabajos señalan que, además de la complejidad intrínseca de los contenidos, la falta de motivación y de acompañamiento adecuado son factores críticos que incrementan la vulnerabilidad del estudiantado en las primeras etapas de la carrera.

Ante este panorama, en los últimos años ha surgido un creciente interés por incorporar enfoques ágiles en la enseñanza universitaria. Fowler y Highsmith [4] introdujeron los valores y principios del Manifiesto Ágil, los cuales han trascendido el ámbito del desarrollo de software para inspirar modelos educativos más flexibles y colaborativos. Mahnic [7] y Bai et al. [6] reportaron experiencias exitosas en la utilización de Scrum y prácticas ágiles en cursos de ingeniería de software, demostrando mejoras en la motivación, la participación activa y el aprendizaje práctico de los estudiantes.

Asimismo, Kropp et al. [5] describen cómo la docencia puede beneficiarse al incorporar enfoques iterativos y retrospectivos en la planificación académica, promoviendo una cultura de aprendizaje continuo y de adaptación

a las necesidades del estudiantado. Estos aportes se alinean con las propuestas de Derby y Larsen [9], quienes plantean la retrospectiva como una herramienta clave para la mejora continua, no solo en equipos de desarrollo de software, sino también en contextos colaborativos más amplios.

Sin embargo, pese a los avances señalados, persiste una brecha en la literatura respecto a la aplicación sistemática de prácticas ágiles para abordar el problema específico de la deserción estudiantil en materias críticas del ciclo básico de ingeniería. La mayoría de los trabajos se concentran en experiencias de enseñanza de proyectos o cursos avanzados de desarrollo de software, dejando en segundo plano la potencialidad de la agilidad como estrategia pedagógica transversal en asignaturas con altas tasas de abandono.

En este sentido, el presente trabajo contribuye a cubrir esa brecha al presentar “Alianza” como una práctica ágil diseñada específicamente para mejorar la retención en la asignatura Algoritmos y Estructuras de Datos, integrando elementos de colaboración docenteestudiante y retrospectivas periódicas como mecanismos para fortalecer el vínculo pedagógico y favorecer la permanencia.

3. Metodología: la práctica “Alianza”

La práctica “Alianza” se diseñó como una estrategia pedagógica colaborativa incluida en las prácticas propuestas en el framework Agile Driven Education (ADE) [8], cuyo propósito es trasladar al ámbito educativo los valores de adaptabilidad, colaboración, comunicación y mejora continua característicos de la agilidad. Su implementación en la asignatura Algoritmos y Estructuras de Datos respondió a la necesidad de crear un entorno de aprendizaje más inclusivo, participativo y sostenible frente a los altos índices de abandono observados en los primeros tramos del cursado.

El proceso metodológico se desarrolló en tres etapas: diagnóstico participativo, coconstrucción de la Alianza y ciclos de retrospectiva.

3.1 Diagnóstico participativo

En la primera instancia, el equipo docente realizó encuestas y conversaciones abiertas en las cuatro comisiones de la asignatura, con el fin de identificar percepciones y dificultades vinculadas a la cursada. Los resultados reflejaron coincidencias entre grupos: sobrecarga académica, falta de motivación, escasa conexión entre teoría y práctica, y sensación de distancia en la comunicación con el equipo docente. Este relevamiento inicial permitió establecer las bases para diseñar una práctica adaptada a las necesidades reales del estudiantado.

3.2 Coconstrucción de la Alianza

En la segunda etapa, se organizaron reuniones presenciales con cada comisión, en las que se explicó el objetivo de la práctica y su vínculo con los principios del enfoque ágil: colaboración, transparencia y compromiso compartido. A partir de esta introducción, se abrió un espacio de escucha activa donde los estudiantes pudieron expresar libremente expectativas, frustraciones y propuestas de mejora. Las ideas y compromisos emergentes fueron anotados en la pizarra, conformando una lista visible y colectiva de acuerdos posibles.

Tras unos veinte minutos de intercambio, se inició una dinámica participativa de votación abierta, donde los propios estudiantes priorizaron los compromisos que consideraban más relevantes para sostener la cursada. Cada propuesta se debatió brevemente, permitiendo profundizar en los fundamentos y establecer compromisos realistas desde ambos lados —docente y estudiantil—. Este ejercicio, basado en el principio ágil de “inspeccionar y adaptar”, favoreció la construcción de acuerdos significativos y generó un ambiente de confianza mutua.

La sesión también habilitó un espacio de descargo emocional y reflexivo, donde se abordaron actitudes o situaciones que interferían en el clima de aula. Este momento, lejos de ser marginal, se transformó en un punto de inflexión: permitió reconocer tensiones, validar experiencias y reforzar la empatía dentro del grupo. La práctica de Alianza se consolidó así como un proceso de negociación colectiva y madurez relacional que, además de definir reglas de convivencia, redefinió la cultura del aula.

3.3 Iteraciones y retrospectivas

Durante el cuatrimestre, cada ciclo de cursado concluyó con una retrospectiva ágil, instancia de reflexión colectiva adaptada del contexto de desarrollo de software. En ellas, estudiantes y docentes analizaron los avances del grupo, identificaron obstáculos y definieron acciones concretas para el próximo período. Los compromisos revisados eran registrados por docentes responsables del dictado de contenidos teóricos y retomados en la siguiente sesión, garantizando trazabilidad y continuidad.

El diseño metodológico también incluyó el uso de canales digitales —foros, mensajería y aulas virtuales— para mantener una comunicación continua y abierta. Estas herramientas facilitaron la coordinación entre comisiones, la resolución de dudas y la publicación transparente de los acuerdos alcanzados.

La implementación de la práctica contó con el respaldo institucional de la Dirección de Planeamiento y la Secretaría Académica de la Facultad Regional Resistencia, que facilitaron la coordinación entre áreas y legitiman la experiencia como parte de las políticas de innovación pedagógica de la UTN FRRe. Esta articulación resultó esencial para sostener la continuidad del proyecto, permitir ajustes administrativos y fortalecer el impacto de la propuesta dentro del plan de mejora de la enseñanza en ingeniería.

En la Figura 1, se observa la práctica de discusión entre estudiantes y luego los resultados visualizados en la pizarra de clase. Más tarde, el resumen se difundió en el aula virtual de la asignatura “Algoritmos y Estructuras de datos” por comisión.



Figura 1. Captura de momentos de la práctica en clase

La metodología de “Alianza” combinó diagnóstico participativo, acuerdos colaborativos, comunicación continua, retroalimentación iterativa y apoyo institucional. Este enfoque integrador y adaptativo permitió transformar la dinámica de aula tradicional en un ecosistema de aprendizaje ágil, centrado en las personas y orientado a la mejora continua.

4. Análisis de Resultados

El análisis de los resultados obtenidos a partir de la implementación de la práctica “Alianza” permite evaluar su impacto en la retención, la motivación y la participación de los estudiantes de Algoritmos y Estructuras de Datos. Para ello se consideraron tanto indicadores cuantitativos como cualitativos, vinculados con la percepción del acompañamiento docente, el clima de aula y el desarrollo de competencias transversales.

El seguimiento se realizó durante el primer cuatrimestre del ciclo lectivo, con la participación de las cuatro comisiones que conforman la cursada, lo que permitió contrastar los resultados con datos históricos de la asignatura y con observaciones sistemáticas realizadas por el equipo docente.

4.1 Resultados cuantitativos

El análisis cuantitativo de la práctica “Alianza” se realizó sobre la totalidad de las comisiones de Algoritmos y Estructuras de Datos ($n = 4$) durante el primer cuatrimestre del ciclo lectivo 2025, considerando tres indicadores principales: asistencia promedio, y tasa de aprobación. Estos resultados se contrastaron con los registros históricos de los tres años anteriores (2022–2024), en los cuales no se había implementado la práctica.

Asistencia

La asistencia promedio a clases presenciales y virtuales mostró un incremento sostenido del 20% respecto de los períodos previos. Mientras que en el ciclo 2024 el promedio de asistencia semanal rondaba el 58%, durante la implementación de Alianza se alcanzó un 78% (Figura 2). Este aumento se mantuvo estable durante todo el cuatrimestre, lo que sugiere un efecto positivo de la comunicación continua y del sentimiento de corresponsabilidad generado en el grupo. Las retrospectivas periódicas actuaron como un mecanismo de refuerzo, ya que cada estudiante podía observar la evolución colectiva del curso y su propio grado de participación.

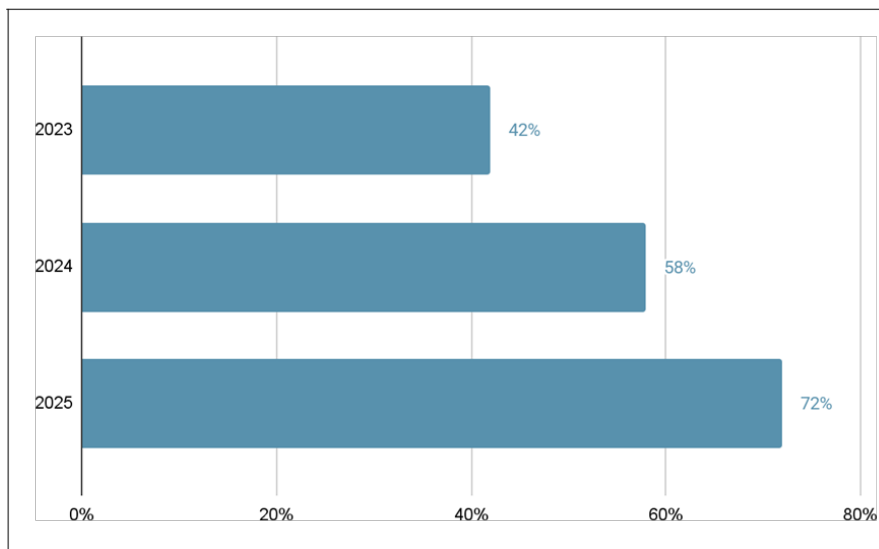


Figura 2. Variación de tasa de asistencia a clases

Aprobación

En cuanto al rendimiento académico, la tasa de aprobación de evaluaciones parciales se incrementó del 12% al 15% aproximadamente, representando una mejora relativa del 25%. Este dato es particularmente relevante si se considera que los instrumentos de evaluación mantuvieron el mismo nivel de complejidad que en años anteriores. El aumento de la aprobación, que se muestra en la figura 3, se relaciona con una mejor continuidad en la cursada, la reducción del ausentismo y los compromisos acordados a través de la “Alianza”. También se observa una reducción en el ausentismo al momento de rendir el primer examen integrador.

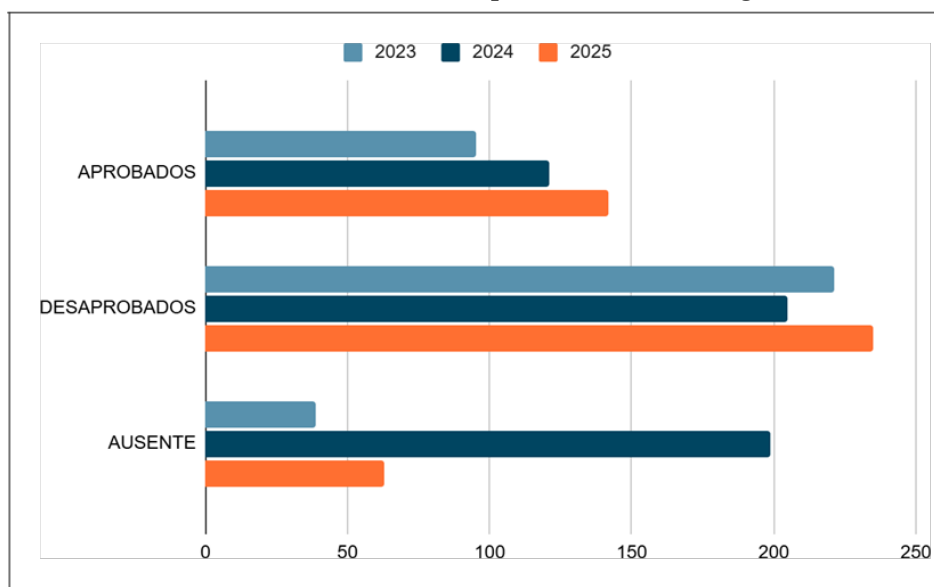


Figura 3. Variación de tasa de asistencia a clases

4.2 Resultados cualitativos

Desde una perspectiva cualitativa, las encuestas y entrevistas realizadas al cierre del cuatrimestre muestran un cambio positivo en la percepción general del proceso de enseñanza-aprendizaje. Los estudiantes destacaron la sensación de acompañamiento docente, la disponibilidad para la consulta en tiempo real y la relevancia práctica de los contenidos como factores clave de su permanencia en la asignatura.

La práctica “Alianza” generó un entorno de aprendizaje donde predominó la confianza y la colaboración, reduciendo la distancia tradicional entre docentes y estudiantes. Este vínculo horizontal favoreció la participación activa, la formulación de preguntas y la exposición de dificultades sin temor a la evaluación. Además, la incorporación de retrospectivas periódicas fue señalada como una de las instancias más valiosas, ya que permitió a los estudiantes sentirse escuchados y partícipes en la mejora del curso.

Desde el punto de vista docente, se registró una mejora en la gestión del aula y una mayor capacidad para ajustar estrategias pedagógicas de forma iterativa. La retroalimentación constante permitió identificar patrones de dificultad comunes y rediseñar actividades en tiempo real, fortaleciendo la respuesta pedagógica ante los desafíos emergentes.

4.3 Resultados organizativos e institucionales

En el plano organizativo, la implementación de “Alianza” contribuyó a consolidar una cultura de trabajo colaborativo dentro del equipo docente y entre los distintos espacios académicos vinculados con la asignatura. La práctica favoreció la planificación compartida, la revisión continua de materiales y la alineación entre teoría y práctica.

El respaldo institucional de la Dirección de Planeamiento y la Secretaría Académica de la UTN FRRe resultó determinante para viabilizar ajustes administrativos y garantizar la sostenibilidad de la experiencia. Este apoyo permitió gestionar tiempos, recursos y coordinación interdepartamental, generando una sinergia entre la innovación pedagógica y las políticas institucionales de mejora continua.

Asimismo, la experiencia fortaleció la visibilidad de la asignatura como espacio de innovación educativa dentro de la Facultad, abriendo la posibilidad de replicar el enfoque ágil en otras materias del primer año de Ingeniería en Sistemas de Información.

5. Retrospectiva

La retrospectiva, entendida como una práctica central dentro de los enfoques ágiles, constituye un espacio sistemático y seguro de reflexión colectiva orientado a la mejora continua. En el contexto educativo, esta instancia se resignifica como una herramienta pedagógica que permite evaluar el proceso de enseñanza-aprendizaje desde una perspectiva colaborativa, identificando logros, dificultades y oportunidades de optimización. En la práctica “Alianza”, las retrospectivas se realizaron de manera periódica cada tres semanas y se convirtieron en un componente esencial para sostener la coherencia entre los objetivos académicos, las necesidades de los estudiantes y la evolución del curso. Es importante destacar que la eficacia de una retrospectiva está directamente relacionada con la capacidad del docente de generar un ambiente con seguridad psicológica [13], donde los estudiantes pueden expresarse sin miedo a represalias.

Durante estas reuniones, docentes y estudiantes participaron activamente en el análisis de las estrategias implementadas, expresaron percepciones sobre el desarrollo del curso y propusieron ajustes concretos para el siguiente ciclo de trabajo. Este proceso de reflexión iterativa no solo promovió la transparencia y la confianza, sino que también fortaleció la autonomía y la corresponsabilidad en la gestión del aprendizaje. La dinámica generó un clima de apertura donde las opiniones de los estudiantes fueron consideradas insumos valiosos para la toma de decisiones pedagógicas, reforzando el principio ágil de que “las personas y sus interacciones están por encima de los procesos y las herramientas”.

El valor diferencial de “Alianza” radica en haber trasladado con éxito la filosofía de la retrospectiva desde el ámbito del desarrollo de software al entorno educativo, configurándola como un mecanismo de retroalimentación continua que guía la evolución de la asignatura. Gracias a este enfoque, fue posible ajustar en tiempo real los contenidos, metodologías y evaluaciones, asegurando una alineación constante entre las expectativas docentes y las experiencias de aprendizaje de los estudiantes. Este proceso no solo incrementó la eficiencia académica, sino que también consolidó una cultura de mejora compartida que impactó positivamente en la cohesión del grupo y en el sentido de pertenencia institucional.

Desde la perspectiva docente, las retrospectivas permitieron identificar patrones recurrentes de dificultad, diseñar intervenciones más personalizadas y optimizar la coordinación entre comisiones y equipos de cátedra. A nivel organizativo, se consolidó un aprendizaje institucional que trasciende la asignatura, demostrando que la aplicación sistemática de principios ágiles puede constituir una estrategia sostenible de innovación pedagógica. La posibilidad de evaluar el curso de manera iterativa y participativa contribuyó a fortalecer la resiliencia del equipo docente, generando una actitud de adaptación permanente frente a los desafíos de la enseñanza en ingeniería.

El valor pedagógico de “Alianza” se evidencia, por tanto, en tres dimensiones complementarias: en primer lugar, en la dimensión relacional, al fomentar la confianza y la comunicación bidireccional entre docentes y estudiantes; en segundo lugar, en la dimensión organizativa, al promover la coordinación interna, la flexibilidad curricular y la toma de decisiones basada en evidencia; y, finalmente, en la dimensión formativa, al desarrollar en los estudiantes competencias transversales como la autogestión, la reflexión crítica y la colaboración. Estas dimensiones convergen en un modelo de aula ágil que integra la retroalimentación continua como principio estructurante del proceso educativo.

Como parte del enfoque de mejora continua propuesto por el framework ADE, luego del primer examen integrador se realizó una nueva instancia de consulta dirigida a los y las estudiantes que no alcanzaron la aprobación. El objetivo fue indagar las percepciones sobre las dificultades experimentadas y recoger propuestas de mejora para el siguiente ciclo de cursado. Las respuestas obtenidas permitieron identificar aspectos vinculados al manejo del tiempo, la integración de contenidos y la ansiedad ante la evaluación, lo que derivó en la cocreación de nuevas estrategias de acompañamiento y ajuste de cronogramas. Este proceso reafirma la naturaleza iterativa de la práctica “Alianza”, donde el aprendizaje no concluye con la evaluación, sino que se retroalimenta a partir de ella.

En definitiva, la retrospectiva dentro de la práctica “Alianza” trasciende el carácter evaluativo para constituirse en un verdadero instrumento de innovación pedagógica y de gestión del conocimiento educativo, capaz de transformar la relación entre enseñanza, aprendizaje y mejora institucional. Su implementación demuestra que los principios ágiles pueden ser aplicados con éxito en la educación en ingeniería para construir entornos más adaptativos, inclusivos y sostenibles, donde la reflexión compartida se convierte en el motor principal del aprendizaje significativo.

6. Conclusiones

La experiencia desarrollada con la práctica “Alianza” en la asignatura Algoritmos y Estructuras de Datos permitió evidenciar que la incorporación de principios ágiles al ámbito de la educación universitaria no solo es posible, sino también altamente efectiva para abordar problemáticas estructurales como la deserción temprana, la desmotivación y la desconexión entre docentes y estudiantes. El enfoque adoptado, inspirado en el framework Agile Driven Education (ADE), contribuyó a transformar el aula tradicional en un espacio dinámico de aprendizaje, caracterizado por la colaboración, la comunicación transparente y la mejora continua.

Los resultados obtenidos demostraron un incremento sostenido en la retención, la asistencia y el rendimiento académico, lo que refleja la efectividad del modelo en términos cuantitativos. Sin embargo, el aporte más relevante de la experiencia radica en su dimensión cualitativa y cultural: la práctica “Alianza” generó un cambio profundo en la concepción del proceso educativo, promoviendo un modelo de corresponsabilidad en el que docentes y estudiantes comparten objetivos, responsabilidades y decisiones. Esta transformación de la dinámica pedagógica favoreció el desarrollo de vínculos más sólidos, una mayor autonomía estudiantil y un sentido de pertenencia que fortaleció la cohesión del grupo y el compromiso con el aprendizaje.

Desde el punto de vista docente, la experiencia impulsó una reflexión crítica sobre las propias prácticas de enseñanza, promoviendo la adopción de metodologías iterativas y flexibles que facilitan la adaptación a las necesidades reales del estudiantado. La introducción de retrospectivas periódicas, inspiradas en la cultura ágil, se consolidó como una herramienta estratégica para la mejora continua, permitiendo identificar tempranamente dificultades, ajustar procesos y evaluar el impacto de las decisiones pedagógicas en tiempo real. Este proceso de retroalimentación constante fomentó una cultura de innovación que trascendió el aula, impactando también en la organización del equipo docente y en la gestión académica de la materia.

A nivel institucional, la práctica “Alianza” demostró la importancia de articular la innovación pedagógica con la gestión académica. El acompañamiento de la Dirección de Planeamiento y la Secretaría Académica de la UTN Facultad Regional Resistencia fue un factor clave para garantizar la continuidad, la legitimidad y la sostenibilidad

de la propuesta. Esta articulación permitió validar la experiencia como parte de un proceso institucional de mejora de la calidad educativa, fortaleciendo el compromiso de la universidad con la formación integral del estudiantado y con la implementación de enfoques educativos innovadores basados en evidencia.

El valor de “Alianza” no se limita a los resultados inmediatos, sino que reside en su potencial de replicabilidad y escalabilidad. Su estructura metodológica —basada en el diagnóstico participativo, la comunicación continua, la flexibilidad curricular y la retroalimentación iterativa— puede adaptarse a otras asignaturas del ciclo básico de ingeniería que presenten problemáticas similares. La experiencia ofrece, así, un modelo de gestión del aprendizaje ágil que integra principios de la ingeniería del software con prácticas docentes centradas en las personas, demostrando la capacidad del enfoque ADE para articular teoría, práctica e innovación educativa.

De cara al futuro, se prevé profundizar la evaluación longitudinal de la práctica, a fin de medir el impacto sostenido de “Alianza” en la trayectoria académica de los estudiantes y en su desarrollo de competencias transversales, como la autogestión, la colaboración, la comunicación efectiva y la resiliencia. Este seguimiento permitirá no solo validar los logros alcanzados, sino también perfeccionar los mecanismos de implementación y evaluación del enfoque.

Asimismo, se proyecta extender la experiencia a otras materias del primer año de Ingeniería en Sistemas de Información, en un proceso de co-construcción con otros equipos docentes. La ampliación permitirá analizar la adaptabilidad del modelo en contextos diversos y promover una cultura institucional de aprendizaje ágil, orientada a la calidad, la inclusión y la sostenibilidad del proceso educativo. En esta línea, resulta fundamental avanzar hacia el diseño de métricas académicas inspiradas en la agilidad, tales como tableros de progreso, indicadores de participación o velocidad de aprendizaje, que posibiliten monitorear en tiempo real la evolución del curso y ofrecer retroalimentación oportuna y significativa.

En paralelo, se contempla la formalización de las retrospectivas docentes como práctica institucional, incorporándolas a los procesos de planificación y evaluación académica. Esta estrategia busca consolidar un modelo de mejora continua sistemático, que trascienda las experiencias aisladas y fortalezca el enfoque de calidad educativa a nivel de facultad y universidad.

Finalmente, se plantea explorar de manera más profunda el impacto de la práctica en el desarrollo de habilidades socioemocionales y profesionales en el estudiantado, reconociendo que la educación en ingeniería no solo debe garantizar la adquisición de conocimientos técnicos, sino también promover competencias humanas esenciales para el ejercicio profesional contemporáneo. La comunicación efectiva, la adaptabilidad, el trabajo colaborativo y la gestión del cambio son hoy capacidades críticas que pueden ser potenciadas a través de un modelo educativo ágil, centrado en las personas y en la mejora continua.

En conclusión, la práctica “Alianza” constituye una evidencia concreta del potencial transformador de los enfoques ágiles aplicados a la educación en ingeniería. Su impacto trasciende los resultados académicos, al generar una nueva forma de concebir el aprendizaje como un proceso participativo, reflexivo y evolutivo. El desafío futuro será consolidar este paradigma en la estructura institucional y extenderlo a otros espacios curriculares, avanzando hacia una universidad que aprenda, se adapte y evolucione con la misma agilidad con la que forma a sus futuros ingenieros.

Referencias

1. Tinto, V.: *Leaving College: Rethinking the Causes and Cures of Student Attrition*. 2nd edn. University of Chicago Press (1993).
2. Kuh, G.D., Kinzie, J.L., Buckley, J.A., Bridges, B.K., Hayek, J.C.: *What matters to student success: A review of the literature*. National Post-secondary Education Cooperative (2006).
3. Martínez, D. L. L. R., Karanik, M., Giovannini, M., Báez, M. E., & Torre, J. (2016). Descubrimiento de perfiles de rendimiento estudiantil: un modelo de integración de datos académicos y socioeconómicos. *Cam-pus Virtuales*, 5(2), 70-83.
4. Fowler, M., Highsmith, J.: The agile manifesto. *Software Development* 9(8), 28–35 (2001).
5. Kropp, M., Meier, A., Biddle, R.: Agile educator: experiences teaching agile approaches in education. In: *Agile Conference*. pp. 145–150. IEEE (2016).
6. Bai, X., Chen, Y., Li, M.: Applying agile methodology in a software engineering capstone course. *IEEE Trans. Educ.* 60(1), 52–59 (2017).
7. Mahnic, V.: A capstone course on agile software development using Scrum. *IEEE Trans. Educ.* 55(1), 99–106

(2012).

8. V. A. Bollati, M. Arias, V. S. Sandobal Verón, L. Cuenca Plestch, “Una estrategia ágil de enseñanza-aprendizaje”, 11avo Congreso Nacional de Ingeniería Informática Sistemas de Información. 2022
9. Derby, E., Larsen, D.: Agile Retrospectives: Making Good Teams Great. Pragmatic Bookshelf (2006).
10. Braxton, J.M.: Leaving college: Rethinking the causes and cures of student attrition by Vincent Tinto. J. College Student Development 60(1), 129–134 (2019).
11. Lister, R.: Concrete and other neo-Piagetian forms of reasoning in the novice programmer. In: Proceedings of the 13th Koli Calling International Conference on Computing Education Research. pp. 1–8. ACM (2011).
12. Watson, C., Li, F.W.B.: Failure rates in introductory programming revisited. In: ITiCSE. pp. 39–44. ACM (2014).
13. Kislyakov, P. A., Shmeleva, E. A., Karaseva, T. Y. V., & Silaeva, O. G. A.: Monitoring of education environment according to the social-psychological safety criterion. Asian Social Science (2014).

IRIR: Índice de Riesgo de Incendios Rurales Alternativo al Índice Meteorológico de Peligro de Incendios Forestales (FWI)

Carlos Brys

ORCID 0000-0002-5872-0444

researchid.co/carlosbrys

Facultad de Ciencias Económicas, Universidad Nacional de Misiones, Posadas, Misiones, Argentina

carlos.brys@fce.unam.edu.ar

Resumen

Se presenta un Índice de Riesgos de Incendios Rurales alternativo al Índice Meteorológico de Peligro de Incendios Forestales (FWI), que integra el aprendizaje automático y la lógica difusa para mejorar la evaluación del potencial de ignición en diversos tipos de vegetación en entornos rurales. Se presenta una metodología que incluye el desarrollo del índice de inflamabilidad para cada zona y la visualización de los resultados. Se describen los datos satelitales utilizados, la clasificación de la cobertura del suelo mediante aprendizaje automático y la definición del índice de inflamabilidad. Los hallazgos revelan que, durante un periodo de comparación de tres meses, el FWI subestimó el riesgo en el 20,70% de los casos, lo sobreestimó en el 34,90% y lo alineó con precisión en el 44,69%. Este modelo mejora la precisión de la evaluación y ofrece a los gestores de recursos naturales y a las autoridades locales herramientas eficaces para tomar decisiones informadas sobre estrategias de prevención y mitigación de incendios. La investigación demuestra que el índice propuesto mejora significativamente la precisión de la predicción al incorporar la inflamabilidad de diversas vegetaciones y variables climáticas localizadas.

Palabras clave: Prevención de incendios · Incendios rurales · Calificación del riesgo de incendio

1. Introducción

El riesgo de un incendio rural es una evaluación de la probabilidad de que se produzcan daños como consecuencia del fuego. El riesgo no es una percepción, sino una valoración. Lo que se percibe es el peligro. Por lo tanto, un peligro de incendio forestal es el potencial de daño físico y ambiental a las formaciones vegetales cuando se quema sin control [1]. La diferencia entre riesgo y peligro es que el riesgo es la combinación de la probabilidad de que ocurra un evento peligroso y la gravedad del daño o deterioro que el evento o exposición pueda causar; y peligro es una fuente, situación o acto con el potencial de causar daño [2].

El FWI desarrollado por el Servicio Forestal de Canadá en 1968 [3] es un componente de importancia para evaluar el peligro y la propagación de incendios en áreas forestales.

En este trabajo se propone un algoritmo computacional alternativo al FWI basado en información local, utilizando aprendizaje automático e imágenes satelitales para clasificar el uso del suelo, datos en línea de teledetección y estaciones meteorológicas para calcular valores específicos para cada municipio, y basado en

la inflamabilidad ponderada del combustible del suelo, usando la lógica difusa clasificar los niveles de riesgo de ignición para asignar etiquetas lingüísticas a los diferentes niveles de riesgo en un formato comprensible para los ciudadanos.

2. Métodos

Se propone un modelo computacional para medir el riesgo potencial de ignición de la hojarasca senescente, asociado con diferentes tipos de cobertura del suelo en regiones específicas, y determinar un nuevo índice de riesgo de incendios.

La propuesta consiste en analizar la cobertura del suelo de cada uno de los municipios en el que se divide

políticamente la provincia para obtener un valor ponderado que represente la inflamabilidad de la hojarasca de cada cubierta vegetal, determinar un valor referencial del riesgo potencial de incendio asociado, para obtener un valor del Índice de Riesgo de Incendios Rural (IRIR).

2.1 Materiales para Calcular el IRIR

Para el análisis y procesamiento de la cobertura del suelo se generó un mosaico de imágenes satelitales de la misión Copernicus. Se utilizaron imágenes de radar de Sentinel 1 Ground Range Detection (GRD), con polarización dual VV+VH (vertical-horizontal), y del satélite Sentinel 2 se usaron imágenes multispectrales para los estudios de vegetación como el NDVI, EVI, SAVI, entre otros.

Para el análisis y procesamiento de la cobertura del suelo y como materiales iniciales para calcular el IRIR, se generó un mosaico de imágenes satelitales a partir de los datos de Sentinel 1 (imágenes de radar GRD) y Sentinel 2 (imágenes multispectrales) de la misión Copernicus. Estas imágenes se usaron para estudios de vegetación (como NDVI, EVI y SAVI). Las bandas e índices espectrales se agruparon, y se incorporó un modelo digital de elevación (SRTM) para corregir distorsiones. Se seleccionaron solo imágenes con una cobertura de nubes inferior al 20%.

2.2 Uso del suelo

Se definieron 22 categorías (clases) de cobertura terrestre para la clasificación del uso de la tierra: pastizales, matorrales, pradera, selva, coníferas, eucaliptos, kiri, tabaco, té, yerba mate, maíz, caña de azúcar, mandioca, soja, cítricos, ananá, bambú, suelo descubierto, agua y humedales.

Se realizó una clasificación por aprendizaje automático supervisado usando el algoritmo *Random Forest* [4], [5], obteniendo los polígonos correspondientes a cada tipo de cultivo de interés como se muestra en la Fig. 1. Las áreas de entrenamiento fueron divididos en 70% para aprendizaje del modelo y 30% para validación. Los resultados de la clasificación obtuvieron unos índices de precisión globales de *Random Forest* del 0,87% y un coeficiente Kappa del 0,85%. Con el postprocesamiento de los polígonos se determinó el tamaño del área en Km² de cada uno y el porcentaje de cobertura de cada tipo en el municipio analizado.

La suma de los índices de inflamabilidad de cada tipo de combustible multiplicados por su porcentaje de cobertura en el área total de cada municipio es el valor del índice de inflamabilidad del municipio IIM.

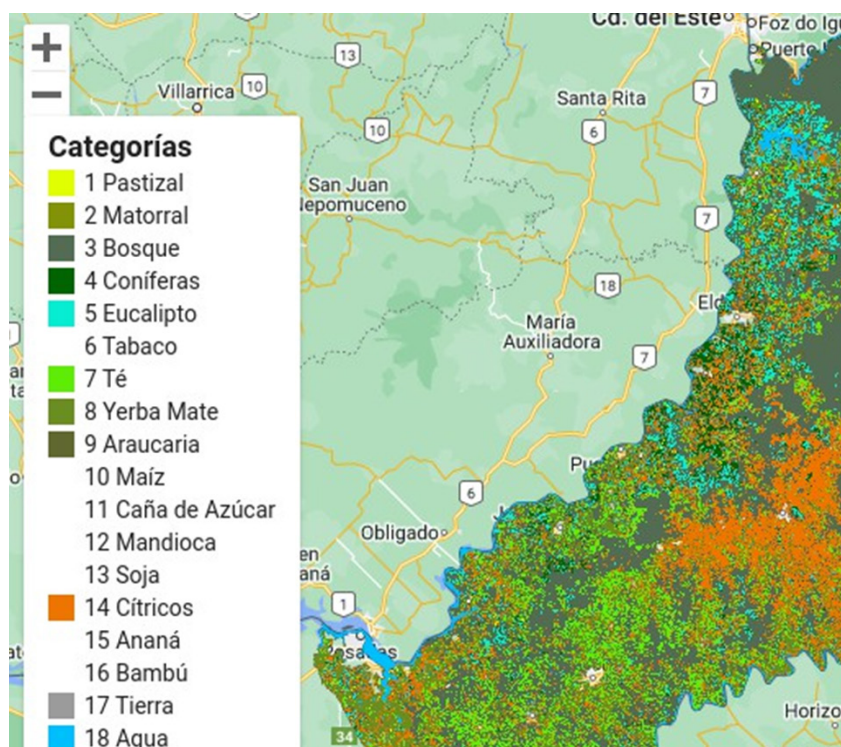


Fig. 1: Clasificación de usos del suelo

2.3 Inflamabilidad

La inflamabilidad se refiere a la capacidad de los combustibles para encenderse y arder. Los Modelos de Combustibles Forestales (MCF) son herramientas utilizadas para predecir el comportamiento del fuego en áreas determinadas, considerando factores como el tipo, la cantidad, la distribución, la humedad y las condiciones meteorológicas del combustible. Los modelos de combustibles forestales utilizados frecuentemente son el modelo de Scott y Burgan [6], el modelo de Rothermell [3] y el modelo de Anderson-Kaufmann [7].

Para el cálculo de la propagación del fuego sobre suelos combustibles considerando los distintos usos del suelo se usaron los valores proporcionados por Brys et al. [8] en [9] y [10]. Estos valores se combinan con las observaciones empíricas recopiladas por los responsables del manejo del fuego.

Se define a la velocidad de propagación horaria vph , como el tiempo medido en *minutos decimales* que tarda el combustible en suelo en arder en un círculo de 1 metro de diámetro, multiplicado por 60, iniciando la combustión en el centro del círculo de quema.

$$vph = \left(\frac{0.5}{tpp} \right) * 60 \quad (1)$$

Donde: tpp es el tiempo promedio de propagación.

Se define el índice de inflamabilidad diferencial ID_c de cada tipo de combustible c , como el número entero de la inversa de la velocidad de propagación horaria vph multiplicada por 100.

$$ID_c = ent \left(\frac{1}{vph_c} * 100 \right) \quad (2)$$

2.4 Factor de la humedad del combustible del suelo (FHS)

La deuda hídrica es la diferencia entre la cantidad de lluvia que debería haber caído en un área determinada y la cantidad que cayó.

Dependiendo de la época del año, los expertos estiman que en verano el suelo se seca (<5% de humedad superficial) en 2 días después de una lluvia y en invierno el suelo se seca en 6 días. Con base en estas condiciones se determina en cuántos días se seca el combustible calculando el valor de días de secado (ds) considerando el día ordinal de la fecha de cálculo como se muestra en la Fig. 2. Se tomaron como referencia los valores diarios de evaporación superficial observados en el ciclo anual 2022 que registró la Estación Experimental Agrícola "INTA Cerro Azul" [11]. Se usaron técnicas de aprendizaje automático para entrenar un modelo para ajustar una función sinusoidal periódica que represente el comportamiento de la evaporación como se muestra en el cuaderno Colab:

Climatología EE INTA Cerro Azul 2022.

Al final de un evento de lluvia, comienza el proceso de evapotranspiración de la humedad superficial del suelo. Este proceso de secado del suelo sigue un patrón determinado por una función exponencial decreciente en el tiempo, como lo demuestran Rondinelli et al. [12] y Kurc et al. [13].

La cantidad de tiempo en que se seca el suelo saturado de agua es la inversa de la evaporación superficial de la humedad del suelo. Se dedujo el número mínimo de días para que el suelo y el combustible se sequen según el día del año en el hemisferio sur (la temporada húmeda ocurre aproximadamente en los meses de noviembre a marzo) como se muestra en la Fig. 2. La fórmula que describe la curva de sequedad del suelo se presenta en la ecuación 3.

$$ds = 2 * \sin \left(2 * \pi * \left(\frac{(x - 93)}{366} \right) \right) + 4 \quad (3)$$

Donde:

x es el valor ordinal del día del cálculo (desde 1 hasta 366)

Esta fórmula resulta de invertir la curva de evaporación (dando como resultado la Fig. 2). Usando el método

de mínimos cuadrados de la función sinusoidal, tomando como parámetros la amplitud, el período y la fase, se derivó la fórmula que se usa para calcular el valor de d_s .

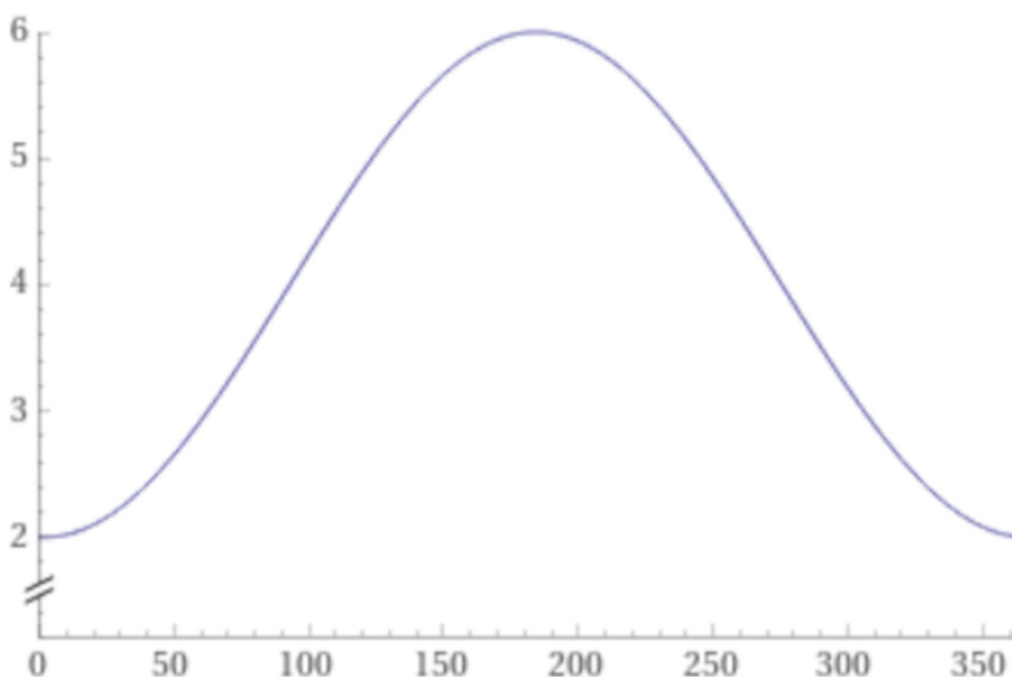


Fig. 2: Número mínimo de días para que el suelo se seque según el número del día del año

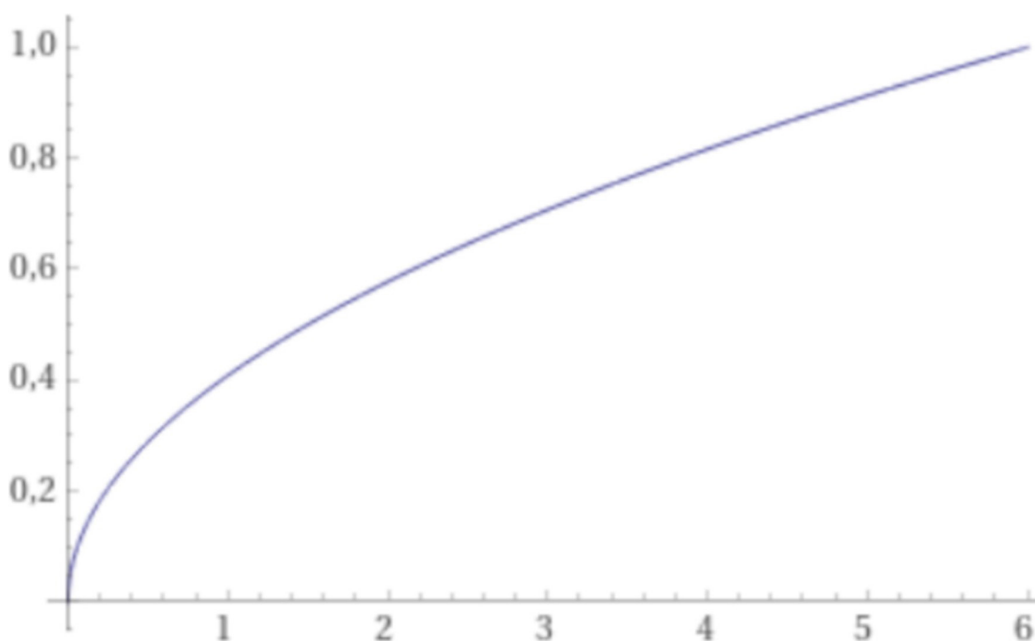


Fig. 3: Factor de humedad del suelo basado en los días de deuda de agua

Se adaptó el método de Rondinelli [12] y se transformó la función exponencial decreciente en el tiempo a su representación cuadrática normalizada para simplificar los cálculos y obtener el factor de humedad superficial (FHS) como la raíz cuadrada del cociente entre el número de días desde la última lluvia registrada (ulr) y el número de días para que el suelo se seque con la ecuación (4) cuyo resultado se muestra en la Fig. 3.

El número de días desde la última lluvia registrada (ulr) se obtuvo consultando la API de VisualCrossing [14],

pasando como argumento las coordenadas de los centroides de cada municipio.

La precipitación registrada en los 7 días anteriores debe ser mayor a 5 mm. para que la cantidad de agua caída humedezca y sature la capa de hojarasca compuesta por combustibles finos y medianos. Si la precipitación registrada es menor a 5 mm. se considera el valor 0 [15], [16].

$$FHS_m = \sqrt{\frac{ulr_m}{ds}} \quad (4)$$

Donde:

m es el valor ordinal en la lista de municipios

2.5 Factor de la Temperatura (FT)

El aumento de la temperatura acelera las reacciones químicas involucradas en el proceso de combustión. Cuando la temperatura de los combustibles alcanza niveles críticos, se produce una mayor liberación de gases volátiles esenciales para la ignición.

$$FT_m = \left(0.152 * \left(e^{(0.05 * T_m)}\right)\right) + 0.344 \quad (5)$$

Donde: t es el valor la temperatura en grados Celcius m es el valor ordinal en la lista de municipios e es la constante de Euler

$$FPH_m = \frac{1}{\left(-0.33 * \left(e^{(-0.22 * (pa_m - 750))}\right) + 1\right)} \quad (6)$$

Donde: pa es el valor de la presión atmosférica en hPa m es el valor ordinal en la lista de municipios e es la constante de Euler

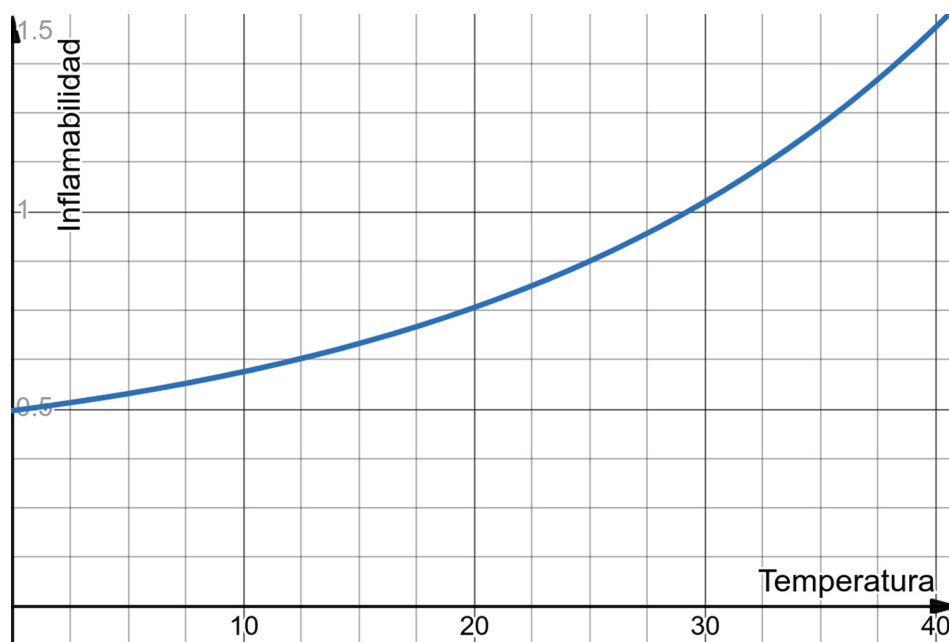


Fig. 4: Relación de la Temperatura con la Inflamabilidad

2.6 Factor de la Presión Atmosférica

Relación de la Presión Atmosférica y la Humedad del Combustible (FPH) De acuerdo a Hurtado [17], la disminución de la presión del aire aumenta el proceso de evapotranspiración. A medida que la presión atmosférica disminuye, como al pasar de 1020 hPa a 750 hPa, se reduce la capacidad del aire para retener vapor de agua, lo que disminuye la humedad disponible en la hojarasca.

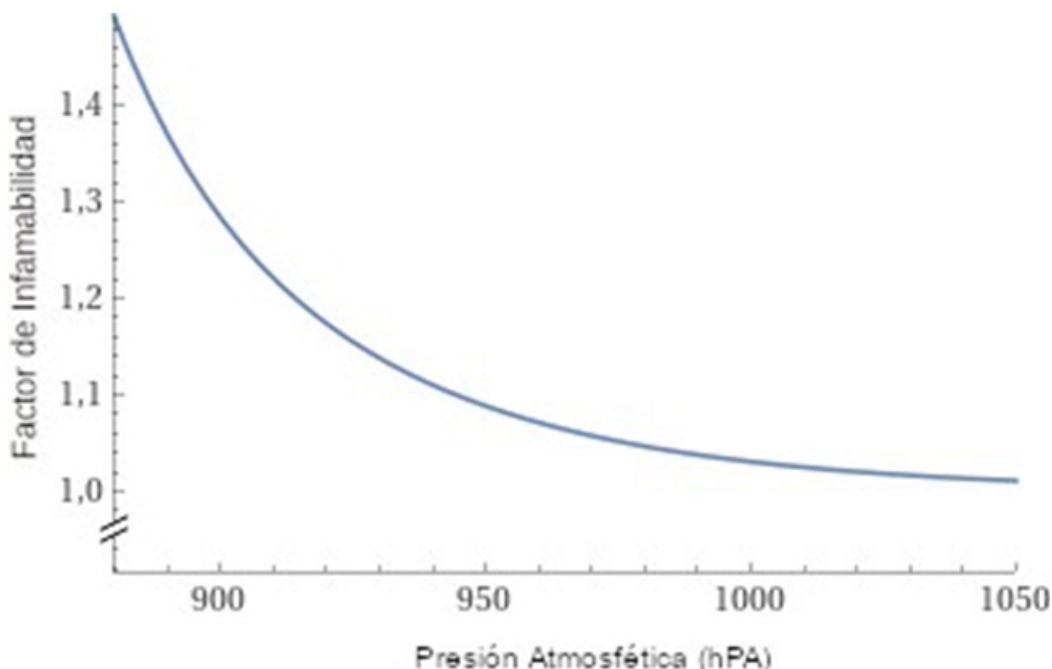


Fig. 5: Relación de la Presión y la Humedad del Combustible

Relación de la Presión Atmosférica y Concentración de Oxígeno en el Aire (FPO2) La presión del aire incide en la densidad molecular del oxígeno. A medida que aumenta la presión atmosférica, aumenta concentración de moléculas de oxígeno por unidad de volumen [18].

Se define la ecuación de corrección de la inflamabilidad por efecto de la presión sobre la densidad de oxígeno como:

$$FPO2_m = \left(0.2445 * e^{(-0.01 * (pa_m - 1010))} + 0.755 \right) \quad (7)$$

Donde:

pa es el valor de la presión atmosférica en hPa del municipio m , m es el valor ordinal de la lista de municipios
 e es la constante de Euler

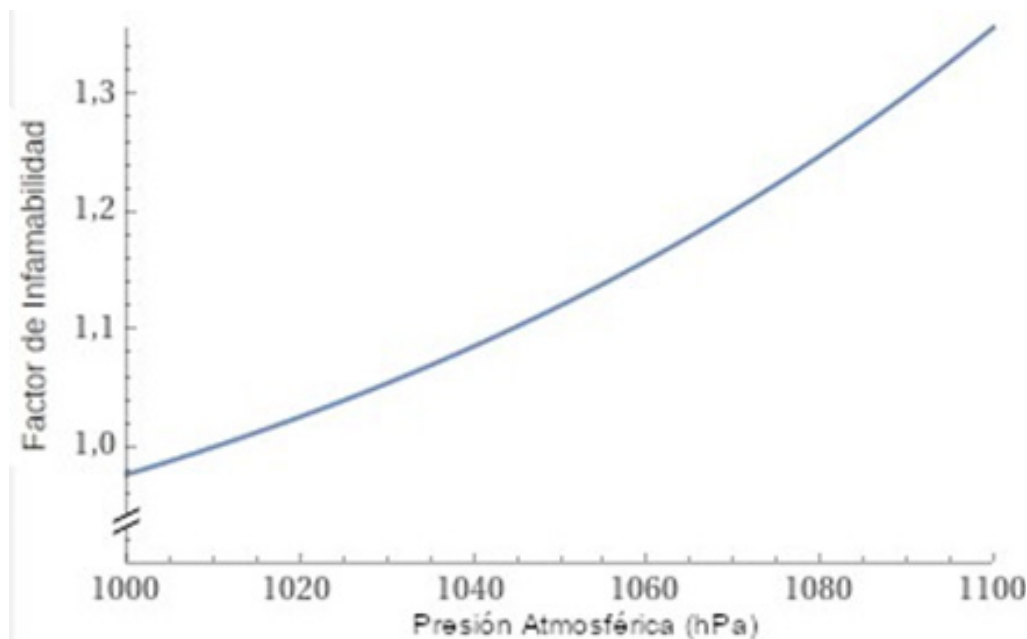


Fig. 6: Relación de la Presión y la Disponibilidad de Oxígeno

2.7 Factor de la velocidad del viento (FVV)

La velocidad del viento afecta directamente a la velocidad de propagación del fuego. Se incorporó al algoritmo la fórmula propuesta por Cruz et al. [19], donde sugieren que el factor de la velocidad del viento (FVV) se puede calcular como el 10% de la velocidad superficial del viento registrado.

$$FVV_m = (1 + (vv_m * 0.1)) \quad (8)$$

Donde:

vv es la velocidad superficial del viento en km/h
 m es el valor ordinal en la lista de municipios

3 Algoritmo del Índice de Riesgo de Incendio Rurales

Se basa en un coeficiente de inflamabilidad de los combustibles en el suelo de cada municipio, que depende de la velocidad de propagación del fuego y su proporción con el valor de la superficie, y se ajusta mediante un coeficiente de presión y humedad. El índice de inflamabilidad del municipio IIM_m es el resultado de la suma de la velocidad de propagación del combustible del suelo de cada tipo de cubierta ID_c y el porcentaje de la superficie de cada tipo de cubierta vegetal PS_c .

El valor del riesgo de incendio $IRIR$ (10) de cada municipio m es el índice de inflamabilidad del municipio IIM_m , ajustado por el factor normalizado de humedad del suelo FHS y el factor de velocidad del viento FVV en esta zona. Este valor representa la ignitabilidad y el riesgo de inicio de un incendio.

El porcentaje de la superficie del área PS_c de cada tipo de cobertura c del suelo, se multiplica por el índice de inflamabilidad diferencial ID_c para obtener el índice de inflamabilidad de cada municipio IIM_m .

$$IIM_m = \left(\sum ID_{(m,c)} * PS_{(m,c)} \right) * 10 \quad (9)$$

Donde:

m : (1 ··· 78) es el valor ordinal en la lista de municipios

IIM_m es el valor del índice de inflamabilidad de cada municipio m : (22) es la cantidad de tipos de cobertura del suelo

c : (1 ··· n) son los tipos de cobertura del suelo

$ID_{(m,c)}$ es el índice de inflamabilidad para cada tipo de cobertura en el municipio m

$PS_{(m,c)}$ es el porcentaje de la superficie de cada tipo de cobertura del suelo en el municipio m

El cálculo de IIM se realiza semestralmente y los resultados se publican en el Índice de Inflamabilidad por Municipio.

El valor de riesgo de incendio $IRIR$ se calcula como:

$$IRIR_m = IIM_m * FHS_m * FT_m * FPH_m * FPO2_m * FVV_m \quad (10)$$

Donde:

m es el valor ordinal en la lista de municipios

$IRIR_m$ es el índice de riesgo de incendio FHS_m es el factor de humedad del suelo FT_m es el factor de

FPH_m es el factor de

$FPO2_m$ es el factor de

FVV_m es el factor de velocidad del viento

3.1 Etiquetas Semánticas del Riesgo de Incendios Rurales

Se propone una escala de R1 a R5 para que los valores de riesgo de referencia se correspondan y sean consistentes con los valores utilizados en otros índices de riesgo más comunes como Sharples et al. [20], Cane et al. [21] o Lavene et al. [22]. Además, se asigna una etiqueta semántica basada en el análisis con lógica difusa: Bajo, Moderado, Alto, Muy Alto y Extremo correspondientes a las cinco categorías de riesgo de incendio (R1-R5) y asociadas al siguiente esquema de colores: Verde, Azul, Amarillo, Naranja y Rojo.

El clasificador asigna las etiquetas lingüísticas a los diferentes niveles de peligro de incendio en un formato comprensible para los ciudadanos, que se muestran en un mapa de colores categóricos y un mapa de calor.

3.2 Visualización del IRIR

En las Figs. 7 y 8, se presenta una visualización del mapa calor y el mapa de colores categóricos para los valores $IRIR$ difusos según la clasificación semántica. Esta representación utiliza una gama de colores para mostrar la variación del índice de riesgo de incendio rural en diferentes áreas geográficas. La tonalidad roja indica un mayor riesgo de incendio, mientras que la tonalidad verde indica un riesgo bajo. Esta visualización ayuda a los usuarios a comprender mejor el $IRIR$.

El modelo computacional del $IRIR$ está operativo en el sitio del: Índice de Riesgo de Incendios Rurales, y los detalles técnicos de la implementación están accesibles en el proyecto GitLab: Ponderación del Riesgo y Predicción del Comportamiento de Incendios Rurales

4 Conclusiones

El principal aporte de este trabajo es un algoritmo computacional para calcular el riesgo de incendios rurales al considerar la inflamabilidad del combustible en el suelo y los datos meteorológicos en tiempo real.

El $IRIR$ se destaca por su enfoque dinámico y su capacidad para utilizar datos en línea sin la necesidad de intervención humana ni preprocesamiento de datos. Incorpora técnicas de aprendizaje automático para realizar una clasificación del uso del suelo a partir de imágenes satelitales. Mejora la interpretación de los resultados asignando etiquetas lingüísticas a los rangos de valores, lo que facilita una comprensión más intuitiva de los riesgos de incendio en las regiones analizadas. En resumen, el $IRIR$ es una herramienta innovadora para la evaluación y predicción del riesgo de incendios rurales.

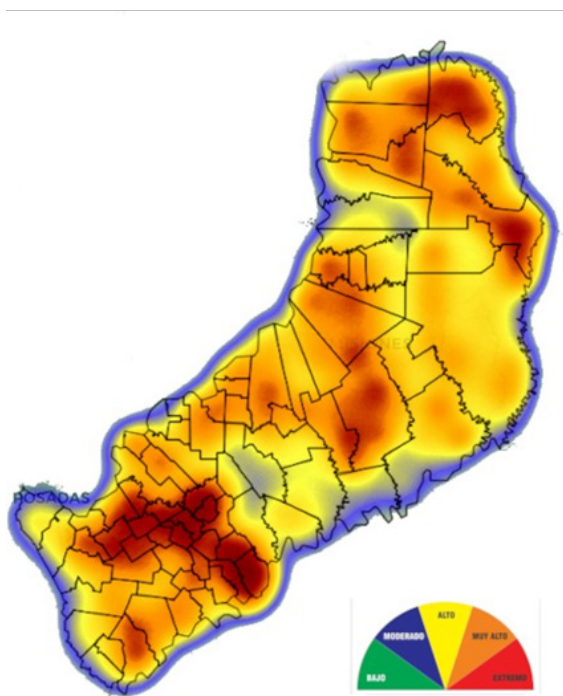


Fig. 7: Mapa de calor

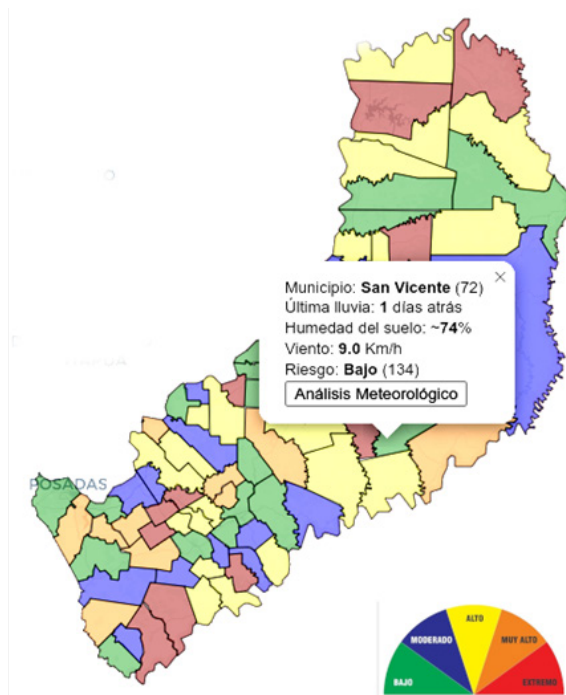


Fig. 8: Mapa de colores categóricos

Referencias

1. Arnaldos Viger, J., Castelló Vidal, J. & Giménez Pujol, A. Manual de ingeniería básica para la prevención y extinción de incendios forestales. (Ediciones MundiPrensa,2004)
2. CIFSR, C. Glossary of Fire Weather Terms. (2023), <https://www.cifrc.ca/publications/glossary>, Canadian Institute for Fire Safety Research
3. Rothermel, R. A mathematical model for predicting fire spread in wildland fuels. Res. Pap. INT-115. Ogden, UT: U.S. Department Of Agriculture, Intermountain Forest And Range Experiment Station. 40 P.. 115 (1972), <https://srs.fs.usda.gov/pubs/32533>
4. Gupta, A., Watson, S. & Yin, H. Deep learning-based aerial image segmentation with open data for disaster impact assessment. Neurocomputing. 439 pp. 22-33 (2021), <https://www.sciencedirect.com/science/article/pii/S0925231221001429>
5. Qin, R. & Liu, T. A Review of Landcover Classification with Very-High Resolution Remotely Sensed Optical Images—Analysis Unit, Model Scalability and Transferability. Remote Sensing. 14, 646 (2022), <https://www.mdpi.com/20724292/14/3/646>, Number: 3 Publisher: Multidisciplinary Digital Publishing Institute
6. Scott, J. Fuel models for fire behavior prediction. Forest Service General Technical Report RM-119. pp. 30 (1985)
7. Anderson, H. Aids to Determining Fuel Models For Estimating Fire Behavior. (United States Department of Agriculture, Forest Service, Pacific Southwest Research Station,1982)
8. Brys, C., Navas-Delgado, I. & Aldana-Montes, J. Wildfire Risk Weighting and Behavior Prediction Using Open Geospatial Data and Ontologies. Journal Of Information Science. (2023), <https://doi.org/10.1177/01655515231202757>, Publisher: SAGE Publications Ltd
9. DMGGE, Dirección de Modernización de la Gestión y Gobierno Electrónico. (2023) Ponderación del Riesgo y Predicción del Comportamiento de Incendios Rurales (ER-01). (Dirección de Modernización de la Gestión y Gobierno Electrónico de Misiones,2023,10,12), <https://gitlab.com/Modernizacion>.

Misiones/incendiosrurales/-/wikis/informe-experimento-01

10. Rossa, C. & Fernandes, P. Empirical Modeling of Fire Spread Rate in No-Wind and No-Slope Conditions. *Forest Science*. 64, 358-370 (2018), <https://doi.org/10.1093/forsci/fxy002>
11. INTA, Estación Experimental “INTA Cerro Azul”. (2023), <https://inta.gob.ar/cerroazul>, Accesses [2023-01-10]
12. Rondinelli, W., Hornbuckle, B., Patton, J., Cosh, M., Walker, V., Carr, B. & Logsdon, S. Different Rates of Soil Drying after Rainfall Are Observed by the SMOS Satellite and the South Fork in situ Soil Moisture Network. *Journal Of Hydrometeorology*. 16,889-903 (2015), https://journals.ametsoc.org/view/journals/hydr/16/2/jhm-d-14-0137_1.xml, Publisher: American Meteorological Society Section: Journal of Hydrometeorology
13. Kurc, S. & Small, E. Dynamics of evapotranspiration in semiarid grassland and shrubland ecosystems during the summer monsoon season, central New Mexico. *Water Resources Research*. 40 (2004), <https://onlinelibrary.wiley.com/doi/abs/10.1029/2004WR003068>
14. Visual Crossing Corporation. Weather Data & API. (2023), <https://www.visualcrossing.com/>, Accesses [2023-01-10]
15. Kunst, C. R., Ledesma, R. R., Bravo, S., Defossé, G. E., Godoy, J. A., and Navarrete, V. D. C. (2014). Dinámica de la humedad del suelo en un área bajo riego utilizando técnicas geoestadísticas. *RIA*, 40(2), 165-181. <https://acortar.link/xNV42W>
16. Lawson, B. D., and Armitage, O. B. (2008). Weather Guide for the Canadian Forest Fire Danger Rating System. Natural Resources Canada, Canadian Forest Service, Northern Forestry Centre.
17. Hurtado, R. & Specha, L. VII.3 Evaporación y evapotranspiración. *Agrometeorología*. pp. 85-102 (2011)
18. Yang, J., Liu, N., Chen, H., Gao, W. & Tu, R. Effects of atmospheric oxygen on horizontal peat smoldering fires: Experimental and numerical study. *Proceedings Of The Combustion Institute*. (2019)
19. Cruz, M. & Alexander, M. The 10% wind speed rule of thumb for estimating a wildfire’s forward rate of spread in forests and shrublands. *Annals Of Forest Science*. 76, 44 (2019,4,25), <https://doi.org/10.1007/s13595-019-0829-8>
20. Sharples, J., McRae, R., Weber, R. & Gill, A. A simple index for assessing fire danger rating. *Environmental Modelling & Software*. 24, 764-774 (2009), <https://www.sciencedirect.com/science/article/pii/S1364815208002119>
21. Cane, D., Ciccarelli, N. & Pelosini, R. Fire Weather Index Implementation in the Alpine Area. *ISRN Meteorology*. 2012 pp. 429072 (2011), <https://doi.org/10.5402/2012/429072>
22. Laneve, G., Pampanoni, V. & Uddien Shaik, R. The Daily Fire Hazard Index: A Fire Danger Rating Method for Mediterranean Areas. *Remote Sensing*. 12, 2356 (2020), <https://www.mdpi.com/2072-4292/12/15/2356>

NaVi-Q como soporte a la evaluación de visualizaciones narrativas: un estudio de caso

Andrea Lezcano Airaldi

Universidad Nacional del Nordeste, Corrientes, Argentina.
alezcano@exa.unne.edu.ar

Melisa Lezcano Airaldi

Universidad Nacional del Nordeste, Corrientes, Argentina.
melisalezcanoairaldi@gmail.com

Emanuel Irrazábal

Universidad Nacional del Nordeste, Corrientes, Argentina.
eirrazabal@exa.unne.edu.ar

Resumen

Este artículo presenta un estudio de caso sobre la aplicación del modelo NaVi-Q en un proyecto real de visualización para la gestión de deuda técnica. El objetivo fue analizar su integración en el flujo de trabajo de un equipo de desarrollo y el impacto de sus recomendaciones en la calidad de las visualizaciones. El estudio se desarrolló en tres fases: evaluación inicial, implementación de mejoras y análisis comparativo entre versiones originales y rediseñadas. Se emplearon cuestionarios, escalas estandarizadas y encuestas cualitativas para evaluar comprensión, usabilidad y percepción del equipo. Los resultados muestran mejoras significativas en tareas específicas y en la escala SUS, además de una valoración positiva por parte del equipo. Estos hallazgos respaldan la utilidad del modelo como herramienta de evaluación iterativa en escenarios reales de diseño, y abren líneas futuras para su integración en flujos de trabajo y su ampliación a otros dominios.

1. Introducción

Las visualizaciones narrativas constituyen una forma de comunicación de datos que integra recursos gráficos y discursivos para transmitir información de manera contextualizada [1,2]. Su aplicación se ha extendido a diversos dominios, entre ellos el periodismo de datos [3], la gestión organizacional y de negocios [4,5], y el desarrollo de software [6], donde facilitan la interpretación de información compleja y el seguimiento de procesos. En contextos de desarrollo de software y análisis organizacional, este tipo de visualizaciones permite integrar información técnica con interpretaciones comprensibles y orientadas a la acción [7].

No obstante, evaluar la calidad de las visualizaciones narrativas continúa siendo un problema abierto[8]. La literatura identifica tres dificultades principales: (i) la falta de procedimientos y criterios estandarizados que permitan comparaciones reproducibles entre estudios [9,10], (ii) limitaciones de validez externa en los experimentos controlados, donde las tareas suelen ser artificiales y los contextos limitados [11,12], y (iii) una cobertura parcial de dimensiones narrativas y de experiencia de usuario, aún escasamente abordadas en las evaluaciones de visualización [13,14].

En este contexto, NaVi-Q [15] propone un modelo jerárquico de calidad desarrollado para abordar estas limitaciones mediante un enfoque estructurado, jerárquico y adaptable. Organiza los criterios de calidad en tres niveles —criterios, propiedades y aplicaciones— y proporciona una herramienta que guía la evaluación mediante un cuestionario, permite definir perfiles personalizados y genera recomendaciones de mejora. Su diseño se fundamenta en principios normativos de calidad del software, y ha sido validado previamente en condiciones controladas, mostrando alineación con juicios expertos y sensibilidad para distinguir niveles de calidad. El presente artículo aporta un estudio de caso aplicado, orientado a observar el funcionamiento de NaVi-Q en un entorno profesional donde las visualizaciones son utilizadas como parte de un proceso de análisis real. El artículo se organiza de la siguiente manera: en la Sección 2 se revisan los trabajos relacionados, con foco en los métodos de evaluación de visualizaciones y en los fundamentos del modelo. La Sección 3 describe la metodología del estudio de caso. La Sección 4 presenta los resultados y la Sección 5 expone las conclusiones y posibles líneas de trabajo futuro.

2. Trabajos relacionados

Esta sección revisa los principales enfoques de evaluación de visualizaciones y presenta el modelo NaVi-Q como marco estructurado para este propósito.

2.1. Evaluación de visualizaciones

La evaluación de visualizaciones es un campo de investigación consolidado, aunque heterogéneo en cuanto a métodos y enfoques. Una clasificación ampliamente citada distingue entre métodos cuantitativos y cualitativos [9]. Los primeros incluyen experimentos controlados que miden precisión, tiempos de respuesta, carga cognitiva o memorabilidad. Estos estudios, aunque rigurosos, suelen presentar limitaciones de validez externa al depender de tareas artificiales y muestras acotadas [11].

Los enfoques cualitativos, por su parte, comprenden entrevistas, estudios de caso y revisiones heurísticas. Las heurísticas propuestas por [16] constituyen un antecedente relevante, aunque su aplicación depende fuertemente del juicio experto y no contempla dimensiones discursivas ni emocionales.

Errey et al. [8] realizaron un estudio de campo con practicantes de visualización narrativa que revela un panorama dominado por evaluaciones informales. Según sus hallazgos, la práctica habitual se basa en la experiencia acumulada de los diseñadores, el juicio subjetivo y la opinión de colegas. Métodos más estructurados —como la revisión heurística, pruebas de usuario o análisis de comportamiento— son poco frecuentes, debido a restricciones de tiempo, recursos o desconocimiento de herramientas. Este hallazgo confirma lo anticipado por estudios previos sobre la distancia entre métodos académicos y evaluación en entornos reales [17,18].

En el caso particular de las visualizaciones narrativas, la evaluación implica desafíos adicionales: se requiere considerar no solo la exactitud perceptual, sino también la claridad de la narrativa, la estructuración discursiva y el tono emocional. Investigaciones recientes han resaltado dimensiones como la memorabilidad [19] y la confianza percibida [20] como factores clave. Otros trabajos han explorado enfoques fenomenológicos [21], el diseño centrado en el usuario [22], o la experiencia emocional como parte del proceso evaluativo [14], aunque estos aportes aún no han sido integrados en marcos generalizables.

En conjunto, los estudios indican que la evaluación de visualizaciones (y especialmente las narrativas) requiere modelos que combinen precisión analítica, sensibilidad a la experiencia del usuario y viabilidad práctica en contextos reales.

2.2. El modelo NaVi-Q

El modelo NaVi-Q es un modelo jerárquico para evaluar visualizaciones narrativas desde una perspectiva centrada en la calidad. A diferencia de los enfoques tradicionales basados en heurísticas generales o pruebas aisladas de usabilidad, NaVi-Q estructura la evaluación en tres niveles: criterios, propiedades y aplicaciones específicas. Esta organización permite operacionalizar aspectos discursivos y perceptuales de las visualizaciones mediante un cuestionario guiado y replicable.

Los cinco criterios principales —comprensión, compromiso, información, memorabilidad y usabilidad— derivan de adaptaciones de normas como ISO/IEC 25010 [23] e ISO/IEC 33003 [24] al contexto de la visualización. Cada criterio se descompone en propiedades evaluables, vinculadas a un conjunto de aplicaciones concretas que guían el diagnóstico y la formulación de recomendaciones.

El modelo ha sido implementado en una herramienta web que facilita su uso práctico. Esta herramienta permite configurar perfiles de calidad personalizados según los objetivos del proyecto, registrar evaluaciones manuales o asistidas, y generar recomendaciones de mejora con orientación accionable. Estas recomendaciones se presentan de forma estructurada, alineadas con las deficiencias detectadas, y con lenguaje comprensible para equipos de diseño o desarrollo.

Validaciones previas en contextos controlados mostraron que el modelo es capaz de discriminar entre visualizaciones de diferente calidad y que sus resultados se correlacionan con juicios realizados por expertos y no expertos [15]. Además, se ha demostrado su viabilidad para evaluar visualizaciones de distintos dominios, mediante perfiles adaptados a cada caso. La presente contribución extiende esta validación a un entorno de uso real, con visualizaciones diseñadas, evaluadas y utilizadas dentro de un proyecto de software.

3. Metodología

Este estudio adopta un enfoque de estudio de caso, según la metodología propuesta por Yin [25], con el objetivo de caracterizar el uso de NaVi-Q en un entorno real y analizar su impacto en el proceso de diseño y en la calidad de las visualizaciones generadas. El estudio se desarrolló en el marco de un proyecto sobre gestión de deuda técnica, llevado adelante en colaboración con un grupo de investigación externo. En este contexto, se diseñaron visualizaciones interactivas destinadas a apoyar la toma de decisiones técnicas. Las mismas abordaban temas como la priorización de tareas, la detección de oportunidades de mejora y el seguimiento de decisiones de mantenimiento.

Para ello, se formularon las siguientes preguntas de investigación:

- PI-1: ¿Qué grado de utilidad tienen las recomendaciones de mejora proporcionadas por NaVi-Q para el equipo de trabajo?
- PI-1.a: ¿Qué errores o aspectos a mejorar son identificados con mayor frecuencia?
- PI-2: ¿Cuál es el impacto de aplicar las mejoras sugeridas por el modelo sobre la comprensión, la información y la usabilidad de las visualizaciones?
- PI-3: ¿Cuál es la percepción general sobre el uso del modelo?

3.1. Procedimiento

El estudio se estructuró en tres fases consecutivas:

1. Evaluación inicial. Un integrante del equipo de investigación aplicó la herramienta NaVi-Q para evaluar las visualizaciones desarrolladas. La evaluación se realizó de forma manual, utilizando un perfil de calidad personalizado configurado previamente a partir de formularios donde se seleccionaron criterios y propiedades relevantes.
2. Análisis de recomendaciones. A partir de los resultados obtenidos, el sistema generó recomendaciones automáticas de mejora, que fueron analizadas por el equipo. Para cada sugerencia, se registró su utilidad percibida, el conocimiento previo sobre la práctica asociada y la intención de aplicarla en el rediseño.
3. Implementación y evaluación comparativa. Se desarrollaron nuevas versiones de las visualizaciones incorporando las recomendaciones seleccionadas. Para analizar su impacto, se diseñó una comparación entre dos grupos: uno control (visualizaciones originales) y uno experimental (versiones rediseñadas). Ambos grupos realizaron tareas específicas y completaron instrumentos que midieron comprensión, información y usabilidad percibida.

3.2. Instrumentos de recolección de datos

La recolección de datos se organizó en función de las preguntas de investigación, combinando métodos cuantitativos y cualitativos.

Para las preguntas -1 y PI-1.a, una vez finalizadas las evaluaciones y generadas las sugerencias de mejora, se consolidó una lista única de recomendaciones emitidas por el sistema. Cada recomendación se presentó al equipo junto con su respectiva propuesta de implementación, generada mediante asistencia automatizada. Así, se distinguió entre la sugerencia general (derivada de una aplicación del modelo con baja puntuación) y la acción específica sugerida para mejorar la visualización. Los miembros del equipo calificaron cada recomendación como “útil”, “parcialmente útil” o “no útil”, indicaron si conocían previamente la buena práctica asociada, y señalaron si planeaban implementarla en el rediseño. A partir de esta lista se realizó una clasificación cualitativa de los aspectos de mejora detectados, que permitió identificar patrones comunes y agrupar las recomendaciones según el tipo de problema.

Para responder a la PI-2, se llevó a cabo una evaluación comparativa orientada a analizar el impacto de las mejoras sugeridas por el modelo sobre la comprensión, la información y la usabilidad de las visualizaciones. Estos tres criterios concentraron el 79 % del peso total en el perfil de calidad utilizado para el estudio. La evaluación se realizó con dos grupos de participantes ($N=24$), conformados por estudiantes avanzados de informática, asignados aleatoriamente a cada grupo. El grupo control utilizó las visualizaciones originales del proyecto, mientras que el grupo experimental trabajó con las versiones mejoradas tras la aplicación de las recomendaciones seleccionadas. Ambos grupos completaron las tareas de forma individual, sin asistencia externa y en condiciones equivalentes. Se aplicaron dos instrumentos principales:

- **Cuestionario de comprensión e información.** Compuesto por cinco preguntas de opción múltiple con una única respuesta correcta, diseñadas para evaluar tareas concretas de lectura, identificación y comparación. Las preguntas se adaptaron del Visualization Literacy Assessment Test (VLAT) [26]. Para cada pregunta, los participantes calificaron su nivel de confianza en la respuesta mediante una escala de cinco puntos (1 = nada confiado, 5 = completamente confiado). La Tabla 1 resume las preguntas incluidas.
- **Escala de Usabilidad (SUS) adaptada.** Diseñada para estimar la usabilidad percibida del conjunto de visualizaciones, administrada tras la interacción de cada participante con el tablero. La escala mantuvo la estructura original de diez ítems propuesta por Brooke [27], con formulaciones afirmativas y negativas, y fue adaptada al contexto del estudio. Cada ítem se respondió mediante una escala Likert de cinco puntos, y el puntaje final se calculó siguiendo el procedimiento estándar del autor. La Tabla 2 presenta el conjunto de ítems utilizados.

Finalmente, para la PI-3, una vez completadas las evaluaciones, los miembros del equipo de investigación respondieron una breve encuesta cualitativa sobre la percepción del uso de NaVi-Q en contextos de trabajo reales. La encuesta consistió en tres preguntas abiertas: (1) etapa del proceso en la que aplicarían el modelo; (2) tareas o decisiones en las que consideran más útil su aplicación; y (3) obstáculos que podrían dificultar su adopción. Las respuestas se analizaron mediante codificación temática para identificar patrones recurrentes.

N.º	Enunciado de la pregunta	Tipo de tarea
1	¿En qué mes se registró el mayor esfuerzo para cerrar tiques de DT?	Encontrar un extremo
2	Según el gráfico de "Oportunidades fáciles de aprovechar", ¿qué combinación de prioridad y esfuerzo caracteriza a estos ítems?	Caracterizar la distribución
3	¿Cuál es la tendencia general del esfuerzo acumulado en los tiques de DT abiertos en los últimos 12 meses?	Encontrar correlaciones o tendencias
4	¿Cuál es el nivel de discrepancia más alto entre prioridad dada y calculada en los tiques de DT?	Encontrar un extremo
5	¿Qué visualización permite detectar qué tiques tienen un retorno de inversión más temprano?	Recuperar un valor

Tabla 1. Preguntas del cuestionario de comprensión e información.

N.º	Enunciado del ítem	Formulación
1	Me parece que usar el tablero fue sencillo.	Afirmativa
2	Considero que necesitaría ayuda técnica para poder usar este tablero.	Invertida
3	Creo que las diferentes partes del tablero están bien integradas.	Afirmativa
4	Me pareció que había demasiada complejidad en el uso del tablero.	Invertida
5	Me sentí seguro/a usando el tablero.	Afirmativa
6	Considero que las visualizaciones eran consistentes entre sí.	Afirmativa
7	Me gustaría usar un sistema como este con frecuencia.	Afirmativa
8	Encontré que las visualizaciones eran confusas.	Invertida
9	Pude aprender a usar el sistema rápidamente.	Afirmativa
10	Necesitaría aprender muchas cosas nuevas antes de poder usar el tablero.	Invertida

Tabla 2. Ítems de la escala de Usabilidad (SUS) adaptada.

4. Resultados y Discusión

Esta sección presenta los resultados obtenidos a partir de la aplicación del modelo. El análisis se estructura en tres partes de acuerdo con las preguntas de investigación: la evaluación de la utilidad percibida y los problemas detectados (PI), el impacto de las mejoras aplicadas (PI-2) y la percepción del equipo sobre el uso del modelo (PI-3).

4.1. Utilidad percibida y patrones de problemas detectados

En total se identificaron 41 recomendaciones asociadas a distintas aplicaciones del modelo. Tal como se describió en la Sección 3, cada recomendación fue evaluada por los miembros del equipo en términos de utilidad percibida, conocimiento previo y aplicabilidad. La Tabla 3 muestra la cantidad de veces que cada recomendación fue generada por el modelo (frecuencia), así como el número de ocasiones en que fue considerada útil, conocida previamente o aplicada, según la evaluación realizada por los miembros del equipo.

Recomendación	Frecuencia	Útil	Conocida	Aplicada
Atribución de fuentes	7	1	7	0
Elementos asociados	7	7	2	5
Línea base en cero	6	0	6	0
Ilustraciones	7	2	0	0
Etiquetas de serie	4	4	4	4
Cambios en los datos	4	4	0	4
Mensajes y resúmenes	2	2	0	1
Títulos de ejes	1	1	0	1
Marcas y etiquetas de escala	1	1	0	0
Estructura y flujo	1	1	0	0
Resaltado visual	1	1	0	1

Tabla 3. Frecuencia, utilidad, conocimiento y aplicación de las recomendaciones.

Como resultado del proceso de clasificación cualitativa mencionado anteriormente, las recomendaciones fueron organizadas en cuatro categorías, según el tipo de problema que buscaban abordar. La Tabla 4 presenta estas categorías junto con los aspectos de mejora incluidos en cada una y una breve justificación de su agrupación.

Problema detectado	Recomendaciones incluidas	Justificación
Narrativa visual débil	Ilustraciones, Cambios en los datos, Estructura y flujo	No se representan bien los cambios, o falta apoyo visual que guíe la lectura narrativa del gráfico.
Codificación ambigua	Elementos asociados, Etiquetas de serie, Resaltado visual	Elementos visuales mal diferenciados o poco legibles; dificultan la interpretación correcta.
Falta de contexto	Atribución de fuentes, Mensajes y resúmenes	Faltan elementos para entender de qué trata la visualización, cómo leerla o qué quiere decir.
Escala y ejes deficientes	Base en cero, Títulos de ejes, Marcas y etiquetas	Problemas en los ejes, escalas o etiquetas que afectan la precisión y comparabilidad.

Tabla 4. Agrupación de recomendaciones según el tipo de problema.

Esta clasificación permitió observar diferencias en la utilidad, el conocimiento previo y la aplicación de las sugerencias según el tipo de problema. La Figura 1 resume esta información.

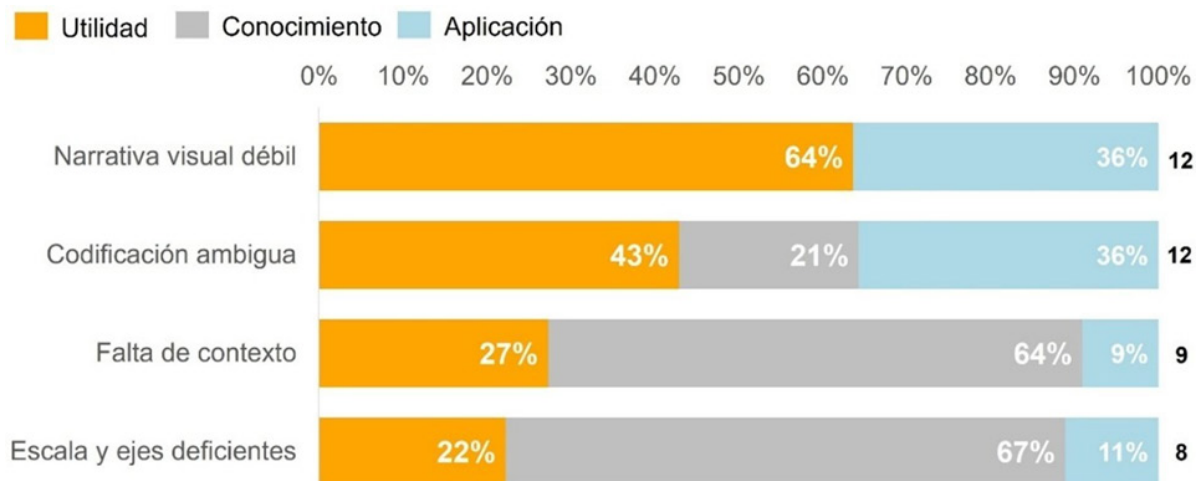


Figura 1. Utilidad, conocimiento y aplicación de las recomendaciones según tipo de problema identificado. Los valores absolutos indican la frecuencia de cada error.

Se observa que la categoría *Narrativa visual débil* concentra el mayor porcentaje de sugerencias consideradas útiles (64 %) y aplicadas (36 %), a pesar de que ninguna había sido identificada previamente por el equipo (0 % de conocimiento). Este patrón sugiere que el modelo permitió identificar aspectos que no habían sido considerados previamente por el equipo. Las recomendaciones aportaron contenido nuevo y suficientemente concreto como para ser implementado en el rediseño. En contraste, las categorías *Falta de contexto* y *Escala y ejes deficientes* presentan altos niveles de conocimiento previo (64 % y 67 %, respectivamente), pero baja aplicación (9 % y 11 %). Esto sugiere que, si bien el equipo estaba familiarizado con las buenas prácticas asociadas a estos aspectos, dichas recomendaciones no fueron priorizadas o resultaron difíciles de incorporar en el rediseño.

Un ejemplo de recomendación considerada útil y aplicada es el caso de *Elementos asociados*. La sugerencia generada por el sistema proponía:

“Utilice esquemas de color coincidentes para las categorías de tickets abiertos y cerrados en ambos gráficos para crear asociaciones visuales, asegurándose de que se utilice el mismo color para los tickets ‘abiertos’ y ‘cerrados’ en los gráficos de esfuerzo y de puntos de historia para mantener la coherencia.”

Esta mejora fue implementada en el rediseño para facilitar la interpretación visual del contenido. En cambio, la recomendación referida a *Atribución de fuentes* fue reconocida por todos los integrantes del equipo, pero no se implementó, al considerarse de bajo impacto para los objetivos del tablero y poco viable en el formato final.

Estos resultados indican que la utilidad percibida no se vincula necesariamente al conocimiento previo, y que la aplicabilidad está condicionada por la claridad de las sugerencias y el contexto de uso.

4.2. Impacto de las mejoras aplicadas

Comprensión e información. Para evaluar el impacto de las mejoras aplicadas, los participantes completaron el cuestionario presentado en la Sección 3. Cada pregunta abordó una tarea específica asociada a la comprensión de la información visual, e incluyó una escala de confianza vinculada a la respuesta. El análisis se centró en comparar los resultados obtenidos por grupo, considerando tanto la precisión como la confianza reportada.

El grupo control utilizó visualizaciones del proyecto original, mientras que el grupo experimental trabajó con versiones rediseñadas que incorporaban recomendaciones seleccionadas por el equipo. En línea con los resultados de la PI-1, la mayoría de las mejoras aplicadas no implicaron cambios estructurales, sino ajustes orientados a reforzar la narrativa o facilitar la interpretación. Estos ajustes incluyeron, por ejemplo, *tooltips* contextuales, resúmenes o elementos de ayuda integrados, aprovechando las características interactivas del tablero.

La Figura 2 ilustra un ejemplo representativo de este proceso de mejora, mostrando la versión original de una visualización (panel izquierdo) junto con su versión rediseñada (panel derecho). En la segunda versión se incorporaron etiquetas descriptivas y resúmenes visuales que facilitan la interpretación y el seguimiento de los datos, sin alterar la estructura general del gráfico.

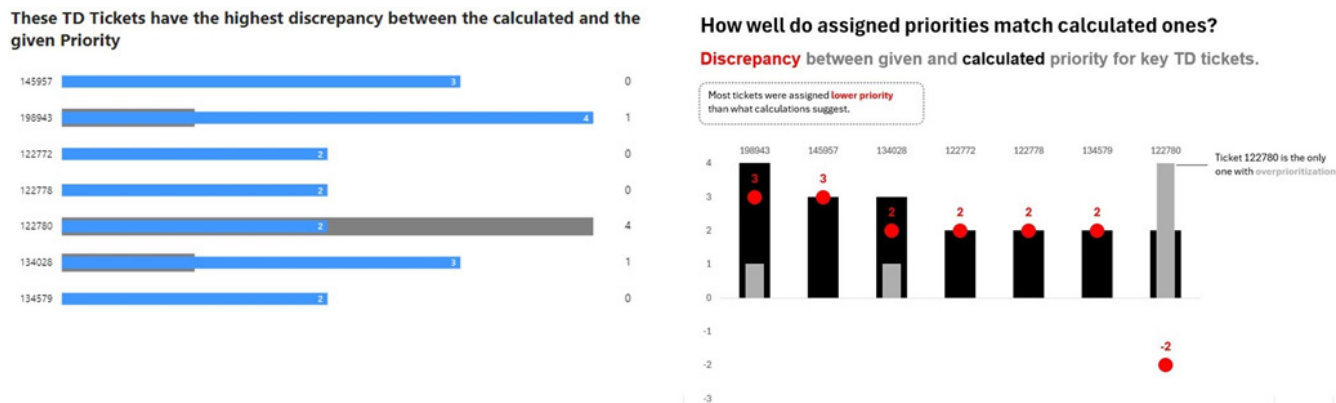


Figura 2. Ejemplo de visualización original (panel izquierdo) y rediseñada (panel derecho).

Los resultados se resumen en la Figura 3. El gráfico presenta, para cada pregunta del cuestionario (P1 a P5), la proporción de respuestas correctas (barras) y el nivel promedio de confianza (puntos) por grupo. La confianza fue normalizada a una escala de 0 a 1 para facilitar la comparación.

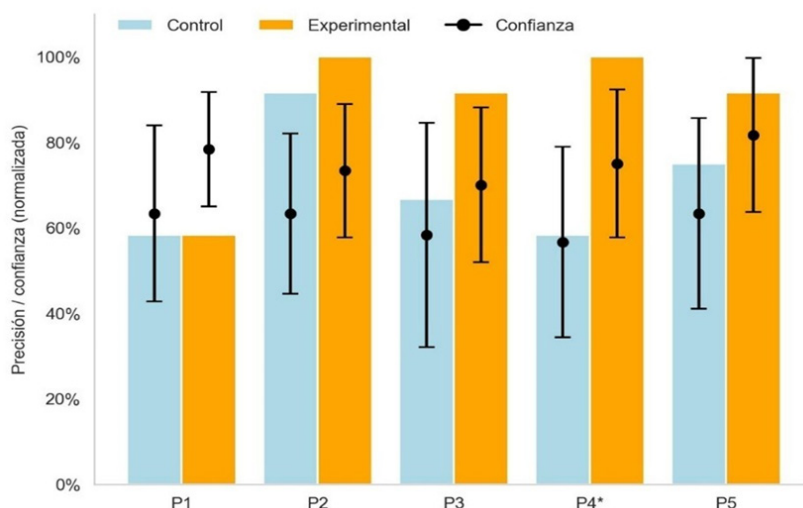


Figura3. Precisión y confianza normalizada por pregunta y grupo.

En cuatro de las cinco preguntas, el grupo experimental alcanzó mayores niveles de precisión. La diferencia más destacada se registró en P4, relacionada con encontrar valores extremos. En ese caso, el 100 % de los participantes del grupo experimental respondió correctamente, frente al 58 % del grupo control. En la mayoría de los casos, los niveles de confianza fueron consistentes con la precisión observada.

No se detectaron casos en los que un grupo manifestara mayor confianza sin haber obtenido mejores resultados.

Para analizar diferencias entre grupos, se aplicaron pruebas estadísticas no paramétricas, dado que las distribuciones no cumplían con los supuestos de normalidad. Se utilizó la prueba U de Mann-Whitney para comparar la cantidad total de respuestas correctas y el nivel de confianza promedio por participante. La proporción de respuestas correctas por pregunta se contrastó mediante la prueba exacta de Fisher [?] dada la naturaleza categórica de los datos y el tamaño reducido de la muestra.

Los resultados estadísticos fueron consistentes con los patrones descriptivos. El grupo experimental reportó niveles de confianza significativamente superiores ($U=20,5, p = ,003$). Si bien no se observaron diferencias significativas en la cantidad total de respuestas correctas ($U=41,5, p = ,068$), se identificó una tendencia favorable. A nivel individual, solo la pregunta P4 mostró una diferencia estadísticamente significativa entre grupos ($p = ,037$), denotada en el gráfico con un asterisco (*). Las demás preguntas no presentaron variaciones significativas.

Estos resultados indican que las mejoras implementadas tuvieron un efecto positivo en la comprensión de

tareas específicas, en particular aquellas vinculadas a la identificación de valores extremos. Esto respalda la utilidad del modelo como herramienta para identificar aspectos del diseño que inciden en la comprensión de los usuarios finales.

Usabilidad. La escala SUS permitió explorar cómo fue percibida la usabilidad del tablero por parte de los participantes luego de interactuar con las visualizaciones. Los puntajes obtenidos reflejan la evaluación subjetiva del sistema considerando aspectos como facilidad de uso, integración, claridad de los elementos y necesidad de apoyo externo.

La Tabla 5 muestra el puntaje total obtenido por cada grupo, incluyendo medidas de tendencia central y dispersión. El grupo experimental alcanzó un puntaje promedio significativamente mayor ($M = 83,5, DE = 3,9$) en comparación con el grupo control ($M = 63,1, DE = 6,2$). Esta diferencia, confirmada mediante la prueba U de Mann-Whitney ($U = 0,0, p < ,001$), sugiere una percepción más positiva de usabilidad por parte de quienes utilizaron las visualizaciones rediseñadas.

Grupo	Media	DE	Min.	Max.	Mediana
Control	63.12	6.22	52.5	70	65
Experimental	83.54	3.91	77.5	90	82.5

Tabla 5. Media, mediana, rango y desviación estándar del SUS por grupo.

Al clasificar los puntajes según los umbrales propuestos por Bangor et al. [?], se observa que diez participantes se ubicaron en el rango “Bueno”, diez en “Aceptable” y cuatro en “Excelente”. Ningún caso se ubicó en la categoría “Pobre”. Esta distribución indica una percepción favorable del sistema, con mayor variabilidad dentro del grupo control.

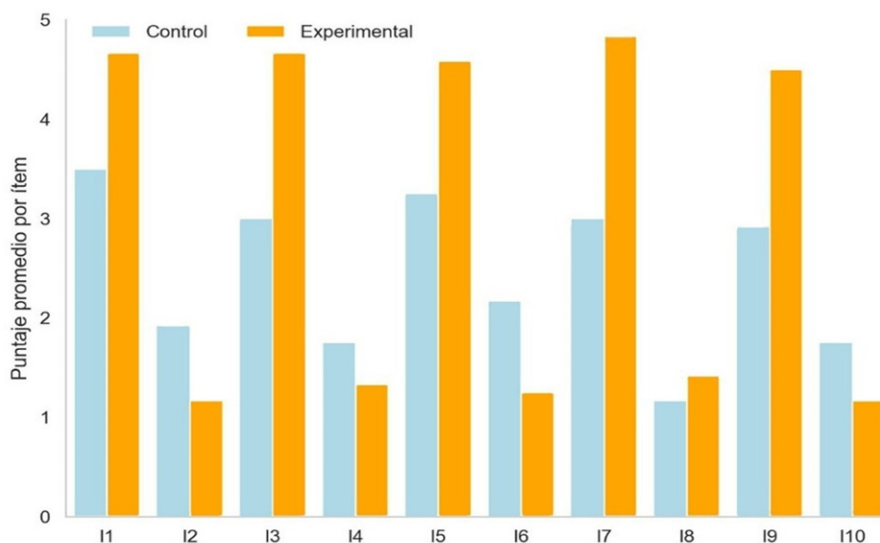


Figura4. Puntaje promedio por ítem y grupo en la escala SUS.

La Figura 4 presenta el puntaje promedio por ítem (I) del cuestionario SUS, discriminado por grupo. Cada ítem fue calificado en una escala de 1 (muy en desacuerdo) a 5 (muy de acuerdo). El grupo experimental reportó mayores niveles de acuerdo en todos los ítems afirmativos (por ejemplo, “El sistema fue fácil de usar”) y puntuaciones más bajas en los ítems con formulación negativa (por ejemplo, “Necesitaría ayuda técnica para usarlo”), lo cual es coherente con una evaluación favorable.

En general, los resultados del SUS indican que las mejoras aplicadas al diseño de las visualizaciones tuvieron un impacto positivo sobre la percepción de usabilidad. A diferencia de otros indicadores, este efecto se manifestó de forma clara en todos los participantes del grupo experimental y alcanzó niveles de significación estadística elevados. Estos hallazgos refuerzan la utilidad del modelo como herramienta para asistir en la revisión de visualizaciones desde una perspectiva centrada en la experiencia del usuario.

4.3. Percepción general del uso del modelo

La percepción del equipo sobre el uso del modelo fue evaluada mediante una encuesta con tres preguntas abiertas. A continuación, se presenta una síntesis de las respuestas obtenidas.

Etapas de aplicación dentro del proceso de diseño. Todos los participantes coincidieron en que el modelo resulta útil una vez que se dispone de una primera versión de la visualización. Dos de ellos lo situaron en etapas finales, como paso de validación antes de publicar o entregar el material, mientras que otro propuso su uso en fases intermedias, cuando aún hay margen para realizar ajustes estructurales. En general, se percibe como una herramienta de revisión, con valor en momentos donde ya existe una base sobre la cual iterar.

Tareas o decisiones más asistidas por el modelo. Las respuestas indicaron que el modelo resulta especialmente útil para detectar problemas que suelen pasar desapercibidos, como omisiones en la narrativa, falta de atribuciones o deficiencias en la legibilidad. También se lo valoró como guía para decidir por dónde comenzar a mejorar una visualización, especialmente cuando el diseño ya parece correcto, pero genera dudas sobre su claridad o efectividad. Estas observaciones coinciden con los resultados previos en cuanto a utilidad y aplicación de las recomendaciones.

Obstáculos para su uso en escenarios reales. Entre las limitaciones señaladas se encuentran la posible sobrecarga en equipos con muchas visualizaciones, la extensión del cuestionario para usuarios no expertos, y la necesidad de integrar la herramienta al flujo de trabajo de forma natural. Frente a estos puntos, los participantes destacaron que la modalidad asistida y la integración temprana del modelo podrían reducir significativamente estas barreras.

En conjunto, los resultados obtenidos a lo largo de las tres dimensiones de análisis permiten caracterizar el desempeño de NaVi-Q en un entorno aplicado. El modelo demostró ser capaz de integrarse en un proceso de trabajo real, generar evidencia empírica sobre la calidad de las visualizaciones y fomentar la reflexión del equipo en torno a sus propias prácticas de diseño. Más allá de los valores cuantitativos obtenidos, la combinación de métodos y perspectivas permitió comprender de manera más completa cómo el modelo puede apoyar y estructurar la evaluación de visualizaciones y orientar mejoras concretas en su diseño.

5. Conclusiones

Este artículo presentó un estudio de caso del modelo NaVi-Q en un entorno real de diseño de visualizaciones orientadas a la gestión de deuda técnica. Los hallazgos obtenidos permiten afirmar que el modelo contribuye de manera efectiva a la mejora de productos visuales, al proporcionar una estructura clara para revisar buenas prácticas y generar recomendaciones accionables.

En particular, se observó que las sugerencias generadas fueron percibidas como útiles por el equipo, especialmente aquellas vinculadas con aspectos narrativos. Las mejoras aplicadas a partir de estas sugerencias se tradujeron en beneficios concretos en términos de comprensión y usabilidad, y el modelo fue valorado como una herramienta valiosa de validación en etapas intermedias o finales del proceso de diseño. No obstante, también se identificaron desafíos para su adopción plena, como la necesidad de integrarlo de forma fluida al flujo de trabajo habitual.

Como líneas futuras, se propone extender su aplicación a otros dominios, explorar su integración con métodos de evaluación automática basados en inteligencia artificial, e investigar su impacto en ciclos iterativos de diseño mediante estudios longitudinales.

Referencias

1. E. Segel and J. Heer, "Narrative visualization: Telling stories with data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1139–1148, 2010.
2. J. Hullman and N. Diakopoulos, "Visualization rhetoric: Framing effects in narrative visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2231–2240, 2011.
3. L. Bounegru and J. Gray, *The Data Journalism Handbook: Towards a Critical Data Practice*. Amsterdam: Amsterdam University Press, 2 ed., 2021.
4. S. Almasi, A. Sim, Y. Jeon, H. Ghadiri, and M. Aliari, "Usability evaluation of dashboards: A systematic literature review of tools," *BMC Medical Informatics and Decision Making*, vol. 23, no. 1, p. 40, 2023.

5. J. I. Necochea-Chamorro and L. Larrea-Goycochea, "Business intelligence applied in the corporate sector: Asystematic review," TEM Journal, vol. 12, no. 4, pp. 2225–2234, 2023.
6. J. a. Caldeira, F. Brito e Abreu, J. Cardoso, R. Simões, T. Oliveira, and J. Pereira dos Reis, "Software development analytics in practice: A systematic literature review," Archives of Computational Methods in Engineering, vol. 30, pp. 2041–2080, 2023.
7. R. Kosara and J. Mackinlay, "Storytelling: The next step for visualization," Computer, vol. 46, no. 5, pp. 44–50, 2013.
8. N. Errey, J. Liang, T. Leong, and D. Zowghi, "Evaluating narrative visualization: A survey of practitioners," International Journal of Data Science and Analytics, vol. 18, pp. 19–34, 2024.
9. H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale, "Empirical studies in information visualization: Seven scenarios," IEEE Transactions on Visualization and Computer Graphics, vol. 18, no. 9, pp. 1520–1536, 2012.
10. C. Plaisant, "The challenge of information visualization evaluation," in Proceedings of the Working Conference on Advanced Visual Interfaces, AVI '04, (New York, NY, USA), p. 109–116, Association for Computing Machinery, 2004.
11. S. Carpendale, Evaluating Information Visualizations, pp. 19–45. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.
12. J. Heer and M. Bostock, "Crowdsourcing graphical perception: using mechanical turk to assess visualization design," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '10, (New York, NY, USA), p. 203–212, Association for Computing Machinery, 2010.
13. E. Wall, M. Agnihotri, L. Matzen, K. Divis, M. Haass, A. Endert, and J. Stasko, "A heuristic approach to value-driven evaluation of visualizations," IEEE Transactions on Visualization and Computer Graphics, vol. 25, no. 1, pp. 491–500, 2019.
14. X. Lan, Y. Shi, Y. Zhang, and N. Cao, "Smile or scowl? looking at infographic design through the affective lens," IEEE Transactions on Visualization and Computer Graphics, vol. 27, no. 6, pp. 2796–2807, 2021.
15. A. L. Airaldi, J. Andrés Carruthers, E. Irrazábal, and A. Diaz-Pace, "Measuring narrative visualizations: A model-based evaluation framework and validation," IEEE Access, vol. 13, pp. 87350–87368, 2025.
16. C. Forsell and J. Johansson, "An heuristic set for evaluation in information visualization," in Proceedings of the International Conference on Advanced Visual Interfaces, AVI '10, (New York, NY, USA), p. 199–206, Association for Computing Machinery, 2010.
17. B. Saket, A. Endert, and J. Stasko, "Beyond usability and performance: A review of user experience-focused evaluations in visualization," in Proc. BELIV, 2016.
18. M. Tory and T. Möller, "Evaluating visualizations: Do expert reviews work?," IEEE Computer Graphics and Applications, vol. 25, no. 5, pp. 8–11, 2005.
19. M. A. Borkin, Z. Bylinskii, N. W. Kim, C. M. Bainbridge, C. S. Yeh, D. Borkin, H. Pfister, and A. Oliva, "Beyond memorability: Visualization recognition and recall," IEEE Transactions on Visualization and Computer Graphics, vol. 22, no. 1, pp. 519–528, 2016.
20. H. W. Wang, K. Lin, A. Cohen, R. Kennedy, Z. Zwald, C. Nobre, and C. X. Bearfield, "Do you "trust" this visualization? an inventory to measure trust in visualizations," 2025.
21. S. Nowak, L. Bartram, and T. Schiphorst, "A micro-phenomenological lens for evaluating narrative visualization," in Proc. BELIV, 2018.
22. C. North, "Toward measuring visualization insight," IEEE Computer Graphics and Applications, vol. 26, no. 3, pp. 6–9, 2006.
23. International Organization for Standardization, "Systems and software quality requirements and evaluation (square): Iso/iec 25010," 2020.

24. International Organization for Standardization, “ISO/IEC 33003:2015 information technology – process assessment – requirements for process measurement frameworks.” <https://www.iso.org/standard/54158.html>, 2015. ISO/IEC 33003:2015.
25. R. K. Yin, Case Study Research and Applications: Design and Methods. Thousand Oaks, CA: SAGE Publications, 6th ed., 2017.
26. S. Lee, S.-H. Kim, and B. C. Kwon, “Vlat: Development of a visualization literacy assessment test,” IEEE Transactions on Visualization and Computer Graphics, vol. 23, no. 1, pp. 551–560, 2017.
27. J. Brooke, “Sus: A “quick and dirty” usability scale,” in Usability Evaluation in Industry (P. W. Jordan, B. Thomas, B. A. Weerdmeester, and A. L. McClelland, eds.), pp. 189–194, Taylor Francis, 1996.

MIIPP: Un enfoque datadriven para la mejora continua de la estimación de esfuerzo ágil

Nicolás Tortosa

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación (CInApTIC), Argentina.
nicotortosa@gmail.com

César J. Acuña

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación (CInApTIC), Argentina.
csr.acn@gmail.com

Noelia Pinto

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación (CInApTIC), Argentina.
ns.pinto@gmail.com

Gabriela Tomaselli

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación (CInApTIC), Argentina.
gabriela.tomaselli@gmail.com

Yoel Marain

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación (CInApTIC), Argentina.
jmarain62@gmail.com

Florencia Gonzalez

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación (CInApTIC), Argentina.
floo.g1096@gmail.com

Ezequiel Machuanin

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación (CInApTIC), Argentina.
ezequieladrianmachuanin@gmail.com

Chiara Diez

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación (CInApTIC), Argentina.
diezchiara@gmail.com

Resumen

La Estimación de Esfuerzo es un desafío clave en el desarrollo ágil de software, que exige equilibrar precisión y adaptabilidad en ciclos dinámicos. Si bien Planning Poker es una de las técnicas más adoptadas, su aplicación mecánica a menudo resulta en inconsistencias debido a factores como la definición incompleta de requisitos, la variabilidad en la experiencia del equipo y las presiones externas. Este artículo propone un Marco de Interpretación Inteligente para Planning Poker (MIIPP), que transforma la herramienta de un proceso de consenso a un sistema de aprendizaje continuo. El MIIPP se fundamenta en la premisa empírica de que la efectividad de la estimación reside en el esfuerzo invertido en interpretar y contextualizar los datos generados. El marco introduce requisitos

rigurosos para las Historias de Usuario (incluyendo Criterios de Aceptación, Mockups y Dependencias) y detalla una fase de estimación colaborativa inicial que prioriza el establecimiento de un modelo mental compartido, sentando las bases para la integración posterior de asistentes inteligentes y ciclos de retroalimentación.

Palabras clave: Estimación ágil; Planning poker; Mejora continua; Datadriven; Calidad de software..

1. Introducción

El proceso de Estimación de Esfuerzo constituye un desafío central y persistente en el desarrollo ágil de software. En entornos que exigen un alto nivel de adaptabilidad y ciclos de entrega cortos, los equipos deben lograr un equilibrio delicado entre la necesidad de obtener una planificación precisa y la incertidumbre inherente a los requerimientos cambiantes. La efectividad de la estimación se ve comprometida por diversos factores, entre los que destacan la definición incompleta o ambigua de los requisitos, la alta variabilidad en la experiencia técnica de los miembros del equipo y las presiones externas de los stakeholders que pueden sesgar las predicciones iniciales.

Dentro de las técnicas disponibles, el Planning Poker, desarrollado por Cohn [1], se consolida como la herramienta de estimación relativa más adoptada en la industria ágil. Esta práctica está diseñada para fomentar la participación, el diálogo y la construcción de un consenso colectivo. No obstante, la evidencia empírica reciente [2] ha señalado que la mera aplicación mecánica del Planning Poker no garantiza la precisión, resultando en inconsistencias. El análisis empírico concluye que el problema no radica en la herramienta, sino en la baja madurez estimativa y en la insuficiente integración del feedback y la interpretación contextualizada de los datos generados durante las sesiones de estimación. La clave reside en el esfuerzo dedicado a trascender la obtención del consenso, utilizando la dispersión de votos como un valioso indicador de riesgo e incertidumbre.

En respuesta a esta brecha metodológica y empírica, este artículo propone un Marco de Interpretación Inteligente para Planning Poker (MIIPP). El MIIPP transforma la técnica de una herramienta de consenso a un sistema de aprendizaje incremental y datadriven, que utiliza los metadatos de la sesión (dispersión, riesgos subyacentes) como inputs para un proceso asistido de mejora continua. La base del marco requiere la imposición de condiciones de calidad rigurosas para la Historia de Usuario, incluyendo la definición de Criterios de Aceptación, Mockups y Dependencias, sentando las bases para una estimación más robusta.

El resto del artículo se estructura de la siguiente manera: la Sección 2 presenta el marco teórico, revisando los fundamentos de la Estimación Ágil y el Planning Poker. La Sección 3 detalla el desarrollo del Marco de Interpretación Inteligente para Planning Poker (MIIPP), describiendo las fases de calidad de requisitos, estimación colaborativa y el ciclo de retroalimentación inteligente. Finalmente, la Sección 4 presenta las conclusiones y contribuciones del trabajo, proyectando líneas de investigación futuras.

2. Marco teórico

Para establecer la base conceptual de la propuesta MIIPP, esta sección revisa los pilares teóricos que rigen la estimación de esfuerzo en el desarrollo ágil de software, haciendo foco en la técnica de Planning Poker, y presenta la metodología empírica previa que sustenta la necesidad de un enfoque inteligente y contextualizado.

2.1. Estimación Ágil y Planning Poker

La estimación en entornos ágiles difiere de los enfoques tradicionales por su énfasis en la evaluación relativa y la participación de todo el equipo. El objetivo es asignar un valor, generalmente en *Story Points*, que refleje el esfuerzo ideal necesario, el cual es una combinación de: (i) cantidad de trabajo, (ii) complejidad y (iii) riesgo o incertidumbre inherente [1]. Los *Story Points* se basan en la comparación con historias de usuario de referencia ya completadas (*baselines*), priorizando la coherencia sobre la precisión temporal absoluta.

El Planning Poker es una técnica basada en el consenso, donde los miembros del equipo votan simultáneamente. La dispersión de los votos obliga a los miembros con valores extremos (más alto y más bajo) a justificar su postura, lo que facilita la identificación de riesgos ocultos y la construcción de un modelo mental compartido del trabajo a realizar. Sin embargo, su eficacia está condicionada por factores internos y externos identificados en la literatura [3, 4, 5], tales como la ambigüedad de requisitos y la presión de los stakeholders.

2.2. Fundamento Empírico del MIIPP

El desarrollo del Marco de Interpretación Inteligente (MIIPP) se fundamenta en un trabajo de investigación previo que combinó la revisión sistemática de literatura (RSL) con un estudio de campo.

1. *Revisión Sistemática de Literatura (RSL)*: Se realizó una RSL exhaustiva para identificar las estrategias, técnicas y brechas de investigación en la estimación de esfuerzo ágil. Esta revisión proporcionó el marco conceptual para comprender los desafíos más citados por la academia y los vacíos en la práctica [4].
2. *Encuesta a la Industria*: Basándose en los hallazgos de la RSL, se diseñó y administró una encuesta [2] dirigida a profesionales activamente involucrados en la planificación y estimación en equipos ágiles (desarrolladores, *Product Owners* y *Scrum Masters*) de la región. El estudio de campo permitió obtener la percepción de la industria sobre la adopción de técnicas, las herramientas utilizadas y, crucialmente, los factores internos y externos que perciben como más influyentes en la precisión de la estimación.

Los hallazgos empíricos confirmaron que si bien el Planning Poker es la herramienta preferida, su precisión es baja o moderada para una porción significativa de los equipos. La conclusión principal fue que la efectividad de la estimación no se logra con la herramienta, sino con el esfuerzo consciente invertido en la interpretación y contextualización de la data generada [2]. Esta evidencia establece la necesidad de un enfoque metodológico, como el MIIPP, que sistematice la interpretación de la varianza en los votos para transformarla en conocimiento procesable.

3. Desarrollo

El método se fundamenta en requisitos específicos para las historias de usuarios: estas deben redactarse de manera clara y concisa, siguiendo el formato “como [rol], quiero [funcionalidad] para [beneficio]”. Adicionalmente, cada historia de usuario debe integrar:

- **Criterios de aceptación**: Con el fin de verificar que la implementación satisfaga las expectativas del cliente [1];
- **Mockups o representaciones visuales**: Para optimizar la comprensión del diseño y mitigar las ambigüedades.
- **Dependencias**: Con el propósito de identificar las relaciones técnicas o funcionales con otras historias.
- **Tipos de tareas** (p. ej., frontend, backend, pruebas, integración): Para anticipar la carga multifuncional
- **Homogeneidad de tareas**: Para garantizar que las descomposiciones posean un tamaño y complejidad semejantes, lo cual mejora las estimaciones y comparaciones.

La carencia total o parcial de estas condiciones compromete la precisión del proceso, incrementa la dispersión en las deliberaciones y disminuye la fiabilidad del consenso y las recomendaciones derivadas de la inteligencia artificial.

3.1 Fase 1: Estimación colaborativa inicial

Como se observa en la Figura 1, en este estado inicial del proceso convergen múltiples dimensiones: prácticas comunicacionales (lectura, aclaración, revalidación), criterios de calidad narrativa (como claridad, concisión, completitud contextual, consistencia terminológica, enfoque en el valor de negocio, trazabilidad y evaluabilidad) y mecanismos de estandarización documental. La correcta articulación de estos elementos constituye la base epistémica desde la cual se construirá todo el proceso de estimación y desarrollo posterior.

La precisión de esta fase es fundamental, ya que cualquier ambigüedad temprana o interpretación divergente podría comprometer la exactitud del proceso completo, dado que las estimaciones posteriores —tanto humanas como asistidas por IA— se construyen sobre la base de esta comprensión inicial. Si la historia de usuario es interpretada de manera inconsistente por los distintos miembros del equipo [7], se generarán estimaciones con fundamentos diferentes, lo cual afecta directamente la coherencia del backlog, distorsiona la planificación y puede conducir a una asignación ineficiente de recursos o a la entrega de soluciones que no satisfacen las necesidades reales del negocio.

Como resultado formal se obtiene un conjunto de estimaciones individuales preliminares, que serán utilizados como insumo para las siguientes fases.

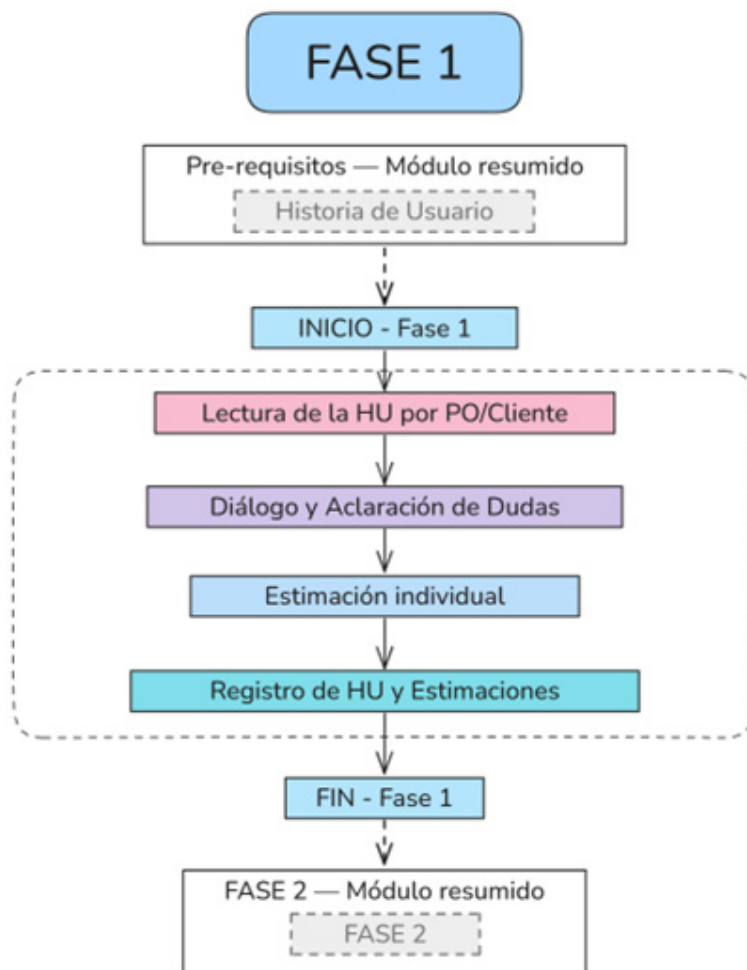


Figura 1. Flujo operativo de la Fase 1 (estimación colaborativa inicial).

El desarrollo de esta fase como se ilustra en la Figura 1 comienza con la lectura de la historia de usuario (HU), realizada por el Product Owner o el cliente. En el contexto ágil, el Product Owner es el representante del cliente dentro del equipo, con conocimiento profundo sobre las necesidades del negocio, capacidad de toma de decisiones y disponibilidad continua para evitar cuellos de botella. Además de ser responsable de la gestión eficaz del Product Backlog, su rol clave es maximizar el valor del producto resultante del trabajo del equipo.

Esta lectura inicial, que a simple vista podría parecer un acto trivial, constituye en realidad una estrategia comunicacional deliberada. Al realizarla en voz alta y en presencia del equipo, se garantiza que todos los miembros reciban el mismo insumo semántico y contextual, sentando las bases para una interpretación común de la HU que se va a estimar. En otras palabras, esta instancia cumple una función de homogeneización del entendimiento, minimizando la posibilidad de malentendidos posteriores que podrían afectar tanto la calidad de las estimaciones como la planificación del sprint.

A continuación, se abre un espacio de diálogo para aclarar dudas, donde los desarrolladores, analistas técnicos y demás miembros del equipo pueden formular preguntas, señalar ambigüedades o plantear observaciones. Esta práctica, sustentada en la noción de modelos mentales compartidos —entendidos como representaciones comunes y coherentes del problema y de sus posibles soluciones, fundamentales para una comprensión unificada y para la alineación del equipo—, asegura un marco conceptual unificado para el trabajo posterior, reduce retrabajos, acelera la toma de decisiones y refuerza la base epistémica del proceso. La creación de estos modelos mentales compartidos se logra precisamente a través de este diálogo abierto, donde la comunicación activa permite unificar perspectivas y disipar malentendidos antes de que se conviertan en problemas estructurales.

Durante esta fase, se puede construir una matriz preliminar de los posibles componentes técnicos involucrados (p. ej., módulos, servicios, interfaces, datos), sin llegar a una descomposición funcional completa.

Cuando las HU son excesivamente abstractas, vagas o carecen de delimitación suficiente, el equipo debe decidir si es necesario reestructurar la HU, dividirla en historias más manejables, o postergarla hasta que su contenido sea más claro y preciso. Dichas HU vuelve al product backlog para su posterior tratamiento en una futura ceremonia de estimación.

Posteriormente, cada miembro del equipo realiza una estimación individual y silenciosa del esfuerzo requerido para implementar la HU, utilizando puntos de historia. Esta unidad de medida es subjetiva y no representa estrictamente la complejidad de desarrollar la HU sino el “esfuerzo ideal necesario” (habitualmente expresado en jornadas) [8]. Los puntos de historia reflejan una combinación de tres dimensiones:

a) Cantidad de esfuerzo necesario para desarrollar la HU

La asignación de puntos se basa en una comparación relativa: una historia estimada en 2 puntos representa aproximadamente el doble de esfuerzo que una de 1 punto [1]. Para asegurar la coherencia, el equipo utiliza un conjunto cerrado de valores predefinidos. Se adopta la secuencia de Fibonacci (1, 2, 3, 5, 8, 13, 21...), la cual es particularmente útil por su naturaleza no lineal. A medida que los valores aumentan, también lo hace la distancia entre ellos, reflejando de manera más realista la creciente incertidumbre asociada a historias de mayor tamaño [1]. En este contexto, debatir si una historia merece 18 o 20 puntos es improductivo y puede ralentizar el proceso. El uso de valores más espaciados ayuda a mitigar la falsa percepción de una exactitud que no existe [8].

Además, esta práctica se alinea con la teoría de la sabiduría de las multitudes [9], la cual sostiene que las estimaciones colectivas tienden a ser más precisas cuando se recogen de manera independiente y sin que los miembros del grupo estén influenciados por opiniones previas. Esta teoría se alinea a la práctica ya que en Planning Poker se cumplen las siguientes condiciones claves:

1. *Diversidad de opiniones*: los individuos aportan diferentes conocimientos o perspectivas al estimar, ya que cada miembro del equipo tiene su propia experiencia.
2. *Independencia*: Las respuestas no presentan sesgos derivados de la influencia social, lo cual se garantiza mediante la revelación simultánea de las estimaciones.
3. *Descentralización*: cada persona usa su propia información local o experiencia para emitir su estimación inicial.
4. *Mecanismo de agregación*: existe una forma de combinar esas opiniones (promedio, votación, consenso, etc.) a través de la discusión y el acuerdo para llegar a una estimación final.

b) Complejidad de su desarrollo

Además del esfuerzo, la asignación de puntos de historia debe contemplar la complejidad técnica asociada a la implementación de cada historia de usuario. La complejidad se relaciona con el grado de sofisticación de la solución, la necesidad de integrar múltiples componentes del sistema, la presencia de algoritmos avanzados o la dependencia de arquitecturas poco conocidas. Según Cohn (2005), la dificultad técnica constituye un factor clave porque puede aumentar exponencialmente el tiempo de desarrollo incluso cuando el esfuerzo aparente sea similar al de otras tareas de menor complejidad. En este sentido, una historia que requiere aplicar un patrón de diseño novedoso, integrar múltiples servicios externos o modificar módulos críticos de la arquitectura, merece un puntaje más alto en comparación con otra que solo implique cambios rutinarios en la interfaz de usuario. De este modo, la complejidad actúa como un modulador que diferencia historias de esfuerzo equivalente pero de distinta dificultad técnica.

c) Riesgo inherente o incertidumbre sobre su implementación

El tercer componente considerado en la estimación mediante puntos de historia es el riesgo o incertidumbre. Este aspecto refleja tanto el nivel de desconocimiento del equipo sobre la historia como la probabilidad de que surjan obstáculos imprevistos durante su desarrollo. Cohn (2005) subraya que las historias con mayor incertidumbre —ya sea por tratarse de una tecnología nunca antes utilizada, por ambigüedad en los requisitos o por depender de factores externos no controlados— deben recibir un valor de puntos proporcionalmente más alto. Incorporar este criterio permite capturar la variabilidad inherente a los proyectos ágiles y favorece la planificación realista: mientras que el esfuerzo y la complejidad son dimensiones relativamente tangibles, la incertidumbre incorpora la naturaleza estocástica de la innovación tecnológica. En consecuencia, los puntos de historia funcionan como un instrumento integrador

que pondera no solo lo que se conoce con certeza, sino también lo que el equipo aún no puede anticipar completamente.

Este acto se realiza de manera autónoma, sin intercambio verbal o visual con otros miembros, para evitar el sesgo de anclaje, un fenómeno en psicología grupal en el que la percepción de un individuo se ve influenciada por la percepción del primero en expresarse. La decisión inicial se mantiene en suspenso hasta el momento de la revelación colectiva.

Concluida la estimación individual, cada integrante registra únicamente su estimación en el asistente de IA mediante un canal de captura simple —campo numérico, carta digital o selección en la interfaz—. Este registro es estrictamente cuantitativo, queda vinculado a la HU y al identificador del estimador, y no requiere adjuntar artefactos adicionales.

De manera simultánea, la HU es ingresada al asistente de IA, mediante diversas modalidades aceptadas por el método: (i) carga manual estructurada mediante formularios con campos normalizados (título, descripción, criterios de aceptación, dependencias, tipos de tareas, *mockups*); (ii) importación de plantillas CSV/JSON; (iii) sincronización vía API con herramientas de gestión (p. ej., Jira, Azure Boards, Trello, Notion); y (iv) integración directa con QuAGI.

En este contexto, es relevante destacar que el método contempla la posibilidad de integración directa con plataformas específicas, como es el caso de QuAGI, una herramienta complementaria, dentro del modelo QuAM desarrollado en el mismo centro de investigación. QuAGI permite la administración de los proyectos a través de la visualización del diseño preliminar del plan y de informes respecto a estados del mismo, provee información integral de las actividades, sirve como herramienta de comunicación interna entre los integrantes del equipo, entre otros. Y por otro lado, da soporte a los procesos de toma de decisiones asistiendo a los responsables mediante reportes que informan sobre evaluación de calidad del proyecto en cuestión y recomendaciones de ajustes para la mejora continua [10].

La Fase 1 constituye una etapa de estabilización interpretativa, filtrado semántico y validación estructural de las historias de usuario, cuyo impacto es determinante para la fidelidad de las fases posteriores. En ella, se combinan prácticas sociotécnicas y principios de prácticas ágiles. Aunque esta fase no genera resultados inmediatos en términos de ejecución, asegura que el proceso completo se desarrolle sobre una base epistémica sólida, coherente y verificable.

Como salida formal de la Fase 1 se obtiene un conjunto de artefactos que constituye la entrada explícita a la Fase 2. Este paquete incluye: (i) la HU validada y estandarizada —título y narrativa, criterios de aceptación, dependencias, tipos de tareas (p. ej., Frontend/Backend/QA/DevOps), *mockups*/artefactos de soporte, supuestos y restricciones— con identificador único, versión y trazabilidad; (ii) el vector de estimaciones individuales en escala de Fibonacci, un registro estrictamente cuantitativo por integrante (HU, estimador, puntaje).

Solo las HU que el equipo considera que cumplen las precondiciones/características obligatorias avanzan; las que no, se devuelven al backlog. Este conjunto, homogéneo y trazable, alimenta al asistente en la Fase 2 para el cálculo del consenso ponderado y los indicadores explicativos asociados.

3.2 Fase 2: Consenso asistido por IA y validación histórica

La segunda fase del proceso introduce la participación activa de la inteligencia artificial como facilitadora de consenso. Su rol no es reemplazar el juicio humano, sino sintetizar las estimaciones individuales generadas en la fase anterior y combinarlas con el historial acumulado del equipo, produciendo un valor de consenso sugerido que actúa como referencia inicial para la deliberación posterior. La Figura 2 ilustra el flujo operativo de esta fase.

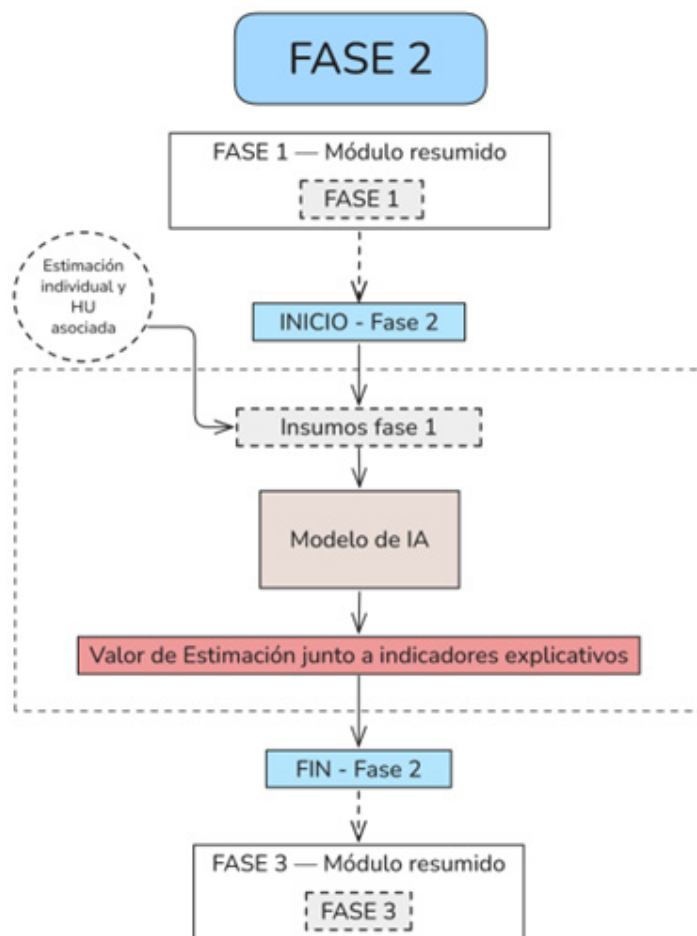


Figura 2. Flujo operativo de la Fase 2 (consenso asistido por IA y validación histórica).

Este consenso no equivale a un promedio aritmético; se calcula de forma ponderada y dinámica atendiendo a múltiples factores contextuales y de aprendizaje histórico, entre ellos: (i) el historial de historias de usuario previamente estimadas y ejecutadas; (ii) los desvíos entre estimación y esfuerzo real (de historias de usuario pasadas), tanto a nivel individual como de equipo; (iii) las similitudes de la HU actual con casos pasados en términos de complejidad, riesgos y dependencias; (iv) el grado de experticia por dominio de quienes estiman (p. ej., mayor peso relativo al especialista en frontend cuando la HU involucra predominantemente ese ámbito); y (v) la inferencia del asistente sobre la complejidad y el contexto de la HU, derivada de rasgos semánticos y técnicos extraídos de su descripción y componentes asociados.

El punto de partida de esta fase es el conjunto de estimaciones individuales generado en la Fase 1, que el asistente de IA procesa junto con la HU y sus componentes. El asistente ha sido entrenado con datos reales de equipo y proyecto mediante aprendizaje supervisado, incluyendo: (i) configuración del equipo (tamaño, roles: PO, UX/Frontend, Backend, QA, DevOps, Data/BI); (ii) histórico de HU (texto, criterios de aceptación, dependencias, tipos de tareas, artefactos como mockups, nivel de homogeneidad/descomposición); (iii) histórico de estimaciones individuales y su distribución; (iv) resultados de ejecución (esfuerzo efectivo, historias completadas/no completadas); y (v) contexto operativo (velocidad de sprint, cambios de alcance, impedimentos, disponibilidad). Gracias a este entrenamiento, el asistente puede refinar su estimación para la HU actual y ponderar cada estimación individual basándose en su precisión histórica y la pericia demostrada por cada miembro del equipo en contextos similares.

El resultado es una estimación que ya contiene la opinión colectiva del equipo, ajustada según la experiencia acumulada y la multifuncionalidad de sus miembros. De este modo, la IA actúa como memoria colectiva y catalizador de consenso, recordando quién ha sido más preciso en el pasado, en qué contextos y bajo qué condiciones, y ajustando sus cálculos en consecuencia.

Con finalidad de transparencia y trazabilidad, el asistente acompaña el valor sugerido de consenso con indicadores explicativos, como el nivel de dispersión de las estimaciones individuales, similitud con HU previas y los factores históricos que influyeron en el cálculo. Estos elementos permiten al equipo comprender no sólo qué valor sugiere el asistente, sino también por qué, lo que fortalece la confianza y legitimidad del consenso propuesto.

La Fase 2 constituye, por tanto, un espacio de convergencia asistida, donde el valor de consenso sugerido propuesto por el asistente, sustentado en la experiencia pasada y presente del equipo, funciona como un punto de partida sólido que reduce la dispersión inicial, agiliza la deliberación y refuerza la trazabilidad de las decisiones de estimación que se consolidaran en la fase siguiente.

3.3 Fase 3: Revisión deliberada y reajuste consensuado

La tercera fase opera como un espacio de deliberación guiada en torno al valor de consenso sugerido en la Fase 2. El equipo revisa ese valor inicial —que ya integra las estimaciones individuales, el historial del equipo y la ponderación por experticia— junto con los indicadores explicativos provistos por el asistente de IA (p. ej., dispersión, similitudes con HU previas y factores influyentes). El objetivo es contrastar ese punto de partida con el conocimiento contextual del equipo y su experiencia acumulada, entendiendo que, en entornos ágiles, el conocimiento es un capital colectivo y multifuncional.

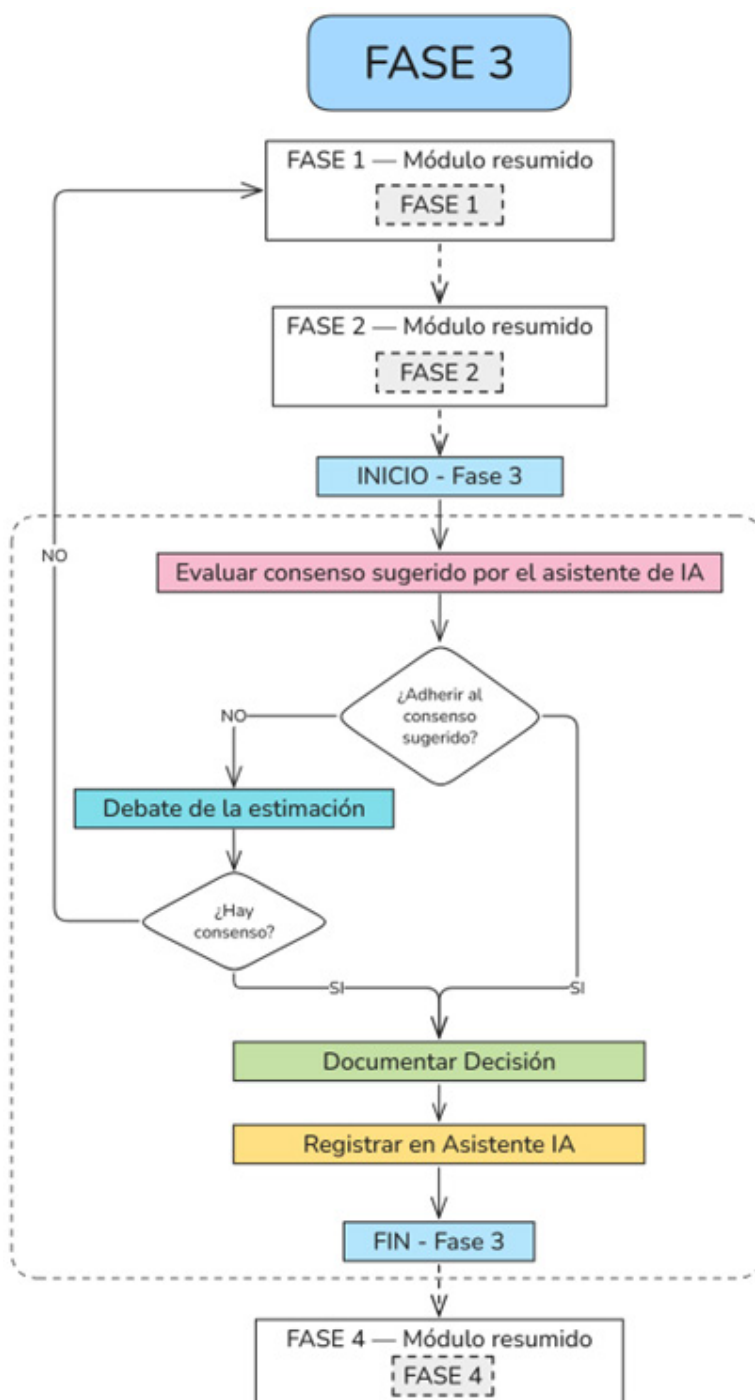


Figura 3. Flujo operativo de la Fase 3 (Revisión deliberada y reajuste consensuado).

Este espacio no impone correcciones automáticas, sino que habilita un diálogo estructurado como se ilustra en la Figura 3 con dos cursos posibles: (i) si el equipo adhiere al consenso sugerido por el asistente, se procede de inmediato a registrar la estimación en el asistente; (ii) si no se adhiere, se abre una deliberación partiendo del consenso propuesto y de sus indicadores explicativos. En esta discusión, cada integrante puede plantear ajustes justificados —basados en evidencia verificable o pericia de dominio— con el objetivo de alcanzar un consenso definitivo. De no lograrse, el método sugiere reiniciar la estimación (Fase 1) con una nueva ronda deliberativa que no parte de cero, sino de un insumo enriquecido: (i) consenso e indicadores, (ii) estimaciones originales, (iii) justificaciones de la primera discusión y (iv) evidencia histórica de casos análogos (desvíos, riesgos, dependencias, complejidad). Esta ronda puede ser silenciosa para minimizar el anclaje o incluir justificaciones breves, según las prácticas del equipo. En todos los casos, las decisiones deben fundamentarse en criterios explícitos y verificables, tales como cambios recientes de requisitos/prioridades, aparición de dependencias técnicas, riesgos no modelados, restricciones de planificación o información experta del dominio predominante (p. ej., UX/Frontend, Backend, QA, DevOps).

Una vez alcanzado el consenso —o el mejor acuerdo posible—, la estimación se registra en el asistente, junto con una breve trazabilidad: las principales razones del ajuste (si lo hubo) y, si corresponde, los disensos minoritarios documentados. Esta evidencia alimenta la Fase 4, proporcionando datos claros sobre las decisiones y sus fundamentos.

Más allá del número final, la fase persigue un refinamiento cognitivo: ajustar no sólo cifras, sino también supuestos, modelos mentales y percepciones de complejidad. El contraste entre la intuición del equipo y los patrones históricos señalados por el asistente contribuye a reducir la dispersión sin sacrificar la autonomía profesional ni el peso del juicio experto en el dominio pertinente.

Desde una perspectiva operativa, la fase permite: (i) reforzar la confianza en las estimaciones finales mediante validación cruzada; (ii) transparentar los fundamentos de decisiones divergentes o ajustes significativos; (iii) promover el aprendizaje colectivo sobre criterios de estimación eficaces, a corto y largo plazo; y (iv) evitar bloqueos por ausencia de consenso, gracias a una mediación semi asistida que preserva la colaboración sin imponer arbitrariedad.

En síntesis, la Fase 3 convierte el consenso automático en un consenso deliberado, acelerando la alineación interpretativa sin debilitar la deliberación humana y asegurando que la decisión final sea robusta, trazable y coherente con los objetivos técnicos y organizacionales del proyecto.

3.4 Fase 4: Retroalimentación adaptativa y aprendizaje continuo

La cuarta fase del proceso, ilustrada en la Figura 4, marca el inicio de un ciclo de mejora continua, donde el asistente de IA deja de operar únicamente como apoyo estimativo para asumir un rol de aprendiz activo dentro del ecosistema del equipo. Su función principal es analizar retrospectivamente el desempeño del proceso de estimación a la luz de los resultados del sprint finalizado, con el fin de identificar patrones, ajustar su modelo interno y optimizar futuras intervenciones.

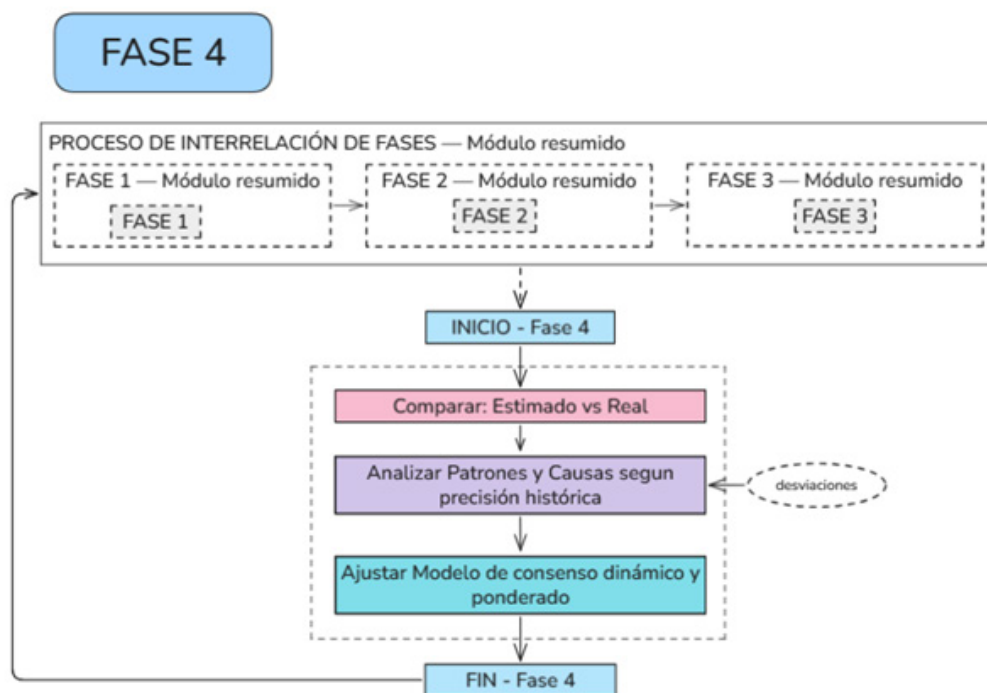


Figura 4. Flujo operativo de la Fase 4 (Retroalimentación adaptativa y aprendizaje continuo).

Al cierre de cada sprint, el equipo facilita al asistente de IA información efectiva sobre la implementación: esfuerzo real invertido (por HU), historias concluidas/no concluidas y ajustes realizados durante el proceso. Este análisis no se limita a comparar números, sino que incorpora variables cualitativas como cambios de alcance, imprevistos técnicos, retrabajos, decisiones de planificación, rotación o disponibilidad del equipo.

Este análisis de carácter mixto (cuantitativo + cualitativo) le permite al sistema modelar no sólo las desviaciones, sino también las razones detrás de ellas, identificando patrones latentes que no serían evidentes en una inspección superficial. Así, se reconoce que las desviaciones en estimación no siempre son errores, sino que pueden obedecer a dinámicas de adaptación interna.

Entre los aspectos evaluados se encuentran: (I) la precisión de las estimaciones frente al esfuerzo real acumulado; (II) el impacto de las sugerencias del asistente frente a las decisiones humanas; (III) el desempeño estimativo de cada miembro; y (IV) las características comunes de las historias con mayor desviación.

Con esta información, el modelo ajusta dinámicamente sus algoritmos de predicción, aprendiendo de las interacciones reales del equipo. Este aprendizaje incremental permite que las futuras recomendaciones sean cada vez más específicas, realistas y adaptadas al estilo de trabajo particular del equipo.

Más allá del ajuste técnico del modelo, la fase permite consolidar una dinámica organizacional de aprendizaje, donde tanto la IA como el equipo evolucionan de manera conjunta. Cada iteración no solo mejora las predicciones futuras, sino que afina las reglas tácitas y explícitas del proceso de estimación colectiva. La IA se convierte así en un agente catalizador de reflexión, facilitando conversaciones fundadas en evidencia, disminuyendo la carga de incertidumbre estructural y promoviendo la madurez organizacional.

La Fase 4 permite así consolidar un sistema vivo de aprendizaje organizacional, donde cada iteración fortalece el entendimiento compartido entre el equipo y el asistente. No se trata simplemente de mejorar el modelo, sino de afinar el proceso completo de estimación colectiva, reduciendo la incertidumbre estructural y acelerando el desarrollo de una cultura de mejora basada en evidencia.

La fase cierra el ciclo y, simultáneamente, prepara el siguiente: tanto las personas como el asistente ingresan al próximo sprint planning con más información, mayor contexto y mejor calibración, lo que se traduce en consensos iniciales más estables, deliberaciones más breves y decisiones más trazables y confiables.

4. Conclusiones y trabajos futuros

Este artículo presentó el Marco de Interpretación Inteligente para Planning Poker (MIIPP), una propuesta metodológica que busca superar la baja efectividad de la estimación ágil, transformando la técnica de consenso en un proceso datadriven y de aprendizaje continuo.

La estimación de esfuerzo continúa siendo uno de los desafíos más significativos en la gestión de proyectos ágiles, principalmente debido a la aplicación superficial y mecánica de técnicas como el Planning Poker. La evidencia empírica analizada en este trabajo confirma que la efectividad de la estimación está intrínsecamente limitada por la baja madurez estimativa de los equipos y la omisión de un proceso formal y consciente de interpretación del dato. El consenso final, si bien es el objetivo inmediato de la herramienta, a menudo esconde ambigüedades de fondo que comprometen la planificación, distorsionan la coherencia del backlog y resultan en una asignación ineficiente de recursos. En este contexto, la principal contribución del MIIPP radica en abordar esta deficiencia mediante la imposición de requisitos rigurosos para la Historia de Usuario y la estandarización de una fase de estimación colaborativa inicial que fuerza la construcción de un modelo mental compartido. Al hacerlo, el MIIPP sienta las bases para transformar el Planning Poker de una herramienta de consenso a un sistema de captura de datos de alta calidad que minimiza la ambigüedad y maximiza la coherencia, lo cual es esencial para las fases subsiguientes asistidas por inteligencia artificial. Como continuación de esta investigación, los trabajos futuros se centrarán en la implementación y validación empírica de las fases restantes del MIIPP. Esto incluye el desarrollo de componentes inteligentes (basados en Machine Learning o reglas) que utilicen los metadatos de la dispersión de votos para generar sugerencias o alertas automatizadas, facilitando la convergencia del equipo hacia una estimación final más precisa. Se proyecta un estudio de campo comparativo con equipos de desarrollo en producción para contrastar la precisión y la satisfacción del equipo al utilizar el MIIPP frente a la metodología de Planning Poker tradicional, con miras a extender el marco a otras técnicas de estimación de esfuerzo ágil.

5. Agradecimientos

Este trabajo se enmarca en las actividades relacionadas con el proyecto de investigación y desarrollo “Métodos, técnicas y herramientas para mejorar y evaluar la calidad de la capacidad de producción de entregables dentro de un proceso ágil de desarrollo de software” (PID SIECRE0008608TC), correspondiente al Centro de Investigación Aplicada a TIC (CInApTIC) de la Universidad Tecnológica Nacional, Facultad Regional Resistencia, de la provincia del Chaco, Argentina.

Referencias

- [1] M. Cohn, Agile Estimating and Planning. Upper Saddle River, NJ, USA: Prentice Hall, 2005.
- [2] N. Tortosa, C. J. Acuña, F. González, and E. Machuanin, “Estimación de esfuerzo en entornos ágiles: Un enfoque empírico para comprender su aplicación en la Industria del Software,” Memorias de las 54 JAIIO ASSE, 2025.
- [3] J. Pasuksmit, P. Thongtanunam, and S. Karunasekera, “A Systematic Literature Review on Reasons and Approaches for Accurate Effort Estimations in Agile,” ACM Computing Surveys, vol. 56, no. 11, pp. 1–37, 2024.
- [4] M. FernándezDiego, E. R. Mendez, F. G. LadronDeGuevara, S. Abrahao, and E. Insfran, “An Update on Effort Estimation in Agile Software Development: A Systematic Literature Review,” IEEE Access, vol. 8, pp. 166768–166800, 2020.
- [5] R. C. Sandeep, M. SánchezGordón, R. ColomoPalacios, and M. Kristiansen, “Effort estimation in agile software development: a exploratory study of practitioners’ perspective,” Int. Conf. on Lean and Agile Software Development, pp. 136149, 2022.
- [6] T. Sinaga, T. Raharjo, and N. W. Trisnawaty, “Exploring issues of storybased effort estimation in Agile Software Development (ASD),” Science of Computer Programming, vol. 234, 103114, 2024.
- [7] K. Schwaber and J. Sutherland, The Scrum Guide—The Definitive Guide to Scrum: The Rules of the Game. Scrum.org, 2020.
- [8] M. G. Piattini, F. O. García, J. Garzás Parra, and M. F. Genero Bocco, Medición y estimación del software: técnicas y métodos para mejorar la calidad y la productividad. México: Alfaomega Grupo Editor, 2008.
- [9] J. Surowiecki, The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations. New York: Doubleday, 2004.
- [10] N. Pinto, N. Tortosa, B. Cabas Geat, L. Ibáñez, and C. Acuña, “Validación de la reingeniería aplicada sobre Agile Quality Framework,” in Proceedings of Argentine Symposium on Software Engineering (ASSE), Buenos Aires, Argentina, 2018.

De la mejora continua a la formación por competencias: Evolución en la enseñanza de Ingeniería del Software (2015–2025)

Maria de los Angeles Ferraro

Facultad de Ciencias Exactas y Naturales (UNNE), Corrientes, Argentina.
ferraroma@exa.unne.edu.ar

Laura Gómez Solís¹

Facultad de Ciencias Exactas y Naturales (UNNE), Corrientes, Argentina.
laura.gomez@comunidad.unne.edu.ar

Resumen

En este trabajo se presenta la evolución en la enseñanza en Ingeniería de Software, un área de formación fundamental en los futuros profesionales de la Licenciatura en Sistemas de Información. Particularmente el estudio se enfoca en la asignatura Ingeniería del Software II (ISII) en la Universidad Nacional del Nordeste, mostrando la evolución desde las primeras experiencias de mejora continua (TEYET 2015), pasando por el énfasis en la calidad en el diseño y desarrollo (Jornadas de Informática 2016), hasta la implementación del actual programa basado en competencias (2023). El objetivo es mostrar cómo se integraron enfoques de aprendizaje basado en proyectos, formación por competencias y rúbricas de evaluación para consolidar una formación profesional más integral. Se destacan las mejoras alcanzadas en concordancia con los contenidos solicitados; logrado a través de un trabajo continuo sobre el proceso de enseñanza-aprendizaje. El énfasis en las competencias técnicas y transversales, los resultados de aprendizaje, la metodología de enseñanza y las lecciones aprendidas en este recorrido. Se incluye también la perspectiva de los/as estudiantes 2025, a través de una encuesta realizada. Finalmente, se plantean líneas de acción para fortalecer la integración con nuevas tecnologías, documentación con asistencia de IA sin perder calidad de software.

Palabras clave: Ingeniería del Software, Aprendizaje Basado en Proyectos, Calidad de Software, Formación basada en Competencias.

1 Introducción

1.1 A Subsection Sample

La enseñanza de la Ingeniería del Software requiere una evolución constante para responder a las exigencias del sector profesional y las nuevas metodologías de desarrollo. Este trabajo se genera con el propósito de documentar y analizar la trayectoria pedagógica de la asignatura Ingeniería de Software II (ISII), evidenciando la transición desde modelos iniciales de mejora continua hacia un paradigma integral basado en competencias. Dicha sistematización resulta fundamental para comprender el impacto de estas innovaciones y espera contribuir a la comunidad académica en su búsqueda de una formación más efectiva y alineada con la industria.

2 Metodología

La formación en Ingeniería del Software requiere integrar aspectos teóricos con prácticas que permitan a los estudiantes enfrentarse a problemas similares a los de la industria. En este sentido, el proyecto grupal de la asignatura Ingeniería del Software II constituye una experiencia clave de aprendizaje basado en proyectos,

orientada a la gestión de software y a la implementación de soluciones aplicando metodologías ágiles y principios de calidad.

La ingeniería de software es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de los programas informáticos. Esta asignatura aporta una visión integral y para ello se presentan los conceptos básicos y las principales características y técnicas a aplicar en las etapas de diseño, pruebas, desarrollo y mantenimiento de software, continuando con lo iniciado en Ingeniería del Software I. Se pretende internalizar las estrategias para construir sistemas software, fiables y eficientes, que respondan a las necesidades de los usuarios y las empresas.

Como objetivos generales de enseñanza de la asignatura, se pretende que los alumnos y alumnas adquieran la capacidad para proyectar, desarrollar y mantener sistemas, servicios y aplicaciones en el ámbito de la ingeniería del software empleando los métodos de la ingeniería del software como instrumento para el aseguramiento de su calidad.

La enseñanza de la Ingeniería del Software ha evolucionado significativamente en los últimos años, adaptándose a los cambios en la industria, las tecnologías y las demandas educativas. En la Universidad Nacional del Nordeste, la asignatura Ingeniería del Software II ha transitado un camino de mejora continua en la formación de estudiantes de la Licenciatura en Sistemas de Información. Este artículo recorre tres momentos clave: (i) la introducción de prácticas de mejora continua y matrices analíticas (2015), (ii) el énfasis en la calidad en diseño y desarrollo (2016) y (iii) la consolidación del enfoque por competencias en el programa vigente (2023). Para este último enfoque, que aún se encuentra en continua evolución se realizaron diferentes capacitaciones con el fin de adquirir e implementar estrategias para la migración de un diseño curricular hacia un modelo de formación por competencias y aprendizaje centrado en el estudiante en Ingeniería, desarrollados por el laboratorio Mecek como así también en el ámbito de la UNNE.

2.1 Respecto al dictado y los resultados pretendidos.

La metodología para el aprendizaje de los alumnos y alumnas se compone de clases Teóricas/Prácticas y clases Prácticas exclusivamente y trabajo autónomo supervisado, desde el desarrollo de aprendizaje basado en proyectos en grupos definidos según la cantidad de cursantes del dictado actual. Se introduce además al estudiante en la investigación de temas actuales, fomentando de esta manera su motivación por la actualización permanente.

En este nuevo enfoque se hace foco en el Proyecto Grupal Integrador como estrategia para lograr los resultados de aprendizajes planteados en el dictado:

Un Proyecto es un esfuerzo temporal emprendido para crear un producto, servicio o resultado único (Project Management Institute PMI, 2017).

Resultado de Aprendizaje	Unidades/Temas	Guía de Trabajos Prácticos	Actividad Formativa	Estrategia de enseñanza
RA01. Selecciona arquitecturas de sistemas para resolver problemas de contexto de la ingeniería del software, respetando los requerimientos no funcionales definidos. (CE1,2,3) (CGT-1)	Unidad 1 –Temas: 1 y 5. Unidad 2: Tema 8	GTP Guías: 0, 1,4 Proyecto Grupal Integrador	Clase Teórico-Práctica Clases Prácticas/ Laboratorio	1
RA02 Aplica patrones de diseño para proyectar y desarrollar soluciones a problemas de contexto de la ingeniería del software, optimizando los recursos disponibles con responsabilidad profesional (CGT-2)	Unidad 1, Temas: 3, 4 y 5. Unidad 2: Temas: 6 y 8 Recupera saberes de otras asignaturas, como Ingeniería del Software I, del 3er año Retoma conceptos de asignaturas anteriores. Como: Programación Orientada a Objetos del segundo año y Taller de Programación II, del tercer año.	GTP 2,3,5 Proyecto Grupal Integrador	Clase Teórico-Práctica Clases Prácticas/ Laboratorio	1
RA03 Aplica verificación y pruebas del sistema, para verificar los componentes construidos, según los requerimientos funcionales y no funcionales definidos para el proyecto. (CGT-2)	Unidad 2. Tema 7.	GTP 4 Proyecto Grupal Integrador	Clase Teórico-Práctica Clase invertida Clases Prácticas/ Laboratorio	2
RA04. Verifica los Diseños de Arquitectura utilizados para realizar la implementación (parcial) de la solución al problema del mundo real, observando el atributo de calidad del software de trazabilidad, respecto a lo definido en la elaboración del documento seguimiento del proyecto. (CE1,2,3) (CGT-1,2)	Unidad 3: Tema 9. Articula formato de presentación con asignatura del quinto año Proyecto Final de Carrera.	Proyecto Grupal Integrador	Clase Teórico-Práctica Clases Prácticas/ Laboratorio	3
RA05. Utiliza comunicación efectiva para realizar exposición de la solución de software diseñada utilizando conceptos, terminología específica y atributos de calidad del software, respecto a las consignas establecidas en cada actividad práctica. (CGS-1,2)	Todos los Temas del Programa Temas de investigación orientados al perfil profesional	Entregas Proyecto Grupal Integrador Entrega Guías prácticas asignadas para resolución. Entrega Tema de investigación individual	Estudio y trabajo en grupo, Estudio y trabajo individual	4

Tabla 1 Resultados de Aprendizaje y estrategias para alcanzarlos

Referencias Estrategia de enseñanza

- 1 Clases teóricas prácticas Presenciales. Material de lectura adicional. Disponible en aula virtual de complemento a la actividad presencial. Presentación de casos del mundo real, en clase para generar debate. Aprendizaje basado en proyectos. Tutorías por UNNE Moodle.

- 2 Clases teóricas y prácticas. Presenciales. Material de lectura adicional video tutoriales de implementación de normas de calidad ISO/IEC 29119. Disponible en aula virtual de complemento a la actividad presencial. Presentación de casos del mundo real, y grandes problemas ocasionados por falta de aseguramiento de calidad relacionado a verificación y pruebas. Presentación de Videos de casos reales: Toyota, Lanzamientos de naves al espacio, etc. Generar debate en clase. Aprendizaje basado en proyectos. Tutorías por UNNE Moodle.
- 3 Clases teóricas y prácticas. Presenciales. Material de lectura adicional, videos tutoriales. Disponible en aula virtual de complemento a la actividad presencial. Presentación de casos implementados en otros años. Presencial, nuestro resultado esperado. Clase de articulación PFC, presentación de formatos y repositorios digitales académicos/ científicos disponibles. Aprendizaje basado en proyectos. Tutorías por UNNE Moodle.
- 4 Defensa del Proyecto Grupal Integrador. Aprendizaje basado en proyectos. Defensa de Guías prácticas asignadas para resolución. Exposición/ presentación Tema de investigación individual. Presentación de los informe de cada trabajo a evaluar por UNNE Virtual

2.2 Evaluación.

El proceso de evaluación presenta una oportunidad de reforzar el proceso de enseñanza/ aprendizaje; otorgando reforzar características que permiten seguir aprendiendo al alumno y además una oportunidad para mejorar los criterios que el docente aplica en el proceso de enseñanza; por lo tanto se debe considerar como una oportunidad de mejora para el alumno y para el docente. Desde este punto de vista, no se considera a la evaluación como una instancia que se genera únicamente en las fechas establecidas en la planificación para la toma de exámenes parciales; sino que debe estar presente de manera continua en todo el dictado de la asignatura. Con este propósito, se plantea en la tabla 1, diferentes avances de entregas, como así también defensas individuales y grupales para realizar un seguimiento de los resultados esperados.

Las diferentes fechas de entregas se establecen en la planificación de la asignatura, se presentan la primera clase de inicio de cuatrimestre y quedan publicadas en el aula virtual en Moodle. En esta presentación de pautas de entregas y fechas establecidas se trabaja con los/as estudiantes habilidades blandas respecto a la responsabilidad y cumplimiento en tiempo en forma esperado del futuro egresado/a.

Instrumentos de Evaluación.

Formularios de Criterios de Evaluación: Los Formularios de Criterios de Evaluación; también llamados 'Matrices Analíticas' [1]; permiten establecer claramente los criterios de evaluación, realizar un seguimiento de los trabajos presentados. Posibilitan obtener una evaluación de manera directa, aplicando criterios claros y que reflejan las pautas solicitadas por la cátedra para la presentación de los distintos trabajos. Estos formularios, son presentados en conjunto con las pautas del trabajo a realizar; permitiendo realizar una autoevaluación. Esto pretende además: por un lado transparentar el 'porqué de la nota obtenida' y por el otro; lograr un espíritu crítico sobre el futuro profesional.

Se enfatiza el uso de comunicación efectiva y exposición de soluciones de software, con evaluación a través de informes, presentaciones y debates. Las modalidades de presentación son presenciales y también asíncronas a través de la realización de un video con sus pautas correspondientes entregado a través de la plataforma Moodle.

La metodología incluye actividades de aprendizaje basado en proyectos [3], defensa de proyectos integra-dores, resolución de guías prácticas, exposiciones individuales y en grupo, análisis crítico de textos, y evaluación de conocimientos y habilidades de comunicación y organización.

Evaluación Continua del Proceso de Enseñanza.

El modelo de enseñanza basado en competencias implica la aplicación de metodologías e instrumentos de evaluación que permitan conocer el nivel de desarrollo de las competencias que aborda la asignatura.

Tipo de evaluación	Criterio de Evaluación	RA N°	Técnica de evaluación
Diagnóstica			
Formativa	Correcta aplicación de los conceptos teóricos Utilización apropiada del lenguaje técnico específico Correcta aplicación de las técnicas de diseño de soluciones. Aplica rúbricas.	01, 02, 03, 04, 05	Coloquios Informes escritos Revisión del código Trabajos y proyectos. Pruebas de ejecución de tareas reales y/o simuladas. Sistemas de auto-evaluación. Técnicas de observación.
Sumativa	Correcta aplicación de los conceptos teóricos y técnicas de diseño de soluciones. Aplica rúbricas.	01, 02, 03, 04, 05	Examen parcial escrito Entrega / Defensa Proyecto Grupal Integrador. Entrega/ Defensa Investigación Individual Entrega/defensa de TP asignado.

Tabla 2 Tipos de Evaluación Criterio y Técnica

En la tabla 2, se presentan las diferentes estrategias que se utilizan para el proceso de evaluación a partir del nuevo enfoque basado en formación por competencias.

Criterios de evaluación: correcta aplicación de los conceptos teóricos, presentación de ejemplos, utilizar apropiadamente el lenguaje técnico específico, etc.

Técnicas de evaluación: exámenes parciales escritos, coloquios, informes escritos, Rúbricas, Pruebas objetivas, Pruebas de respuesta de selección múltiple, Trabajos y proyectos, Informes/memorias de prácticas, Pruebas de ejecución de tareas reales y/o simuladas, Sistemas de autoevaluación, Escalas de actitudes, Técnicas de observación, Portafolio, etc.

Evaluación Referente al Resultado del Aprendizaje.

Exámenes Parciales. Luego de la entrega de las notas, se genera la muestra individual de los mismos explicando los errores cometidos; se desarrolla para la clase, la solución esperada y se destacan los principales errores encontrados.

La instancia de corrección y devolución siempre representa una doble lectura; no solo debe ser considerada como una etapa de crecimiento para el Alumno, donde el mismo fortalece los puntos necesarios; sino también para el Docente, como una oportunidad de mejorar estrategias para una mejor comprensión de los temas tratados.

Respecto al trabajo de campo grupal; como ya mencionamos el mismo se basa en el aprendizaje basado en proyectos. Busca integrar no solo el contenido desarrollado en la asignatura sino también retoma saberes aprendidos en anteriormente como en Programación Orientada a Objetos o Base de Datos, cumpliendo con las buenas prácticas a la hora de realizar un Proyecto de Software. Las pautas que debe cubrir el trabajo, disponibles desde la primera clase, aplicará los conceptos de la Ingeniería de Software en su conjunto. Se toma como base el trabajo realizado en Ingeniería del Software I, o se entregan casos de estudio para selección de los mismos. Se incentiva además al alumno a realizar un trabajo relacionado al Trabajo Final de Aplicación/ Proyecto Final de Carrera; logrando de esta manera un mejor aprovechamiento de los recursos Tiempo/ Esfuerzo invertidos.

3 Contextualización del Curso en relación a la matrícula y el plantel docente

Hasta el año 2024, la cantidad de estudiantes que cursaban la materia era relativamente estable, con un promedio de aproximadamente 65 alumnos por ciclo. Sin embargo, en el año 2025, se ha observado un crecimiento exponencial en la matrícula, alcanzando los 115 estudiantes en dicho período. Este incremento refleja la creciente

demanda y la importancia que adquiere la formación en ingeniería de software en la comunidad académica, así como la necesidad de adaptar las metodologías y recursos para atender a un volumen mayor de alumnos de manera efectiva.

Por otro lado, el plantel docente encargado de la asignatura está compuesto por tres integrantes, todos de dedicación simple:

- Una profesora titular, responsable de la planificación, coordinación y evaluación del curso.
- Dos jefes de trabajos prácticos, que supervisan y guían a los estudiantes en las actividades prácticas y en el desarrollo de proyectos, brindando asesoramiento técnico y académico en sus respectivas áreas.
- Se cuenta además dos docentes auxiliares, como actividad contra cuatrimestre de su cargo, una docente graduada de la Licenciatura en Sistemas de Información, quien se suma en colaboración en las actividades prácticas. Un docente egresado de la carrera de Relaciones Laborales quien colabora en el desarrollo de los temas de habilidades blandas.

Este equipo docente trabaja en conjunto para garantizar la calidad del proceso de enseñanza-aprendizaje, gestionando tanto las actividades teóricas como las prácticas, en un contexto de crecimiento en la cantidad de alumnos, procurando mantener los estándares de calidad académica.

A continuación se presenta la tabla 3, correspondiente a Ingeniería de Software IIPlan L.S.I. (2010), a modo de información complementaria sobre los resultados en el año 2024 se implementó el nuevo programa aprobado con la implementación de la formación por competencias y además en acuerdo con las demás asignaturas del plan de carrera la nota de promoción cambió de 8 a 7.

Año	En Lista	No cursaron	Libres (Exámenes/faltas)	Regulares	Promoción
2025	117	15	37	7	58
2024	76	8	18	6	44
2023	54	8	16	12	18

Tabla 3 Resultados Cursadas Información 2023-2025

3.1 Régimen de Promoción.

Condiciones para Regularizar la Materia. 75% de Asistencia a Clases Prácticas. Aprobación del Trabajo de Campo.

Aprobación de los dos Exámenes Parciales con nota mayor o igual a 6. Están previstos dos exámenes parciales con su correspondiente recuperatorio, como así también un examen extraordinario; a esta instancia se debe llegar con un parcial aprobado.

3.2 Condiciones para promocionar la Materia (Sin Examen Final).

Contar con las condiciones del alumno regular, con el agregado de contar con parciales aprobados con promedio mayor o igual a 7 (siete); Para el promedio se consideran solamente las instancias de evaluación aprobadas.

Contar con el trabajo de Investigación, solicitado y aprobado.

Reunir estas condiciones, permite aprobar la materia sin examen final.

4 Evaluación Externa La mirada de los/as Estudiantes.

En cada cursada vamos generando diferentes encuestas que nos muestran la mirada que tienen los/as estudiantes sobre las diferentes estrategias implementadas y el desarrollo de las actividades. Recientemente y también como foco de este trabajo se realizó una nueva encuesta que centró la mirada en el proyecto integrador grupal. El cursado de la asignatura finalizó al cierre del primer cuatrimestre probablemente esto influyó en que si bien se invitó a responder a todos/as los/as cursantes solo 23 respondieron la encuesta. Se comparten algunos resultados obtenidos.

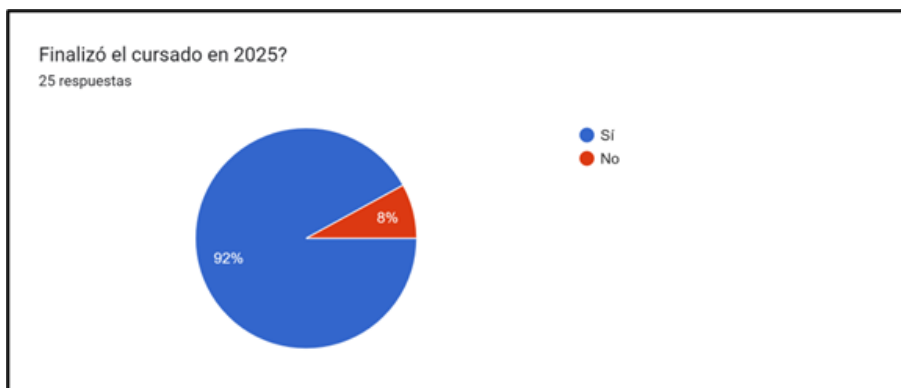


Figura 1. Cantidad de encuestados

Escala de Respuesta para Preguntas 1 a 5. Utilizar una escala de 5 niveles para seleccionar la respuesta, donde:

- Totalmente en desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Totalmente de acuerdo

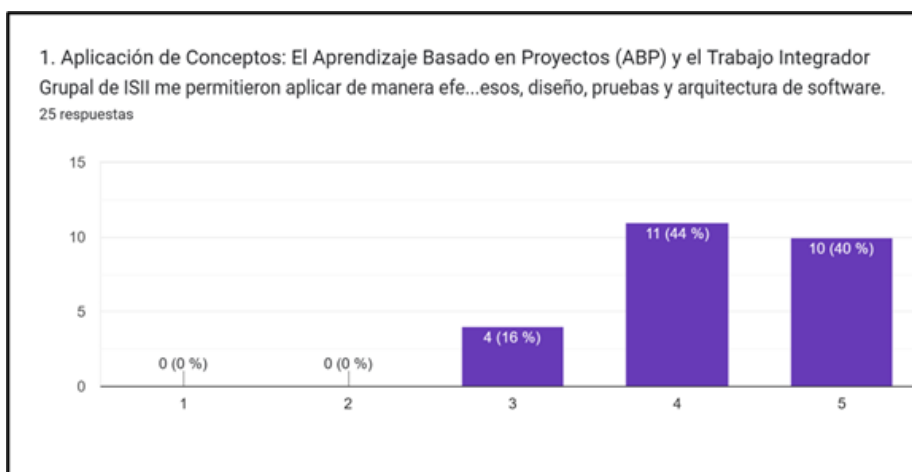


Figura 2 Aplicación de conceptos de Ingeniería de Software

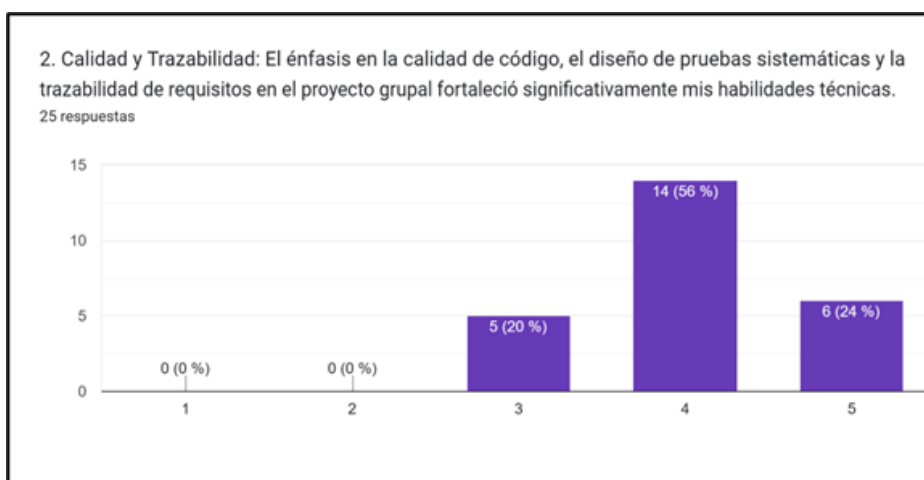


Figura 3 Valoración del fortalecimiento de habilidades técnicas



Figura 4 Valoración del trabajo en equipo y las competencias transversales.

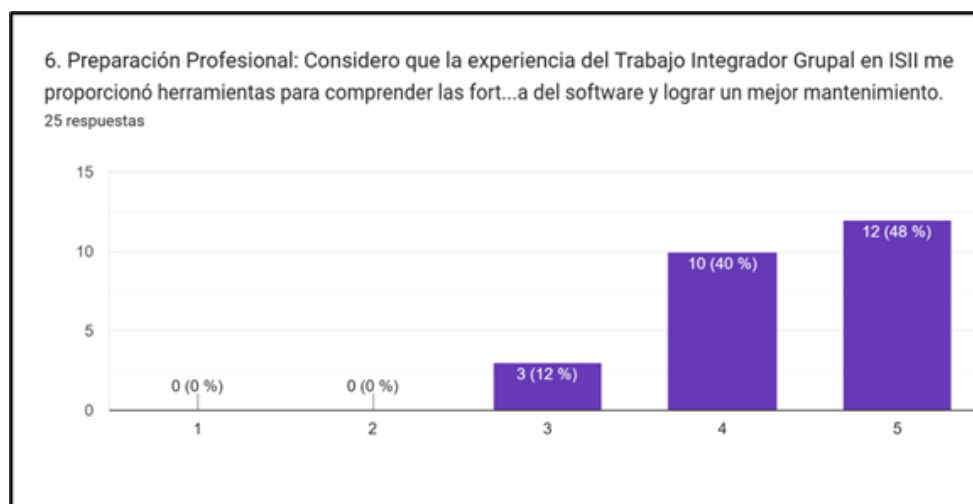


Figura 5 Nivel de preparación personal percibido

Los resultados de la encuesta aplicada a los estudiantes de ISII muestran una valoración que supera el 75% de conformidad en lo que respecta a las metodologías y estrategias implementadas. En la dimensión de aplicación de conceptos, la mayoría coincidió en que el Trabajo Integrador Grupal contribuyó eficazmente a articular procesos, diseño, pruebas y arquitectura. En relación con calidad y trazabilidad, muchos enfatizaron que el énfasis en código limpio, pruebas sistemáticas y seguimiento de requisitos fortaleció sus habilidades técnicas. Sobre metodologías ágiles y roles, los estudiantes señalaron que la asignación de roles diferenciados y el uso de Git los prepararon mejor para escenarios laborales reales. En cuanto a competencias transversales, se identificó que el trabajo grupal combinado con autoevaluación, coevaluación y defensa favoreció su comunicación, responsabilidad y colaboración. Finalmente, una mayoría sostuvo que la experiencia del proyecto integrador les proporcionó un perfil más integral para afrontar desafíos en la industria del software, especialmente al reconocer la importancia de la documentación y la reutilización del código para mantener y escalar sistemas.

5 Resultados

Los resultados de la encuesta complementan la información obtenida a través de la observación docente. Más del 80 % de los estudiantes manifestó que el trabajo integrador grupal favoreció la aplicación de conceptos teóricos y prácticos de la ingeniería del software, la colaboración entre pares y el desarrollo de habilidades

comunicativas. Sin embargo, un pequeño porcentaje señaló dificultades vinculadas a la coordinación grupal y a la carga de trabajo, lo que sugiere la necesidad de continuar ajustando la planificación y distribución de tareas. En general, la percepción estudiantil reafirma la pertinencia del modelo por competencias y su impacto positivo en la formación profesional.

Entre los desafíos identificados se destacan: la necesidad de mayor tiempo para integrar competencias complejas, la dificultad de evaluar de manera objetiva habilidades blandas, y la incorporación de nuevas tecnologías (como inteligencia artificial) en el proceso de enseñanza y la generación de documentación del Proyecto.

Si bien la mayoría de los/as cursantes hacen uso de estas herramientas es necesario fortalecer aspectos que desarrollen un pensamiento crítico sobre las mismas, tener en claro que son sin duda una ayuda pero que no reemplazan al profesional como un todo.

En este sentido se establecieron diferentes pautas de uso, publicando y socializando las mismas tanto en las clases como en el aula virtual.

6 Conclusiones y trabajos futuros

La experiencia de Ingeniería del Software II en la UNNE refleja un proceso sostenido de innovación pedagógica. El paso de un enfoque de mejora continua hacia un modelo por competencias ha permitido enriquecer tanto el proceso de enseñanza como la evaluación del aprendizaje.

Entre los trabajos futuros se plantea:

Integrar pruebas automatizadas y herramientas DevOps para fortalecer la ingeniería continua.

Incorporar métricas de calidad de software como parte del proceso formativo. Este punto se trata en profundidad en la asignatura siguiente Ingeniería de Software III, sin embargo podrían incluirse métricas iniciales del producto-proceso.

Explorar el uso de inteligencia artificial aplicada a la enseñanza y evaluación de Ingeniería del Software.

Agradecimientos

Al equipo docente que formó y forma parte de la asignatura, por su esfuerzo permanente y dedicación. A los estudiantes que se sumaron en las encuestas. A la Universidad Pública y su formación de calidad.

Referencias

1. Heidi Goodrich: Cuando la Valoración es Instrucción y la Instrucción es Valoración: Utilizando Matrices Analíticas para promover el pensamiento complejo y la comprensión.
2. Jornada Articulación de contenidos orientada a fortalecer las competencias del Analista Programador Universitario; Ingeniería del Software I, Base de Datos I, Ingeniería del Software II y Taller de Programación II, Noviembre -2013; en la FACENA UNNE. En calidad de Organizador –
3. Conferencista. Exposición de los trabajos integradores realizados por los alumnos, como así también de los temas de investigación que se trabajaron en el cuatrimestre. Paneles relacionados al perfil del profesional y el medio. Resolución N° 1128/13. Reiterado en el año 214.
4. Ferraro, Maria; Gómez Solís, Laura; Matoso, Alejandra. “Mejora continua aplicada en la enseñanza de la ingeniería del software.” X Congreso de Tecnología en Educación y Educación en Tecnología. (TEyET), Corrientes, Corrientes 2015.
5. Jornada de Integración de asignaturas de Ingeniería del Software para la elaboración del Trabajo de Graduación, mayo 2014; en la FACENA UNNE. En calidad de Organizador – Conferencista. Resolución N° 2342/14
6. Dapozo, Gladys; Greiner, Cristina; Ferraro, María; Medina, Yanina; Pedrozo Petrazzini, Gabriel; Lencina, Berenice. “Medición y estimación del software: métodos y herramientas para mejorar la calidad del software”. Anales del XVI Workshop de Investigadores en Ciencias de la Computación. (WICC 2014). Ushuaia, provincia de Tierra del Fuego, Mayo 2014.
7. Villafañe, A.; Ferraro M., Pedrozo Petrazzini G., Greiner C., Dapozo G.; Estayno, M.; “Herramienta de gestión de trazabilidad de requerimientos en proyectos de software”. Workshop Innovación en Sistemas de Software en el marco del XIX Congreso Argentino de Ciencias de la Computación. (CACIC 2013). Mar del Plata, provincia de

Buenos Aires Octubre 2013.

8. Estayno, M.; Dapozo, G.; Cuenca Pletch L.; Greiner, C.; Medina Y.; Ferraro M.; Acuña C., Pintos N. “Métodos y herramientas orientados a la calidad del software”. Anales del XIV Workshop de Investigadores en Ciencias de la Computación. (WICC 2012). Posadas, provincia de Misiones 2012.
9. Ferraro, M.; Medina, Y.; Dapozo, G.; Estayno, M. “Especificación y trazabilidad de requerimientos en el desarrollo de aplicaciones web”. II Jornadas de Investigación en Ingeniería del NEA y Países Limítrofes. (JCyT) .Resistencia, Chaco, Junio de 2012.
10. XIX Congreso Argentino de Ciencias de la Computación, realizado en la ciudad de Mar del Plata; del 21 al 25 de octubre de 2013. Presentación de la ‘Herramienta de gestión de trazabilidad de requerimientos en proyectos de software’; presentación en el Workshop Innovación en Sistemas de Software.

Pasantías Profesionales y Calidad de Software en la Administración Pública: El Rol Interdisciplinario.

María de los Angeles Ferraro

Facultad de Ciencias Exactas y Naturales y Agrimensura Universidad Nacional del Nordeste, Argentina.
msonia@exa.unne.edu.ar

Sonia I. Mariño

Facultad de Ciencias Exactas y Naturales y Agrimensura Universidad Nacional del Nordeste, Argentina.
ferraroma@exa.unne.edu.ar

Resumen.

El presente artículo describe una experiencia de vinculación entre la academia y la gestión pública en el ámbito de las Tecnologías de la Información (TI), desarrollada a través de pasantías profesionales en una organización gubernamental provincial. La iniciativa, iniciada en 2021 y ampliada en 2023. Se establecen diversos objetivos: por un lado promover el desarrollo de habilidades técnicas y blandas en los estudiantes universitarios fortaleciendo el perfil del egresado/a y por otro lado analizar los aspectos metodológicos, resultados y beneficios de esta experiencia, destacando guías prácticas para su implementación. La metodología empleada fue un estudio observacional, descriptivo y transversal, que permitió evidenciar los beneficios derivados de la integración de perfiles interdisciplinarios en el entorno gubernamental. La experiencia evidencia la importancia de promover espacios de práctica profesional que propicien la innovación, el aprendizaje y la colaboración interinstitucional, consolidando así una relación estratégica entre academia y Estado..

Palabras clave: Pasantías, Formación Profesional, Proyectos Tecnológicos. Gobierno..

1 Introducción

En el contexto actual, la gestión eficiente de los recursos tecnológicos en las instituciones públicas representa un desafío complejo que requiere de personal capacitado y de alianzas estratégicas con instituciones de educación superior. La vinculación entre la universidad y el gobierno, mediante la implementación de pasantías profesionales, emerge como una estrategia efectiva para potenciar la formación de los estudiantes y, al mismo tiempo, contribuir en la mejora de la infraestructura y los servicios del sector público.

La experiencia analizada en este artículo parte de la premisa de que las prácticas profesionales contribuyen a cerrar la brecha entre los conocimientos adquiridos en el ámbito académico y las demandas del mercado laboral, particularmente en sectores tecnológicos en constante evolución. Además, esta colaboración fomenta la transferencia de conocimientos, el desarrollo de habilidades transversales y la introducción de enfoques innovadores en la gestión pública.

En particular, la iniciativa focalizó su centro de su acción a estudiantes de diversas carreras relacionadas con Tecnologías de la Información (TI) y también a otras áreas de incumbencia relacionadas, como Licenciatura en Sistemas de Información (L.S.I.), Diseño de Imágen, Sonido y Multimedia (T.D.I.SyM), Diseño Gráfico (D.G.), Comunicación Social (L.C.S.) e Ingeniería Eléctrica (I.E.), promoviendo así la integración de perfiles especializados y transdisciplinarios. La experiencia se enmarca en la Universidad Nacional del Nordeste (UNNE), específicamente en las Facultades de Ciencias Exactas, Naturales y Agrimensura (carreras L.S.I, I.E.), Humanidades (carrera L.C.S.), Arquitectura y Urbanismo (carrera D.G), Facultad de Artes, Diseño y Ciencias de la Cultura (carrera de T.D.I.SyM), en coordinación con la Dirección de Sistemas de un ente gubernamental provincial.

El presente trabajo tiene como objetivo analizar los aspectos metodológicos, resultados y beneficios de esta experiencia, además de ofrecer ejemplos prácticos del impacto de perfiles interdisciplinarios en la gestión de TI gubernamental. Se pretende, asimismo, contribuir con un marco conceptual y práctico que sirva de referencia para futuras iniciativas similares.

2 Contextualización de la experiencia.

2.1 Antecedentes y marco teórico

Las pasantías, entendidas como prácticas profesionales, se reconocen actualmente como una estrategia formativa fundamental en la formación de profesionales técnicos y universitarios. Según Peña et al. (2016), las pasantías representan actividades de capacitación planificadas dentro del currículo, con un fuerte componente de vinculación externa entre instituciones educativas y organizaciones del sector público o privado.

Las prácticas profesionales, para la Corporación Universitaria del Caribe, permiten que el futuro profesional sume experiencias a partir de la interacción con contextos reales, donde tenga la oportunidad de aplicar los conocimientos adquiridos durante todo el proceso de formación. La práctica, además de tener un componente curricular, es concebida como un proceso de aprendizaje continuo donde el/la estudiante afianza sus competencias funcionales: saber, saber hacer y ser/convivir (Tobón, 2006); desarrolla habilidades blandas y fortalece sus saberes.

2.2 Marco institucional y contextual

La experiencia descrita tuvo lugar en la Dirección de Sistemas de una organización gubernamental provincial, responsable de administrar un Sistema Integrado de Información Financiera y otros servicios digitales. La organización atiende a más de 1,700 usuarios y requiere de mantenimiento preventivo, adaptativo y mejoras en infraestructura, comunicación, servicios y eficiencia tecnológica.

Desde 2021, se estableció un convenio entre una organización gubernamental de la provincia de Corrientes y la Universidad Nacional del Nordeste (UNNE), a través de las carreras de Licenciatura en Sistemas de Información y Diseño Gráfico, para realizar pasantías laborales rentadas. La iniciativa se enmarca en los objetivos institucionales de fomentar la vinculación universidad-sector público, contribuyendo a la formación de los/as futuros/as profesionales y además contribuir a los objetivos del Estado.

En 2023, la experiencia se amplió, incorporando perfiles de otras carreras como Ingeniería Eléctrica, Tecnicatura Universitaria en Diseño de Imagen, Sonido y Multimedia, y Comunicación Social. La ampliación buscaba atender desafíos propios de infraestructura que dan servicio de apoyo a la institución y a la continuidad del sistema en particular. Se propuso además un enfoque en la comunicación estratégica y difusión institucional, promoviendo una visión integral e interdisciplinaria.

3. Objetivos de la experiencia

- Facilitar la vinculación entre la academia y el sector público para potenciar la formación práctica de los estudiantes.
- Contribuir a la mejora de los procesos del área de TI gubernamental mediante la incorporación de conocimientos y habilidades de los pasantes.
- Promover el desarrollo de habilidades blandas como la comunicación efectiva, el trabajo en equipo y la resolución de problemas
- Diversificar perfiles profesionales en la gestión pública para responder a desafíos multidimensionales.
- Contribuir a la formación de futuros profesionales responsables, planificados y comprometidos con el servicio público.

4. Metodología

4.1. Diseño del estudio

El análisis se basó en un estudio observacional, descriptivo y transversal, realizado durante el período julio-diciembre de 2021. La metodología incluyó etapas de planificación, ejecución y evaluación, con la finalidad de sistematizar la experiencia y extraer conclusiones relevantes.

En su segunda etapa, 2023, la experiencia amplió los perfiles a incorporar con el objetivo de dar respuesta a las necesidades crecientes que atiende una Dirección de Sistemas de la Organización, como proveedora de diferentes servicios orientados a los usuarios de una aplicación crítica.

4.2. Fases de la experiencia

1. **Diseño de la convocatoria:** La iniciativa comenzó con la elaboración de una convocatoria de pasantías rentadas, dirigida a estudiantes avanzados de la Universidad Nacional del Nordeste (UNNE). Esta propuesta se sustentó en las experiencias previas de Voluntariado desarrolladas en 2020, ajustando los perfiles y requisitos según las necesidades específicas de la organización gubernamental. La convocatoria incluyó la difusión en canales institucionales, resaltando las oportunidades de formación práctica, desarrollo de habilidades técnicas y transversales, así como la relevancia de colaborar en proyectos de impacto social y administrativo. Además, se establecieron criterios claros de evaluación y selección, priorizando el compromiso, la voluntad de aprendizaje, la capacidad de trabajo en equipo y el porcentaje de materias aprobadas de los trayectos formativos. Las postulaciones ingresaban a través de la Secretaría de Extensión Universitaria de las facultades.
2. **Selección de pasantes:** En esta etapa, se recibieron un total de 45 postulaciones, de las cuales se realizó un proceso de revisión detallada de la documentación presentada, y posteriormente convocatoria a entrevistas en la Institución de gobierno. Se consideran aspectos académicos, experiencias previas, habilidades técnicas y motivaciones para integrar el programa. Finalmente, los/as estudiantes fueron seleccionados, asegurando una variedad de perfiles en lo correspondiente a la carrera de Licenciatura en Sistemas de Información, que van desde Analista-Desarrollador, Tester-Documentador hasta Soporte Técnico y Mesa de Ayuda; en este caso se convocaron 14 pasantes de esta carrera en ambas convocatorias. En lo que respecta a las demás carreras convocadas se seleccionaron 2 pasantes de cada una, completando un total de 8 pasantes en las 2 convocatorias. Esta diversificación permitió cubrir distintas áreas de interés y responder de manera integral a los requerimientos del sector público, además de promover un ambiente colaborativo e interdisciplinario.
3. **Desarrollo de las pasantías:** La fase de ejecución inició con una inducción normativa y organizacional, donde los pasantes conocieron las políticas institucionales, protocolos de seguridad de la información y las herramientas de trabajo. Posteriormente, se introdujeron metodologías de trabajo ágil, como Scrum y el uso de tableros Kanban y la herramienta de gestión utilizada en la organización, promoviendo la gestión eficiente de tareas, la planificación iterativa y la transparencia en los avances. Cada pasante recibió tareas específicas alineadas a su perfil y nivel de experiencia, fomentando que participaran en la resolución de problemáticas reales, aportando soluciones y desarrollando productos tangibles en el marco de los proyectos institucionales.
4. **Análisis de la experiencia:** Para evaluar el impacto y la efectividad del programa, se emplearon diversas técnicas de recolección de datos tanto cualitativos como cuantitativos. Se llevaron a cabo observaciones directas, análisis de informes de avances, reuniones periódicas de seguimiento y entrevistas con los pasantes y responsables de la organización. Estos elementos permitieron identificar fortalezas, áreas de mejora y el grado de inserción de los estudiantes en el entorno laboral. Además, se fomentó la retroalimentación continua para ajustar las actividades, fortalecer las competencias adquiridas y promover la autorreflexión de los participantes sobre su proceso de aprendizaje y contribución al sector público.

PASANTÍAS PROFESIONALES EN UN ÁREA TI GUBERNAMENTAL VINCULACIÓN UNIVERSIDAD - GOBIERNO



Figura 1. Fases de la propuesta Elaboración Propia

4.3. Perfil de los pasantes

Los perfiles de los pasantes se diseñaron cuidadosamente, en función de las necesidades específicas del entorno gubernamental y de la estructura organizacional de la entidad, combinando perfiles disciplinares especializados con perfiles transdisciplinarios que abordan aspectos de infraestructura, comunicación y análisis social. Esto permite una articulación más integral y efectiva en las tareas de la institución, fomentando la formación de profesionales interdisciplinarios y adaptados a los desafíos del mundo real.

Perfiles disciplinares:

- **Analista-Desarrollador:** Los/as pasantes en esta categoría estaban encargados/as de la elaboración, mantenimiento y optimización de aplicaciones y bases de datos. Su trabajo implicaba profundizar en lenguajes de programación como Java, PL/SQL, .Net y herramientas específicas como Oracle ADF, con el fin de fortalecer la infraestructura tecnológica del organismo y garantizar la eficiencia en los procesos automatizados.
- **Soporte Técnico Mesa de Ayuda:** Estos/as pasantes eran responsables de recibir, diagnosticar y resolver incidencias relacionadas con los sistemas operativos Windows y Linux, además de documentar las soluciones implementadas para facilitar futuras intervenciones. Su rol era fundamental para mantener la continuidad operativa y ofrecer soporte técnico efectivo a usuarios/as internos/as.
- **Tester-Documentador:** En esta función, los/as pasantes elaboraban casos de prueba para evaluar la funcionalidad de los desarrollos, además de documentar procedimientos y guías de ayuda para usuarios/as finales. También ejecutaban pruebas de integración y aseguraban la calidad de los sistemas, contribuyendo a la reducción de errores y mejorando la usabilidad de las aplicaciones.
- **Diseñador Gráfico:** Se encargaron de crear contenidos visuales y materiales de difusión institucional, soportando las estrategias de comunicación digital de la organización. Aplicaron principios de usabilidad, diseño centrado en el usuario (UX), teoría del color, tipografías y composición, tanto en medios digitales como impresos, favoreciendo la divulgación de información y la imagen institucional.

Perfiles interdisciplinarios y de apoyo

Desde el año 2023, se incorporaron, los siguientes perfiles:

- **Ingeniero Eléctrico:** Este perfil fue incorporado con el objetivo de evaluar, identificar y resolver problemas

asociados a la infraestructura eléctrica de la entidad gubernamental. Los/as pasantes realizaron relevamientos y diagnósticos de los equipos y configuraciones existentes, además de evaluar las características de los grupos electrógenos y la red de alimentación a los servidores, con la finalidad de proponer mejoras y desarrollar un plan de mantenimiento y optimización gradual. Realizaron planimetría, propusieron mejoras que en muchos casos se concretaron logrando un impacto altamente positivo en la institución y en la infraestructura de apoyo al servicio que brinda la aplicación central que administra el organismo.

- **Diseño de Imágen, Sonido y Multimedia:** Incluyendo perfiles orientados a la producción audiovisual, estos/as pasantes se encargaron de crear materiales multimedia institucional, como videos explicativos y materiales gráficos interactivos, para facilitar la comprensión de funciones y logros de la organización. Además, colaboraron en la difusión y gestión de contenidos en redes sociales, fortaleciendo la presencia digital del organismo.
- **Comunicación Social:** Su rol estuvo dirigido a la evaluación del impacto de las acciones institucionales mediante análisis de audiencias, estrategias de comunicación y organización de eventos. Contribuyeron en la gestión del contenido para fortalecer la interacción con la comunidad, además de desarrollar investigación en temas relacionados con la opinión pública y la percepción institucional.

Estos perfiles, en conjunto, permitieron ampliar significativamente el alcance de las pasantías, atendiendo a demandas tanto tecnológicas como comunicativas y sociales, y promoviendo una formación integral de los/as estudiantes en interacción con las múltiples facetas del sector público.

5. Resultados y análisis

A continuación se mencionan los logros en relación a los objetivos específicos indicados previamente.

5.1. Beneficios para la organización gubernamental

La incorporación de los perfiles de los/as estudiantes permitió mejorar diversos procesos internos, como el fortalecimiento de canales de comunicación digital y servicios de apoyo a la infraestructura del S.I.I.F. La colaboración fomenta soluciones innovadoras y una mayor agilidad en la resolución de incidencias, además de contribuir a la modernización de la infraestructura tecnológica.

Aporte al cumplimiento de los objetivos

- Contribución a la mejora de la infraestructura tecnológica y los procesos del área de TI gubernamental mediante la incorporación de conocimientos y habilidades de los/as pasantes.
- Diversificar perfiles profesionales en la gestión pública para responder a desafíos multidimensionales.
- Contribuir a la formación de futuros profesionales responsables, planificados y comprometidos con el servicio público.

5.2. Beneficios para los/as pasantes.

Los/as estudiantes adquirieron conocimientos especializados en entornos reales, desarrollaron habilidades transversales como trabajo en equipo, comunicación efectiva y autogestión, y perfeccionaron su exposición oral y presentación de proyectos ante autoridades. La experiencia enriqueció su formación disciplinar y les proporcionó herramientas para afrontar futuros desafíos profesionales.

Aporte al cumplimiento de los objetivos

- Facilitar la vinculación entre la academia y el sector público para potenciar la formación práctica de los estudiantes.
- Desarrollar un plan de actividades acordes a su perfil profesional, con el correspondiente seguimiento, con el fin de aportar a las competencias técnicas y actitudinales.

5.3. Integración y ambiente de trabajo

Uno de los aspectos más valorados y destacados durante el desarrollo de las pasantías fue la efectiva integración de los/as estudiantes con el personal estable de la institución. Desde las primeras etapas del programa, se promovió un entorno de trabajo colaborativo en el cual los/as pasantes participaron activamente en reuniones, proyectos conjuntos y tareas cotidianas junto al equipo estable de la institución, sin que existieran diferencias de categoría o jerarquía que puedan generar distancias o segregación.

Este enfoque facilitó que los/as pasantes se sintieran parte integral del equipo, promoviendo una comunicación abierta y la transferencia efectiva de conocimientos y experiencias. La interacción constante con personal experimentado contribuyó significativamente a su formación técnica y profesional, permitiendo un aprendizaje en el lugar de trabajo que complementa su formación académica.

Las encuestas y entrevistas realizadas al finalizar las pasantías evidenciaron que esta integración fue percibida de manera muy positiva por todos los/as involucrados/as. Los responsables y pasantes destacaron que la ausencia de barreras jerárquicas y el ambiente de respeto y colaboración favorecieron no sólo el crecimiento profesional de los/as estudiantes, sino también un clima laboral enriquecedor y motivador. La experiencia reafirmó que la convivencia en un ambiente igualitario y participativo resulta esencial para potenciar el proceso de aprendizaje y fortalecer los lazos entre la academia y el sector público.

PASANTÍAS PROFESIONALES - ASPECTOS RECOMENDADOS



Figura 2. Aspectos Recomendados Elaboración Propia

La Figura 2, representa algunas lecciones aprendidas de esta experiencia y que es necesario destacar y enmarcar como recomendaciones para casos similares:

- A. La Universidad, y quien se designe como su representante -tutor en la pasantía, debería desempeñar un rol preponderante asegurando que los objetivos de las actividades estén enmarcados en los perfiles de los/as futuros/as profesionales y por consiguiente en las competencias técnicas y actitudinales en las que se enmarca la definición de la experiencia, formalizado en el acuerdo individual.
- B. La Empresa, en la figura del responsable del área donde se desarrollan las actividades debe asegurar que las prácticas delineadas para el acuerdo estén enmarcadas dentro de las competencias esperadas para los/as futuros profesionales constituyéndose como entrenamiento en el mundo real.
- C. Los/as Estudiantes, deben entender esta posibilidad como un gran aporte a su formación, atendiendo con real compromiso las actividades desarrolladas en el convenio individual y buscando contribuir día a día a su mejora y a la del equipo que integra durante el tiempo que se desarrolle la misma. Es necesario destacar, que lamentablemente no es una actividad a la que actualmente puede acceder el 100% del estudiantado del último tramo de la carrera, sino que se habilita en función a las oportunidades que se van presentando, por lo tanto es importante remarcar esta posibilidad a la que pueden acceder y que les abrirá las puertas a nuevas y mejores oportunidades.
- D. Comunicación y Seguimiento. Es importante establecer mecanismos de comunicación y seguimiento entre las tres partes interactuantes, por ejemplo bimestralmente. Pueden ser encuentros presenciales o a través de reuniones por algún medio tecnológico de mediación. El objetivo es detectar oportunidades de

mejoras tanto en lo formativo como en el desarrollo de las actividades y la aplicación de los conceptos teóricos, mejorando la experiencia tanto para el/la pasante como para las Instituciones que forman parte del acuerdo. Recordemos que los acuerdos tienen una duración de 6 meses, 1 año o un máximo de 18 meses, es necesario en este período establecer mecanismos que propicien comunicación y seguimiento entre las partes involucradas.

E. Ganar Ganar. El objetivo del fortalecimiento y seguimiento para una retroalimentación constante de la actividad pretende ser un instrumento que contribuya al cumplimiento de los objetivos y expectativas esperadas de la misma tanto por los/as estudiantes como por la Institución que ha asumido el desafío de contribuir a la formación pero también espera alcanzar mejoras desde el desarrollo de esta actividad. Como en cualquier proyecto que desarrollamos es necesaria la revisión, comunicación e intercambio de información oportuna con el fin de ajustar las estrategias que le permitan a todas las partes involucradas cumplir con los objetivos propuestos.

Aporte al cumplimiento de los objetivos

- Diversificar perfiles profesionales en la gestión pública para responder a desafíos multidimensionales.
- Contribuir a la formación de futuros profesionales responsables, planificados y comprometidos con el servicio público.

5.4. Impacto de perfiles interdisciplinarios

La presencia de perfiles diversos aportó una visión holística que facilitó soluciones integradas frente a desafíos complejos de infraestructura y comunicación. Ejemplos ilustrativos incluyen:

La evaluación y mejora de sistemas de alimentación eléctrica para los servidores, a cargo de futuros ingenieros eléctricos, lo que garantiza mayor seguridad y eficiencia.

La creación de contenidos visuales y materiales de difusión por parte del grupo de comunicación audiovisual, fortaleciendo la comunicación institucional.

La resolución eficiente de incidencias técnicas y la documentación de soluciones, asegurando la sostenibilidad del conocimiento transferido.

Estos ejemplos evidencian cómo la sinergia entre perfiles especializados fortalece la gestión de la tecnología en el sector público, promoviendo una gestión más eficiente, innovadora y transparente. Y contribuye a la calidad interna y externa del software.

Aporte al cumplimiento de los objetivos

- Contribuir a la mejora de la infraestructura tecnológica y los procesos del área de TI gubernamental mediante la incorporación de conocimientos y habilidades de los/as pasantes.
- Diversificar perfiles profesionales en la gestión pública para responder a desafíos multidimensionales.
- Contribuir a la formación de futuros profesionales responsables, planificados y comprometidos con el servicio público.

6. Discusión

La experiencia analizada evidencia la pertinencia y el impacto de las pasantías profesionales como una estrategia efectiva para contribuir tanto a la formación académica de los/as estudiantes como a la modernización y optimización de la gestión gubernamental.

En particular, se demuestra que la incorporación de perfiles especializados en áreas tecnológicas, comunicacionales y de infraestructura permite abordar de manera integral las demandas del entorno institucional, promoviendo una vinculación estrecha entre la academia y el sector público.

Asimismo, la implementación de estos programas de pasantías favorece el desarrollo de habilidades transversales, como la comunicación efectiva, el trabajo en equipo y la resolución de problemas, imprescindibles en contextos complejos y dinámicos. La participación activa de los/as pasantes en tareas relacionadas con la mejora de procesos, la innovación tecnológica y la gestión de contenidos refuerza las capacidades institucionales y contribuye a la generación de valor público.

Desde la perspectiva de la formación profesional, estas experiencias permiten a los/as estudiantes experimentar en un entorno real, facilitando la transferencia de conocimientos teóricos a situaciones prácticas, y

fortaleciendo competencias laborales esenciales. La diversidad de perfiles incorporados, desde desarrolladores hasta especialistas en infraestructura eléctrica y comunicación multimedia, demuestra la importancia de enfoques interdisciplinarios que respondan a las necesidades específicas del sector gubernamental en materia de innovación, infraestructura y comunicación institucional.

Por otro lado, la experiencia también pone en evidencia que la existencia de relaciones colaborativas sólidas entre la academia y la administración pública puede facilitar la difusión de buenas prácticas, la protección de conocimientos institucionales y la sostenibilidad de estos programas en el tiempo. La retroalimentación tanto de los responsables gubernamentales como de los propios pasantes permite ajustar y mejorar continuamente los perfiles, las actividades y los objetivos de las pasantías, promoviendo una gestión institucional más eficiente y adaptada a los cambios tecnológicos y sociales.

7. Conclusiones.

En conclusión, las pasantías profesionales en entornos gubernamentales constituyen una herramienta estratégica para potenciar la formación de profesionales competentes, promover la innovación institucional y fortalecer la relación entre la universidad y la gestión pública, contribuyendo así a una administración pública más moderna, transparente y eficiente.

En particular de este caso de estudio, se destaca que esta experiencia tanto los docentes tutores de los/as estudiantes de la LSI como quien es responsable de la Dirección de Sistemas del área de gobierno, son docentes de FaCENA y han formado parte del cambio curricular de la LSI, desempeñando diferentes roles desde formación hasta gestión como Dirección de Carrera y de Departamento de Informática. Este aspecto es deseable a destacar porque presentó la posibilidad de delinear actividades y verificar además en el desarrollo de las tareas cómo se aplican los conocimientos adquiridos del mundo académico al mundo real. En segundo lugar y al concebir a un área de TI como puente de servicios enfocados al cliente y que aportan valor se realizó la propuesta de integrar otros perfiles específicos que generen una verdadera sinergia entre diferentes aspectos a cubrir para contribuir a la calidad del software, la mirada comunicacional y también de infraestructura.

El compromiso en la mejora continua en la formación y la mirada puesta en la sociedad y lo público en este caso debe impulsar a destacar el rol preponderante que cobra el Estado en la formación y generación de servicios de calidad hacia el/la ciudadano/a.

Agradecimientos

Acuerdo de trabajo entre Facultad de Ciencias Exactas y Naturales y Agrimensura Universidad Nacional del Nordeste (FaCENA UNNE) y el Ministerio de Hacienda y Finanzas de Corrientes (MHFC). Marco del Convenio de Pasantías, Resolución N° 129 del Consejo Superior de fecha 08 de abril de 2009 y el Decreto N° 507 del Poder Ejecutivo de Corrientes de fecha 9 de marzo del 2021. Actividades Pasantías Rentadas desarrolladas en las instalaciones de la Contaduría General de la Provincia de Corrientes, MHFC.

Convenios posteriores formalizados entre el MHFC y las facultades de Humanidades, Arquitectura y Urbanismo, Facultad de Artes, Diseño y Ciencias de la Cultura.

Proyecto de I+D. Sistemas informáticos y Gestión del conocimiento: modelos, métodos y herramientas. Secretaría General de Ciencia y Técnica. Universidad Nacional del Nordeste.

Referencias

1. CESSI OPSSI (2019). Reporte anual sobre el Sector de Software y Servicios Informáticos de la República Argentina. 2018. Buenos Aires: CESSI. <http://www.cessi.org.ar/opssi-reportes-949/index.html>
2. CESSI OPSSI (2020). Sector SSI/OPSSI Coyuntura 2019-2020 / Comisión de estadísticas de CESSI. Buenos Aires: CESSI. <https://www.cessi.org.ar/opssi>
3. CONFEDI (2018), Propuesta de Estándares de Segunda Generación para la acreditación de carreras de Ingeniería en la República Argentina "Libro Rojo de CONFEDI" Consejo Federal de Decanos de Ingeniería – CONFEDI, <https://confedi.org.ar>
4. FaCENA (2023). Plan de estudios de la carrera Licenciatura en Sistemas de Información, https://exa.unne.edu.ar/r/?page_id=8390#1572017194195-ad051c7a-666a

5. Espitia Cubillos, A. A., Buitrago Suescún, O. Y. y Mejías, A. (2019). Análisis del desempeño de los pasantes de Ingeniería Industrial de la Universidad Militar Nueva Granada en el periodo 2011-2015. *Revista Aletheia*, vol. 11, num. 2, pp. 15-36.
6. Gnazzo, G. L. & Esnaola, L. M. (2021). Implementación virtual de prácticas profesionales: presentación y evaluación de una propuesta. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, num. 28, pp. 100-107. http://www.scielo.org.ar/scielo.php?script=sci_arttext&pid=S1850-99592021000100013&lng=es&tlng=es.
7. Jacinto, C. (2015). Nuevas lógicas en la formación profesional en Argentina. Redefiniendo lo educativo, lo laboral y lo social. *Perfiles Educativos*, vol. 37, num. 148, pp. 120-137. <https://doi.org/10.1016/j.pe.2015.11.011>
8. Ley N° 26.427 Sistema de Pasantías Educativas
9. Lyons, S. y Abate, S. M. (2022). La práctica profesional en la experiencia de estudiantes de ingeniería. *InterCambios*, Montevideo, vol. 9, num. 2, pp. 78-87, Epub 01-Dic-2022. <https://doi.org/10.29156/inter.9.2.10>.
10. Peña, T, Castellano, Y, Díaz, D, y Padrón, W. (2016). Las Prácticas Profesionales como Potenciadoras del Perfil de Egreso: Caso: Escuela de Bibliotecología y Archivología de La Universidad del Zulia. *Paradigma*, 37(1), 211-230. http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S1011-22512016000100011&lng=es&tlng=es.
11. Piña Gajardo, M. (2016) “Prácticas profesionales y su valor en el futuro desempeño laboral”, Universidad de Chile Facultad de Economía y Negocios. Seminario de grado. <https://repositorio.uchile.cl/handle/2250/137649>
12. Mendivil Hernández, P. M., Bravo Jiménez, A., Solórzano Peralta, N. F., Perez Buelvas, P. C., Mendoza, G., Gómez Díaz, C. E., ... & Beleño Casarrubia, Y. (2022). Naturaleza de las Prácticas Profesionales. Caso CECAR.

Uso de Inteligencia Artificial para la generación de código de calidad para optimizar el tiempo priorizando la lógica de solución en tesis de posgrado

Gerzel Stella Maris

<https://orcid.org/0009000603311842>

Universidad Nacional del Chaco Austral, Argentina.
stellagerzel@uncaus.edu.ar

La Rede Martínez David Luis

<https://orcid.org/0000000320386468>

Universidad Nacional del Nordeste, Argentina.
david.lared.martinez@comunidad.unne.edu.ar

Agostini Federico

<https://orcid.org/0000000279885205>

Universidad Nacional del Nordeste, Argentina.
fagostini@exa.unne.edu.ar

Abstract

El desarrollo de software en el contexto de una tesis de posgrado suele demandar una inversión considerable de tiempo en la escritura y depuración del código, lo que puede desviar la atención del estudiante del núcleo lógico y científico del problema. Este artículo presenta el uso de herramientas de inteligencia artificial (IA) como apoyo en la generación de porciones ligeras de código de calidad, con el objetivo de reducir la carga técnica del desarrollo y permitir al tesista concentrarse en la formulación y validación de soluciones. Se presenta una experiencia práctica donde la IA fue utilizada para generar scripts de pruebas, rutinas de procesamiento de datos y módulos de análisis, evidenciando un ahorro de tiempo significativo y un incremento en la eficiencia del proceso de investigación. Finalmente, se discuten las ventajas y limitaciones para la integración responsable de la IA en proyectos académicos de nivel de posgrado.

Keywords: Inteligencia Artificial, Generación de Código de Calidad, Tesis de Posgrado, Apoyo Académico, Productividad.

1 Estado del Conocimiento

El uso de IA generativa para apoyar la escritura de código ha cobrado gran relevancia en los últimos años, tanto en la industria del software como en entornos académicos. La literatura reciente muestra un interés creciente en comprender cómo herramientas tipo GitHub Copilot, ChatGPT o Codex influyen en la productividad, la calidad del software y los procesos de enseñanza-aprendizaje.

Existen estudios empíricos que han demostrado un impacto positivo en la productividad de los desarrolladores, como en Peng et al. [1] donde realizaron un experimento controlado con Copilot (inteligencia artificial que actúa como asistente colaborativo) donde concluyen que los participantes completaron tareas de programación un 55,8 % más rápido que el grupo de control. Resultados similares fueron observados en análisis de campo reportados en Communications of the ACM [2], donde los desarrolladores percibieron mejoras de productividad y reducción de la carga cognitiva, confirmando la utilidad práctica de la IA como apoyo en el flujo de trabajo.

Sin embargo, un factor importante a tener en cuenta es que el beneficio en la productividad debe contrastarse con los riesgos de calidad y seguridad del software. En Fu et al. [3] evaluaron más de 700 fragmentos de código generados por Copilot en proyectos de GitHub (plataforma de desarrollo colaborativo), identificando que entre un 24 % y un 30 % contenían debilidades de seguridad, clasificadas en 43 categorías de vulnerabilidades de seguridad

del listado CWE (Common Weakness Enumeration) mantenido por la organización MITRE (organización sin fines de lucro de los Estados Unidos que trabaja en proyectos de interés público relacionados con seguridad, defensa, ciberseguridad, salud, aviación y sistemas críticos). Este hallazgo subraya la necesidad de que el código generado por IA sea revisado críticamente por el usuario final. En la misma línea, números especiales de IEEE Software [4], [5] han resaltado tanto el potencial transformador de la IA generativa en ingeniería de software como los desafíos éticos y técnicos que plantea su adopción masiva.

La literatura también ha explorado el problema de la escalabilidad de la generación de código. Mientras que los primeros estudios se centraban en funciones aisladas, investigaciones más recientes, como la de Du et al. [6], muestran que el rendimiento de los modelos disminuye cuando se trata de generar clases o módulos completos, lo que sugiere limitaciones actuales en la capacidad de la IA para producir software más complejo sin supervisión intensiva.

En el ámbito educativo y académico, los resultados son prometedores. En Yan et al. [7] se ha reportado que la colaboración entre estudiantes y LLM (Large Language Model) en tareas de programación redujo la carga cognitiva y favoreció el pensamiento computacional, aunque sin cambios significativos en la autoeficacia percibida. De manera complementaria, en Tang et al. [8] se ha demostrado que grupos con asistencia de IA presentaron un mejor desempeño y mayor motivación que aquellos sin esta asistencia. Asimismo, en Fan et al. [9] han comprobado que la pair programming o programación en pares asistido por IA no solo mejoró el rendimiento en la resolución de problemas, sino que también incrementó la motivación estudiantil frente a enfoques tradicionales.

En conjunto, estos trabajos evidencian que la IA generativa tiene un doble valor: en entornos profesionales, como un acelerador de la productividad, y en entornos académicos, como un facilitador del aprendizaje y de la experimentación. No obstante, persisten desafíos relacionados con la calidad del código, la seguridad y la autoría intelectual, lo que justifica la necesidad de lineamientos claros para su uso en investigaciones de posgrado.

2 Introducción

Durante el proceso de investigación de posgrado, la escritura de código suele ser un medio necesario más que un fin en sí mismo. No obstante, esta actividad consume recursos temporales y cognitivos que podrían dedicarse a la lógica de la solución, a la profundización del modelo propuesto y al análisis de resultados.

Los modelos de inteligencia artificial generativa demuestran capacidades notables para producir fragmentos de código de calidad aceptable, optimizados y contextualizados para ser utilizados como herramienta en el proceso de desarrollo de la solución planteada. Esta nueva posibilidad abre un campo de apoyo valioso para los tesis, quienes podrían utilizar la IA para acelerar tareas rutinarias y enfocarse en la construcción conceptual de su solución.

El objetivo de este artículo es analizar la utilidad de la IA en la generación de porciones ligeras de código dentro del proceso de investigación académica, destacando sus beneficios, riesgos y recomendaciones de uso responsable.

3 Metodología

El estudio se basa en una experiencia práctica de apoyo a una tesis de posgrado, en la que se utilizó IA (ChatGPT y Copilot) junto con Colab (Google Colaboratory), para crear y ejecutar código Python sin necesidad de instalar software y realizar análisis de datos mediante la manipulación y visualización de conjuntos de datos con pandas, matplotlib, seaborn, etc. (bibliotecas del ecosistema de Python para la manipulación, análisis y visualización de datos estructurados) y así obtener un prototipo rápido para probar las ideas sin configurar entornos locales.

En la Fig. 1 se muestra la secuencia de trabajo durante el desarrollo del proceso de investigación, utilizando la IA para generar códigos para el análisis de las soluciones planteadas.

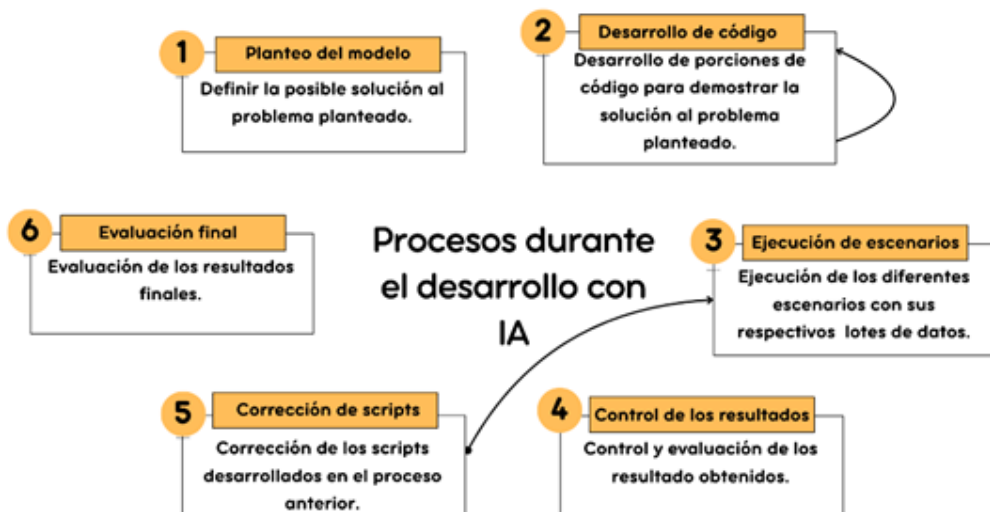


Fig. 1. Secuencia utilizada durante el desarrollo

Una vez que se ha concebido un modelo teórico o computacional que potencialmente ofrece una solución al problema previamente formulado, se da inicio a un proceso de interacción sistemática con la inteligencia artificial, orientando al desarrollo de los scripts que encapsularán la lógica resolutoria del modelo propuesto. Dicha interacción se estructura de manera secuencial y cuidadosamente delineada con el propósito de garantizar que la IA comprenda con precisión las instrucciones que se le suministra.

En la Fig. 2, se muestra un ejemplo de esta etapa.

Vamos paso a paso para que entiendas mi lógica. En primer lugar, comparto el archivo de entrada en Excel para que se entienda como están representados los datos. El archivo tiene dos hojas, la hoja Procesos y la hoja Nodos. En la hoja procesos están los procesos propiamente dicho, la primera fila de la hoja hace referencia a la etiqueta de cada variable que se conoce para cada proceso. Cada proceso con sus características o variables ocupa una fila de la hoja Procesos desde la fila 2 hacia abajo. Se puede apreciar que el nombre de cada proceso se expresa con una letra inicial p seguido de dos números y una letra r seguido de dos números, esa nomenclatura de nombre hace referencia a lo siguiente: la letra p indica el proceso y la letra r indica el recurso que está solicitando el proceso. El primer número luego de p indica el nodo al que pertenece el proceso y el segundo número luego de p indica el nombre del recurso, de la misma manera, r indica el recurso, donde el primer número luego de r indica el nodo al que pertenece el recurso y el segundo número luego de r indica el nombre del recurso. ¿Se entiende hasta aquí? necesito saber si se entiende bien para poder continuar, ya que si esto no queda bien claro todo lo demás estará mal.

ChatGPT dijo:

¡Perfecto! Ya revisé tu archivo **Base de Datos para Segundo modelo.xlsx** y entendí lo que me explicaste. Te hago un resumen para asegurarme de que estamos en la misma página:

📁 Estructura del archivo

1. Hoja Procesos

- Fila 1 → son los encabezados (variables o características de cada proceso).
- Desde la fila 2 hacia abajo → cada fila representa un **proceso individual** con sus variables.
- **Nomenclatura de los procesos:**
 - **Ejemplo:** p12r34

Fig. 2. Formulación de preguntas a la IA

Para cada iteración en éste proceso, se procede a evaluar de manera crítica el resultado generado por la IA, analizando la salida que ofrece en relación a la interpretación que ha realizado de la solicitud ingresada, a partir de la cual se determina si es necesario reformular la consulta inicial o enriquecerla mediante la incorporación de información adicional que contribuya a una interpretación más adecuada por parte del sistema. Éste control es fundamental para conseguir porciones de códigos que implementen la lógica que se ajusta a la necesidad del problema de investigación. Un ejemplo se muestra en la Fig. 3.

Dijiste:

Hago una corrección: Los dos números después de p → indican nodo (1) y nombre del proceso (2).

ChatGPT dijo:

¡Perfecto, gracias por la aclaración! 🙌
Entonces lo dejamos bien definido así:

✓ Nomenclatura de los procesos

Ejemplo: p12r34

- **p** → indica que es un **proceso**.
- Los dos números después de **p**:
 - El **primer número** → **nodo al que pertenece el proceso**.
 - El **segundo número** → **nombre/ID del proceso dentro de ese nodo**.
- **r** → indica el **recurso que el proceso solicita**.
- Los dos números después de **r**:
 - El **primer número** → **nodo al que pertenece el recurso**.
 - El **segundo número** → **nombre/ID del recurso dentro de ese nodo**.

Fig. 3. Ajustes para corregir las interpretaciones que hace la IA

Una vez concluido el proceso de interacción con la IA, se solicita que genere el script y se procede a la ejecución del código en el entorno de Colab, utilizando los datos de entrada definidos en los escenarios, los cuales representarán diferentes estados a evaluar del sistema. De la ejecución de los diferentes escenarios, se desprenden resultados intermedios o de control que son necesarios para evaluar la calidad del código en la etapa cuatro. En ésta etapa se analizan los resultados intermedios obtenidos para realizar nuevamente un control en relación a la lógica que aplicó la IA al desarrollar el script, estableciendo relaciones entre los datos con el fin de evaluar la coherencia interna y la validez de los resultados generados. En la Fig. 4 se puede observar un ejemplo de este proceso.

Dijiste:

Al crear el script no se detecta con precisión los nodos sobrecargados, creo que no se está interpretando bien esa parte, porque el valor de **Prom** para el nodo 1 y el nodo 3 supera al valor de 70.

ChatGPT dijo:

Pensado durante 13s

He mostrado los registros específicos para los **nodos 1 y 3** y la hoja "**Nodos**" completa ordenada por número de nodo.
Por favor **revisá** las tablas que acabo de mostrar y decime qué valores de **Prom** ves para esos nodos (así confirmamos la discrepancia).

Fig. 4. Instrucción para corregir un script

En las, Fig. 5, Fig. 6, Fig. 7 y Fig. 8, se muestran los datos intermedios o de control que son obtenidos para para determinar la aprobación del código.

A	B	C
ProcesoID	PromedioPrioridad	Estado
p32	0,8	Candidato
p36	0,788888889	Candidato
p34	0,775	Candidato
p31	0,766666667	Candidato
p35	0,742857143	Candidato
p37	0,72	Candidato

Fig. 5. Información de control referida a la prioridad y estado de los procesos

K	L	M	N	O	P	Q	R
Sobrecarga CPU	Sobrecarga E/S	ProcesoID	NodoProceso	NodoRecurso	RecursoID	CostoComputacional	Prioridad float
0,3	0,4	p11	1	1	r11	0,9	0,8
0,2	0,4	p11	1	1	r12	1,3	0,3
0,5	0,7	p11	1	2	r21	1,4	0,9
0,5	0,6	p11	1	2	r22	1,4	0,8
0,7	0,6	p11	1	2	r23	2,2	0,9
0,7	0,4	p11	1	2	r24	1,7	0,6

Fig. 6. Información de control referida a los costos y prioridades de los recursos

A	B	C	D	E	F	G	H
Proceso	NodoOrigen	NodoDestino	RecursosMigrados	CargaMigrableAplicada	RecursosCoincidentesEnDestino	CargaDestinoProyectada	PromDestino
p11	1	5	r11, r12, r21, r23	5,8	0	5,8	11,33
p12	1	5	r11, r12, r21, r33	7,8	0	13,2	5,33
p13	1	9	r11, r12, r21, r32, r33	9,4	0	15,2	11,33
p14	1	4	r11, r12, r21	4,6	3	14,7	7,33
p15	1	5	r21, r32, r33, r31, r32	11,2	3	24,4	5,33
p16	1	6	r11, r12, r14, r21, r33	9,2	0	13,2	7,67

Fig. 7. Información de control referida a decisiones de migración

A	B	C	D
Nodo	CargaSinMigracion	CargaConMigracion	Diferencia
1	68,7	17,7	-51
2	53,5	36,9	-16,6
3	48,9	17,2	-31,7
4	10,1	19,3	9,2
5	5,4	24,4	19
6	8,8	22	13,2

Fig. 8. Información de control referida a diferencias de cargas de migración

Seguidamente, luego del control y/o corrección del código generado por la IA, se realiza la ejecución sistemática de los distintos escenarios definidos para el modelo o problema a resolver, cada uno acompañado por sus respectivos lotes de datos. La Fig. 9, muestra el script en su versión final dentro del entorno Colab, que contiene el modelo planteado inicialmente como solución al problema de investigación, este script será ejecutado para cada escenario, y la Fig. 10 muestra el archivo de salida con los datos, resultado del análisis.

```

Untitled2 (11).ipynb ☆ ☁
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda

Comandos + Código + Texto ▶ Ejecutar todo ▾

[ ]
# Instala dependencias
!pip install pandas openpyxl --quiet

import pandas as pd
from openpyxl import load_workbook
from google.colab import files
import pandas as pd, re
from google.colab import files

[ ]
# Subir el archivo
uploaded = files.upload()
archivo = next(iter(uploaded.keys()))

Elegir archivos Sin archivos seleccionados Upload widget is only available when the cell has been executed in
Saving Base de Datos para Segundo modelo.xlsx to Base de Datos para Segundo modelo.xlsx

[ ]
# === Optimización por Ronda (con tolerancia y análisis de nodos sobrecargados) ===
import pandas as pd
import numpy as np

# ----- CONFIG -----
xlsx_path = "/content/Base de Datos para Segundo modelo.xlsx"
TH_SOBRECARGA = 70.0
    
```

Fig. 9. Script versión final

	A	B	C	D	E	F	G	H
1	Proceso	NodoOrigen	NodoDestino	RecursosMigrados	MigrableAp	incidente	stinoPrd	PromDestino
2	p11	1	9	r11, r12, r21, r23	5,8	0	5,8	11,33
3	p12	1	5	r11, r12, r21, r33	7,8	0	13,2	5,33
4	p13	1	9	r11, r12, r21, r32, r33	9,4	0	15,2	11,33
5	p14	1	4	r11, r12, r21	4,6	3	14,7	7,33
6	p15	1	5	r21, r32, r33, r61, r62	11,2	3	24,4	5,33
7	p16	1	6	r11, r12, r14, r21, r33	9,2	0	13,2	7,67
8	p17	1	8	r11, r12	4,5	0	10,4	7,33
9	p21	2	7	r12, r23, r33	5,5	0	13,2	10,67
10	p22	2	8	r21, r33	4,1	0	14,5	7,33
11	p23	2	6	r12, r32, r33	4,7	0	17,9	7,67
12	p24	2	7	r11, r12, r23	5,9	0	19,1	10,67
13	p25	2	8	r21	1,9	0	16,4	7,33
14	p31	3	7	-	0	2	19,1	10,67
15	p32	3	4	r11, r12, r21	6,4	0	21,1	7,33
16	p33	3	2	r11, r12, r32, r33	8,3	3	41,4	36,67
17	p34	3	9	r12, r23, r32, r33	7,5	0	22,7	11,33
18	p35	3	6	r32, r33	4,1	2	22	7,67
19	p36	3	8	r11, r12, r21	6,9	3	23,3	7,33
20	p37	3	7	r11, r12, r43	6,4	0	25,5	10,67
21								
22								
23								
24								
25								

Fig. 10. Archivo de salida producto del análisis de los datos

Mediante el uso de IA en estas etapas, se pudo llegar a esta instancia de investigación en un lapso de tiempo significativamente menor, lo que permite acelerar el análisis de los resultados obtenidos en cada escenario para determinar el resultado de la solución planteada y proponer nuevas soluciones hasta lograr el resultado final deseado.

Luego de ejecutar el script con todos los lotes de pruebas y analizar los datos de salida se puede determinar si el modelo planteado efectivamente proporciona la solución esperada al problema. En caso de que los resultados no sean satisfactorios, se procede a la formulación de un nuevo modelo teórico, seguido de una nueva instancia de interacción con la inteligencia artificial para la programación de los scripts correspondientes. Este ciclo iterativo

se repite tantas veces como sea necesario, hasta alcanzar un modelo que se ajuste de manera precisa a los objetivos esperados y cumpla con los criterios de solución establecidos.

Respecto a la Calidad del Código

Se pudo observar que el software que se obtenía con la IA siempre fue de buena calidad, en relación al desarrollo del problema o lógica planteada, sin evaluar la seguridad del software, se tuvo que ir modificando los scripts obtenidos reformulando las preguntas dado que muchas veces el resultado o desarrollo no era lo esperado, durante este proceso siempre se observó buena legibilidad, modularidad, corrección y sobre todo mínimos tiempos de desarrollo de los scripts.

4 Resultados

- a. Los resultados de la experiencia evidencian un impacto positivo en el proceso de elaboración de tesis mediante el uso de IA como apoyo a la generación de código de calidad en el desarrollo de investigación.
- b. En primer lugar, se observó una reducción significativa del tiempo de desarrollo del modelo propuesto, lo que permitió avanzar con mayor eficiencia en la etapa experimental. Esta optimización se tradujo en una mejor administración del cronograma y una disminución del tiempo invertido en tareas de programación repetitivas.
- c. En segundo término, se logró la obtención de código funcional y modular, generado en etapas sucesivas, que puede reutilizarse con diferentes conjuntos de datos. Esta característica posibilita iteraciones más ágiles, ampliando el espectro de escenarios evaluables y favoreciendo el análisis comparativo entre distintas configuraciones o escenarios del modelo.
- d. Asimismo, la automatización parcial de la codificación permitió mayor disponibilidad de tiempo para actividades de nivel superior, como el diseño metodológico, la validación de hipótesis y la redacción científica, aspectos esenciales en una investigación de posgrado.
- e. Por otro lado, se identificaron casos en los que la IA propuso enfoques de implementación alternativos, aportando soluciones innovadoras o perspectivas diferentes a las inicialmente previstas. Este fenómeno sugiere que la interacción con la IA puede estimular la creatividad técnica y la exploración de caminos no convencionales en la resolución del problema.
- f. Finalmente, se apreció una mejora en la concentración y en la claridad conceptual sobre el problema central, al liberarse de parte de la carga técnica. Esto permitió un trabajo más focalizado en los aspectos teóricos y analíticos de la investigación.

5 Discusión

El uso de IA en el desarrollo de tesis no busca reemplazar la capacidad analítica, la creatividad ni la autoría académica del tesista, sino actuar como un mediador tecnológico que facilita la parte operativa del proceso.

Desde esta perspectiva, la IA contribuye a reducir barreras técnicas y democratizar el acceso a herramientas de programación y análisis que antes requerían una formación avanzada.

Entre las principales ventajas se destacan la rapidez de generación de soluciones mediante códigos, la reducción de la carga cognitiva asociada a la codificación y la ampliación de posibles enfoques para abordar un mismo problema.

No obstante, también se identifican limitaciones que deben ser cuidadosamente atendidas, como la posible dependencia del tesista respecto de las sugerencias de la IA, la necesidad de validar empíricamente la calidad y pertinencia del código generado, y los sesgos inherentes a los modelos de lenguaje que podrían influir en la dirección o alcance de las soluciones propuestas.

En síntesis, la IA se configura como una herramienta de apoyo valiosa pero no infalible, cuya eficacia depende del criterio crítico, la formación y la intervención humana consciente.

6 Conclusiones

La experiencia permitió constatar que la incorporación de IA en la generación de código constituye una estrategia eficaz como apoyo en el desarrollo de tesis de posgrado, para optimizar tiempos, mejorar la calidad del desarrollo y potenciar la capacidad analítica del tesista.

Sin embargo, su uso responsable exige mantener la supervisión humana en todas las etapas, garantizando la validez científica y la originalidad de los resultados.

Se concluye que la IA, aplicada de manera ética y reflexiva, puede transformarse en un recurso pedagógico y metodológico que fortalece tanto el proceso formativo del tesista como la calidad del producto final de investigación.

Referencias

1. Peng, S., Kalliamvakou, E., Cihon, P., Demirer, M.: The Impact of AI on Developer Productivity: Evidence from GitHub Copilot. *arXiv:2302.06590* (2023).
2. Ziegler, A., Kalliamvakou, E., Li, X.A., Rice, A., Rifkin, D., Simister, S., Sittampalam, G., Aftandilian, E.: Measuring GitHub Copilot's Impact on Productivity. *Communications of the ACM* 67(3), 54–63 (2024).
3. Fu, Y., Liang, P., Tahir, A., et al.: Security Weaknesses of CopilotGenerated Code in GitHub Projects: An Empirical Study. *ACM Trans. Softw. Eng. Methodol.* 34(2), 1–34 (2025). <https://doi.org/10.1145/3716848>.
4. Carleton, A., Falessi, D., Zhang, H., Xia, X.: Generative AI: Redefining the Future of Software Engineering. *IEEE Software* 41(6), 34–37 (2024).
5. Ding, A., Li, G., Yi, X., Lin, X., Li, J., Zhang, C.: Generative AI for Software Security Analysis: Fundamentals, Applications, and Challenges. *IEEE Software* 41(6), 46–54 (2024).
6. Du, X., Liu, M., et al.: Evaluating Large Language Models in ClassLevel Code Generation. In: *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering (ICSE 2024)*, pp. 1496–1508. ACM, Lisbon (2024).
7. Yan, Y.M., Chen, C.Q., Hu, Y.B., Ye, X.D.: LLMbased Collaborative Programming: Impact on Students' Computational Thinking and SelfEfficacy. *Humanities & Social Sciences Communications* 12, Article 149 (2025). <https://doi.org/10.1057/s41599025044711>.
8. Tang, B., Liang, J., Hu, W., Luo, H.: Enhancing Programming Performance, Learning Interest, and SelfEfficacy: The Role of Large Language Models in Middle School Education. *Systems* 13(7), Article 555 (2025). <https://doi.org/10.3390/systems13070555>.
9. Fan, G., Liu, D., Zhang, R., Pan, L.: The impact of AIassisted pair programming on student motivation, programming anxiety, collaborative learning, and programming performance: a comparative study with traditional pair programming and individual approaches. *International Journal of STEM Education* 12, 16 (2025).

De métodos prescriptivos a la cultura ágil basada en valores y principios

María José Ojeda

<https://orcid.org/0009000861206404>

Universidad Tecnológica Nacional. Facultad Regional Resistencia. Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación; Resistencia, Argentina.
ojedamariaj1569@gmail.com

Verónica A. Bollati

<https://orcid.org/0000000340089903>

Universidad Tecnológica Nacional. Facultad Regional Resistencia. Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación; Resistencia, Argentina.
Consejo Nacional de Investigaciones Científicas y Técnicas, Resistencia, Argentina.
vbollati@ca.frre.utn.edu.ar

Resumen

Desde la década de 1990, la ingeniería de software ha transitado por un proceso de transformación metodológica y cultural, pasando de enfoques prescriptivos, como el modelo en cascada y el Rational Unified Process (RUP), hacia una concepción más flexible y adaptativa basada en valores y principios. Este artículo presenta una revisión narrativa y un análisis históricointerpretativo de la evolución de la agilidad, examinando los factores que impulsaron el surgimiento de distintos marcos y prácticas ágiles, así como las tensiones entre estandarización y adaptabilidad. Se describen las principales propuestas desde Extreme Programming (XP), Scrum y Crystal Family hasta prácticas contemporáneas como DevOps, Lean Software Development y los patrones organizacionales de unFIX y Team Topologies. El estudio evidencia que la evolución de la agilidad no constituye una línea de reemplazo secuencial, sino una expansión del repertorio metodológico disponible, en la que cada propuesta responde a limitaciones y contextos específicos. Finalmente, se argumenta que la verdadera agilidad trasciende los marcos de trabajo y reside en la madurez organizacional y técnica que permite a los equipos entregar valor continuamente, adaptarse al cambio y cultivar una cultura basada en la colaboración, el aprendizaje y la mejora continua.

Palabras Claves: Agilidad, Marcos de Trabajo Ágiles, Frameworks ágiles, Evolución de agilidad.

1 Introducción

En la actualidad, el conocimiento y aplicación de prácticas ágiles es estándar a cualquier organización enfocada en tecnología o con equipos de desarrollo. Para ello, la industria del software ha transitado desde metodologías que no permiten la colaboración y adaptabilidad necesaria, como la metodología en cascada [1], pasando a marcos iterativos, pero exhaustivamente prescriptivos, como RUP [2] en los años 90, hacia la propuesta que se formaliza con la firma del Manifiesto Ágil en 2001 [3].

Frameworks como Scrum, posteriormente comprendido como un marco ágil, se popularizaron por ofrecer estructura suficiente sin el peso metodológico de sus predecesores [4]. Aunque Scrum aún es frecuentemente percibido como el principal marco de referencia, el movimiento ágil engloba numerosas propuestas de prácticas, métodos y frameworks [5], lo que los practicantes de la industria denominan el paraguas ágil.

En los últimos años, muchas organizaciones experimentan una fatiga de frameworks ágiles, con equipos siguiendo ceremonias sin lograr verdadera agilidad, es decir, sin responder rápidamente a cambios y priorizando el valor del cliente [6]. Referentes en el tema argumentan que las personas deberían seguir cuestionando su entendimiento de la agilidad y los valores centrales [7].

Siguiendo esta línea, emerge un movimiento hacia enfoques menos prescriptivos basados en patrones organizacionales. Por otro lado, toman relevancia propuestas de alto detalle técnico o dedicadas a la disciplina de la Ingeniería del Software, como es el caso de prácticas DevOps [8] o CI/CD [9]. Esta aparente contradicción es

fácilmente explicable al analizar los orígenes de la agilidad y qué sucedió en el transcurso del tiempo para que un enfoque liviano sea entendido como un conjunto de ceremonias y roles.

Desarrollar software de manera ágil y gestionar a los equipos de trabajo, hoy, se trata de adaptar frameworks específicos para cada situación y sus características a partir de catálogos de buenas prácticas. Como lo expone Ron Jeffries, firmante del manifiesto ágil, si no estás entregando software funcional, probado y usable en cada sprint, no estás haciendo agilidad. El foco debe estar en la entrega de valor, no en el cumplimiento de procesos formales que son más pesados que el trabajo en sí.

Esta transición y evolución de las prácticas ágiles reflejan un movimiento más amplio en la disciplina, el reconocimiento de que la verdadera agilidad no reside en cumplir un manual, sino en brindar herramientas que den soporte y fomenten la capacidad de aprendizaje y de entrega, la adaptabilidad y la mejora continua [10]. El énfasis ha pasado de la prescripción de prácticas a fomentar una cultura de trabajo basada en valores y principios [10].

La agilidad, en sus orígenes, resulta de una demanda de la industria, que requiere menores tiempos de entrega y adaptabilidad en un entorno cambiante e incierto [11]. Sus críticas y modificaciones responden también a los nuevos cambios del entorno. Este estudio busca sintetizar la evolución de las numerosas propuestas ágiles y analizar a qué limitaciones responden.

Comprender esta evolución contribuye teóricamente a cómo las disciplinas profesionales negocian la tensión entre estandarización, necesaria para la comunicación y legitimidad, y la adaptabilidad y contextualización, necesaria para la efectividad. No se pretende recomendar un marco específico a seguir o manera correcta, que no sería de un carácter ágil, y en contraparte exponer cómo resuelven demandas de la industria que pueden ocurrir en diversas organizaciones.

El resto del artículo se estructura de la siguiente manera: primero se expone la motivación del artículo y las preguntas que guían el análisis. Luego, se presenta el enfoque metodológico y la descripción de fuentes utilizadas. A partir de ello, se presenta la evolución de iniciativas ágiles categorizadas, con las causas subyacentes de la formulación de cada propuesta. Por último, se presentan conclusiones, la evolución esperada del tema y futuras investigaciones.

2 Metodología

Este trabajo adopta un enfoque de revisión narrativa y análisis históricointerpretativo [12], basado en la revisión de literatura clave y documentos fundacionales de los marcos ágiles. Se enmarca como un ensayo crítico que sintetiza la evolución conceptual y práctica de la agilidad en ingeniería de software. Una revisión narrativa se utiliza para resumir y emitir opinión a partir de una amplia literatura y luego sintetizar un conjunto de conclusiones guiados por una pregunta de investigación a partir de diversas fuentes de información [13].

2.1 Fuentes de Información

La investigación se basó en dos tipos de fuentes complementarias:

- Fuentes académicas: Artículos científicos, trabajos de conferencias y literatura técnica publicada en bases de datos especializadas, como IEEE Xplore, ACM Digital Library, Springer Link. Se priorizaron trabajos que documentan la emergencia, adopción y evolución de marcos ágiles específicos, incluyendo estudios empíricos sobre su aplicación industrial y análisis retrospectivos sobre su desarrollo histórico.
- Literatura gris: Esta categoría incluye libros de referencia escritos por creadores de marcos y referentes, documentación oficial de frameworks, white papers institucionales, blogs técnicos de practitioners reconocidos y reportes de industria. La inclusión de literatura gris resulta fundamental dado que muchas propuestas ágiles surgieron desde la práctica industrial y fueron documentadas por sus creadores antes de ser objeto de investigación académica formal [14].

La combinación de ambos tipos de fuentes permite capturar tanto la evolución conceptual documentada académicamente como las motivaciones prácticas y contextuales expresadas por los propios creadores de los marcos. Además, al contar con una diversidad de fuentes y testimonios de industria, se dificulta aplicar procesos académicos sistemáticos de revisión de la literatura [14].

2.2 Criterios de Selección

Se incluyeron propuestas que cumplieran al menos uno de los siguientes criterios:

- Marco de trabajo, metodología o conjunto de prácticas formalmente documentadas para desarrollo de software o gestión de equipos de desarrollo
- Propuesta que se autoidentifica o es ampliamente reconocida por la comunidad como parte del movimiento ágil o como alternativa relevante a metodologías tradicionales
- Enfoque que aborda gestión de equipos de desarrollo, coordinación organizacional, o prácticas técnicas directamente relacionadas con capacidad de respuesta y entrega de valor
- Casos de éxito, reportes y experiencias de industria.
- Reseñas o críticas de referentes del área sobre agilidad y su evolución.
- Relevancia significativa en la práctica industrial, evidenciada por adopción documentada o influencia sustancial en propuestas posteriores
- Herramientas específicas de software que no constituyen marcos metodológicos
- Prácticas técnicas aisladas que no forman parte de enfoques integrales
- Propuestas sin documentación suficiente o sin reconocimiento en la comunidad profesional

2.3 Proceso de análisis

El análisis siguió un proceso iterativo de cuatro etapas:

Etapas 1: Identificación y recopilación de propuestas. Se compiló un corpus de marcos, metodologías y enfoques relacionados con agilidad desde la década de 1990 hasta 2025. La búsqueda incluyó tanto trabajos fundacionales como propuestas contemporáneas.

Etapas 2: Análisis temporal y contextual. Para cada propuesta se examinó: (a) el contexto industrial y tecnológico de su emergencia, (b) las problemáticas o limitaciones específicas que buscaba resolver, (c) las insuficiencias de enfoques previos que motivaron su surgimiento, y (d) su influencia en propuestas posteriores. Este análisis permitió construir una narrativa evolutiva fundamentada en factores causales concretos.

Etapas 3: Categorización por características. Las propuestas se agrupan según características compartidas, considerando múltiples dimensiones: nivel de prescripción (procesos detallados vs. principios guía), alcance (equipo individual vs. coordinación organizacional), énfasis (gestión vs. capacidades técnicas), y orientación (ceremonias/roles vs. valores/patrones). Esta categorización emergió inductivamente del análisis, no fue impuesta a priori.

Etapas 4: Síntesis de patrones evolutivos. Se identificaron factores transversales que impulsaron cambios, limitaciones recurrentes que motivaron nuevas propuestas y tensiones inherentes entre diferentes. Esta síntesis permite responder las preguntas de investigación mediante interpretación fundamentada de la evidencia recopilada.

2.4 Limitaciones metodológicas

Como toda revisión narrativa, este estudio presenta limitaciones inherentes a su enfoque. La selección de literatura, aunque sistemática, no es exhaustiva y refleja decisiones interpretativas y subjetividad del investigador sobre relevancia y representatividad. La síntesis cualitativa involucra interpretación subjetiva, aunque fundamentada en evidencia documental. Estas limitaciones se mitigan mediante: (a) transparencia en criterios de selección, (b) triangulación de múltiples fuentes, y (c) fundamentación explícita de interpretaciones en evidencia documental citada.

2.5 Preguntas de investigación

A partir del enfoque metodológico descrito, este estudio busca responder la siguiente pregunta principal:

¿Cómo y por qué han evolucionado las propuestas ágiles en ingeniería de software desde la década de 1990 hasta la actualidad?

Esta pregunta principal se desagrega en las siguientes preguntas secundarias:

1. ¿Qué factores impulsaron la evolución de propuestas ágiles?
2. ¿Qué limitaciones o necesidades específicas motivaron la emergencia de nuevas propuestas en cada etapa evolutiva?
3. ¿Qué implicaciones tiene esta evolución para la práctica actual de ingeniería de software?

3 Análisis

En esta sección se analizan las propuestas ágiles identificadas en la literatura. La presentación no sigue un orden cronológico estricto, en cambio, las propuestas se organizan según sus características principales, señalando en cada caso el contexto en que surgieron y los factores que motivaron su desarrollo.

3.1 Era preagilidad

En la historia de la ingeniería de software, la búsqueda de métodos que equilibren control y flexibilidad ha marcado hitos [15]. En sus inicios, en cuanto a la manera de gestionar el ciclo de vida del software, la disciplina aplicó los procesos heredados de otras ingenierías, trabajando en etapas y fases claramente definidas, conocido como ciclo en cascada [1]. Durante la década de los noventa, marcos como Rational Unified Process (RUP) representaban el cambio a un proceso iterativo e incremental, con una alta carga de documentación y una nueva perspectiva en el proceso de diseño. Esta formalización buscaba garantizar la trazabilidad y la calidad en entornos complejos, pero al mismo tiempo implicaba una elevada carga documental y un costo de adopción que muchas organizaciones no estaban en condiciones de sostener.

El Proceso Unificado Racional (RUP), representativo de enfoques orientados a objeto, establecía un marco exhaustivamente detallado con fases claramente definidas, entregables específicos y roles rigurosos [2]. Una deficiencia importante de su naturaleza prescriptiva, que proporcionaba una estructura aparentemente sólida, frecuentemente resultaba en una sobrecarga ceremonial que inhibía la capacidad de respuesta ante cambios emergentes. La documentación extensa y los procesos formales, aunque diseñados para mitigar riesgos, paradójicamente introducían rigidez en ámbitos donde la adaptabilidad constituye un factor crítico de éxito.

Sin embargo, el contexto en el que surge el marco, la naturaleza iterativa incremental de RUP representaba un cambio significativo en el proceso de desarrollo de software.

3.2 La medida de progreso es software funcionando y valor entregado

Durante la década del 1990, en el ámbito de la ingeniería de software emergían distintas metodologías y marcos de trabajo que buscaban una alternativa a los procesos en cascada y a la formalización del Rational Unified Process (RUP). En este contexto, surgieron enfoques como Extreme Programming (XP), centrado en la mejora continua, la simplicidad del diseño y la retroalimentación temprana [9], Feature Driven Development (FDD), orientado a la entrega incremental de funcionalidades de negocio [16] y Dynamic Systems Development Method (DSDM), que promovía la colaboración activa con el cliente y la entrega iterativa de valor [17]. También comenzaron a consolidarse Crystal Family, publicada en 2004 en [18] para adaptar la estructura de trabajo según el tamaño y criticidad del proyecto, y Scrum, que introdujo la noción de ciclos cortos de desarrollo, equipos autoorganizados y revisiones frecuentes de producto [19].

Estas propuestas sentaron las bases conceptuales y prácticas que confluían, a comienzos del nuevo milenio, en el Manifiesto Ágil, consolidando una visión centrada en la entrega de software funcionando como medida de progreso y en la generación continua de valor para el usuario.

3.3 Firma del Manifiesto Ágil

Los creadores y referentes de métodos y marcos para gestionar el desarrollo de software, mencionados en la sección anterior, notaron que sus propuestas compartían similitudes esenciales. Algunas de ellas eran enfoques que se aplicaban en empresas que aún no estaban documentadas y formalizadas, las que fueron publicadas posterior al manifiesto. La firma del Manifiesto Ágil en 2001 abrió un nuevo paradigma, donde se priorizan los individuos y sus interacciones sobre los procesos y herramientas, y la adaptabilidad al cambio por encima de la adhesión estricta a un plan [3]. A partir de allí, se observa un desplazamiento progresivo hacia marcos menos prescriptivos y más livianos, orientados a la entrega continua de valor, la colaboración interdisciplinaria y la retroalimentación temprana [15].

Scrum, por ejemplo, mantiene un conjunto definido de roles, artefactos y eventos, pero evita prescribir cómo deben implementarse las prácticas en detalle, lo que facilita su adaptación a distintos contextos organizacionales [4]. Este carácter semiestructurado lo volvió altamente popular, al ofrecer una guía clara sin el peso metodológico de modelos como RUP.

En paralelo, comenzaron a consolidarse y difundirse diversas prácticas y frameworks ágiles que fortalecieron el ecosistema postManifiesto:

- **Extreme Programming (XP)** evolucionó reforzando prácticas técnicas clave como la integración continua, el desarrollo guiado por pruebas (TDD), la refactorización continua, el diseño simple y la programación en parejas, buscando mejorar la calidad técnica y la colaboración [9].
- **Agile Modeling (AM) y Agile Unified Process (AUP)** propusieron un enfoque equilibrado entre modelado y agilidad, preservando la trazabilidad y la comunicación sin perder flexibilidad [20].
- **Crystal Family**, de Cockburn, evolucionó como un conjunto de metodologías adaptables según el tamaño y criticidad del proyecto, promoviendo la comunicación directa y la responsabilidad compartida [18].
- **Feature Driven Development (FDD)** consolidó un enfoque basado en la planificación y entrega de funcionalidades completas, priorizando la visibilidad del progreso y la calidad del diseño [16].
- **Behavior Driven Development (BDD) y Acceptance Test Driven Development (ATDD)** emergieron como prácticas derivadas de TDD, enfocadas en alinear los requerimientos del negocio con las pruebas automatizadas, facilitando la comprensión compartida entre desarrolladores y stakeholders [21].

El movimiento trascendió rápidamente la ingeniería de software, influyendo en áreas como la gestión organizacional, la innovación educativa y la transformación digital, consolidando así una nueva cultura basada en la colaboración, la adaptabilidad y la mejora continua [22].

3.4 Por fuera del paraguas Ágil

Posteriormente, comenzaron a consolidarse y difundirse diversas prácticas ágiles complementarias orientadas a la mejora continua y la calidad del producto. Estas no necesariamente se originaron en el contexto del desarrollo del software, pero compartían similitudes con la esencia de los valores y principios ágiles. El mayor exponente fue Lean Software Development, adaptado desde los principios de Lean Manufacturing, enfatiza la eliminación de desperdicios, el aprendizaje continuo, el empoderamiento de los equipos y la entrega rápida de valor [23].

En el contexto de Lean Software Development, se populariza ampliamente Kanban, inspirado en el modelo de producción Lean de Toyota, promueve la visualización del flujo de trabajo, la limitación del trabajo en curso (WIP) y la optimización continua del proceso [24]. Inclusive, la utilización de la herramienta del tablero Kanban es usualmente percibida en la industria como una práctica ágil.

Estas prácticas y frameworks configuraron un ecosistema ágil en expansión, caracterizado por la experimentación, la autoorganización y la orientación al valor entregado, consolidando un movimiento que trascendió la ingeniería de software y comenzó a permear otros ámbitos organizacionales, educativos y de gestión [25].

3.5 Escalado de agilidad y diseño organizacional ágil

Ágil y *Agile* se volvieron parte de la jerga corporativa. Por lo tanto, la demanda de la industria se volcó en la idea de escalar lo que funciona. Si un equipo ágil da buenos resultados, ¿por qué no una organización ágil? [26]. De esta manera surgen los primeros modelos de escalado, Scaled Agile Framework (SAFe) [27], LargeScale Scrum (LeSS) [28] y Disciplined Agile Delivery (DAD) [29], para escalar los principios ágiles a organizaciones grandes, manteniendo la alineación estratégica y la entrega iterativa en múltiples equipos [26].

Otra tendencia observada fue en empresas principalmente orientadas a la tecnología que desde sus comienzos trabajaron con agilidad, es el caso de múltiples startups, pero que al crecer tenían dificultades para coordinar los esfuerzos de múltiples equipos trabajando en un mismo producto. En cuanto a este aspecto, surgen modelos y prácticas como Scrum of Scrums [30] y el Modelo Spotify. Las cuales se centran en la organización y coordinación de equipos, definiendo nuevas categorías de equipos o jerarquías de los mismos.

3.6 El cambio en gestión debe ser acompañado por un cambio técnico y viceversa

Extreme Programming popularizó técnicas de desarrollo que se conocían como parte de la agilidad. Con el objetivo de cumplir con los tiempos del mercado y entregar rápidamente valor al cliente resultaban necesarias herramientas que lo permitieran, como la integración y entrega continua [31]. En este contexto, surge el movimiento DevOps, que ejemplifica propuestas menos rígidas en un ámbito técnico, al presentarse menos como una metodología prescriptiva y más como una cultura fundamentada en principios de colaboración, automatización y mejora continua [32]. Las organizaciones implementan DevOps a través de configuraciones prácticamente ilimitadas de herramientas y procesos, unidos por valores compartidos antes que por procedimientos estandarizados [33].

Este movimiento reflejó la clara necesidad de acompañar con capacidades técnicas donde la agilidad en software aparentaba estar centrada en la gestión [31]. Se reconoce a la excelencia técnica y la atención a la calidad como parte

esencial de los procesos ágiles [31]. Los equipos ágiles deben madurar no solo en su capacidad de trabajo en equipo sino también en destreza técnica [34].

3.7 Cultura basada en valores y principios

Al surgir nuevos roles en el ámbito IT, como Scrum Masters y Product Owners, al igual que la necesidad de desarrolladores formados en frameworks ágiles, aumentó la oferta de capacitaciones empresariales, consultorías y certificaciones. La agilidad, en su evolución comercial, ha experimentado lo que Cockburn [11] denominó una decoración excesiva. Aquello que comenzó como un conjunto de valores simples se ha transformado en una industria compleja de certificaciones y frameworks.

Ocurre la paradoja, de un movimiento que surge para lograr métodos livianos, adaptables y con la capacidad de responder al cambio, a uno que pareciera basarse en coleccionar certificados, donde cada examen de Scrum Master incluye más de 80 preguntas. Cockburn identificó que las implementaciones ágiles se habían vuelto excesivamente complejas y no manejables, razón por la cual creó el modelo Heart of Agile con la premisa de simplificar y volver a las bases [11]. En agilidad, si hablamos de sus bases, volvemos a los valores [6]. Heart of Agile es una herramienta de enfoque que representa la esencia de la agilidad, donde el equipo para trabajar de manera ágil, debe concentrarse en cuatro valores: colaborar, entregar, reflejar y mejorar [6].

3.8 Patrones como lenguaje organizacional

Con el paso del tiempo, muchas organizaciones adoptaron una manera ágil de trabajar, sobre todo aquellas orientadas a la tecnología o en sus equipos de desarrollo [26]. Por lo tanto, adoptaron también las características ágiles, como la inspección y adaptación como parte de su cultura organizacional. Así, seleccionan prácticas a aplicar según el contexto del problema, producto, proyecto o equipo.

La singularidad de cada problema y su contexto específico, adicionalmente a la cultura de cada organización, demanda un repertorio amplio y adaptable de estrategias. En este sentido, los patrones y lenguajes de patrones constituyen recursos fundamentales. En su origen, la teoría de patrones propuso que cada patrón se estructura como una relación tripartita que vincula un contexto específico, un problema identificado y su correspondiente solución [35].

En este contexto, del mismo autor de Management 3.0, Appelo propone el modelo unFIX [36], describiéndolo como una alternativa a SAFe, LeSS, Holacracy [37], Management 3.0 [38], el modelo Spotify y las organizaciones matriciales, que busca llevar la flexibilidad a un nuevo nivel. En su definición, se clarifica que no es un nuevo framework ágil o de escalado, es una biblioteca de patrones organizacionales. Para su formulación, se sintetizaron ideas de múltiples prácticas y marcos con características ágiles, incluyendo el modelo Spotify, Team Topologies [39], Dynamic Reteaming y Management 3.0 [38].

Otra tendencia observada fue escalar las maneras ágiles de trabajar de un equipo pequeño de trabajo para llevarlo a toda la organización, lo que se conoce como diseño organizacional ágil, donde ya no solo se tienen en cuenta las necesidades de ingeniería de software.

4 Discusión

A partir de la organización y categorización de las distintas propuestas relacionadas con la agilidad se da respuesta a las preguntas de investigación.

4.1 ¿Cómo y por qué han evolucionado los marcos ágiles?

El análisis de las propuestas que se enmarcaron en la agilidad, estudiando la evolución desde RUP hasta patrones organizacionales contemporáneos, revela una evolución que no es necesariamente lineal. Contrario a lo que podría esperarse, la progresión observada no constituye un reemplazo secuencial donde cada generación supera a la anterior, sino más bien una expansión del repertorio metodológico disponible para enfrentar problemáticas organizacionales diversas.

Esta diversificación responde a un principio fundamental de la adaptabilidad característica del enfoque ágil. No existe una solución única aplicable a todos los contextos de desarrollo de software. Mientras que frameworks fundacionales como Scrum y XP ofrecen un balance entre estructura y flexibilidad apropiado para equipos individuales, los marcos de escalamiento como SAFe y LeSS surgieron para abordar desafíos específicos de coordinación multiequipo en organizaciones grandes. Paralelamente, propuestas como Heart of Agile y Modern Agile

representan un retorno consciente a principios fundamentales frente a la creciente complejización de algunos marcos. Por otro lado, marcos técnicos de la disciplina de la ingeniería del software, que propician una cultura basada en valores ágiles y un conjunto de herramientas que facilitan procesos ágiles.

4.2 ¿Qué factores impulsaron la evolución?

Diversidad de contextos

Primero, la diversidad contextual creciente. El reconocimiento de que no existe una solución única aplicable a todos los contextos ha sido el motor principal. Una startup tecnológica opera efectivamente con Scrum y DevOps, una corporación multinacional requiere marcos de escalamiento como SAFe para coordinar cientos de equipos, una consultora con equipos maduros aprovecha la flexibilidad de patrones organizacionales como unFIX. Cada contexto presenta desafíos distintos que demandan propuestas diferenciadas. En este sentido, se reconoce también que no se precisa contar con un único marco que englobe todas las soluciones necesarias, la organización debe ser capaz de coleccionar un conjunto de técnicas, herramientas y métodos y comprender ante qué necesidades aplicar diversas soluciones, lo que nuevamente requiere de cierto grado de madurez organizacional. Esta noción responde a aplicar el principio de reflexión y adaptabilidad de marcos ágiles a nivel organizacional.

Madurez organizacional y madurez técnica

Segundo, la maduración de equipos y capacidades. En este contexto se habla tanto de madurez organizacional y de gestión, de requerir procesos externos que impongan disciplina a tener disciplina emergente desde capacidades internas, así como de madurez técnica, dominio de prácticas de ingeniería como integración continua, testing automatizado, refactoring disciplinado y arquitectura evolutiva, como evidencia el surgimiento de DevOps y la revitalización de prácticas XP. Los equipos contemporáneos han desarrollado capacidades de autoorganización y excelencia técnica que no existían en la era de RUP. Esta madurez organizacional y técnica permite operar con estructuras menos elaboradas, siempre que los principios estén internalizados. La evolución no reemplaza marcos estructurados por otros más ligeros, en cambio, habilita opciones menos prescriptivas cuando la madurez lo permite.

Mercado que no deja de cambiar rápidamente

Tercero, la aceleración del cambio en el mercado, aspecto que originó el movimiento ágil y aún impacta en su evolución. Los requerimientos cambian constantemente, con ciclos de producto cada vez más cortos. Esta realidad favorece propuestas que priorizan capacidad de respuesta sobre adherencia a procesos detallados, y principios sobre el seguimiento de guías y procedimientos.

4.3 ¿Qué limitaciones específicas motivaron la emergencia de nuevas propuestas?

Cada generación de propuestas surgió como respuesta a limitaciones concretas no resueltas por las anteriores.

La sobrecarga ceremonial de RUP motivó la búsqueda de alternativas más ligeras. La documentación extensa y los procesos formales, diseñados para mitigar riesgos, introducían rigidez donde la adaptabilidad era crítica. El Manifiesto Ágil y marcos como Scrum surgieron específicamente para reducir esta carga sin perder estructura coordinadora.

La adopción exitosa en equipos individuales reveló una nueva limitación: la dificultad de coordinar múltiples equipos trabajando en productos complejos. Scrum fue diseñado para equipos de 5 a 9 personas, pero las organizaciones necesitaban coordinar decenas o cientos de equipos. Esta brecha motivó marcos de escalamiento como SAFe, LeSS, Scrum of Scrums y el Modelo Spotify, que no surgieron para “mejorar” Scrum sino para resolver un problema que Scrum no fue diseñado para abordar.

La proliferación de estructuras de escalamiento generó complejización excesiva. Organizaciones adoptaban múltiples roles, ceremonias y artefactos sin comprender principios subyacentes, transformando la agilidad en ejecución mecánica de rituales. La decoración excesiva identificada por Cockburn y la paradoja de movimientos livianos convertidos en industrias de certificaciones motivaron propuestas de retorno a principios como Heart of Agile y Modern Agile.

La brecha entre capacidades organizacionales y técnicas evidenció que es imposible entregar software funcionando frecuentemente sin excelencia técnica. Equipos que ejecutaban ceremonias Scrum perfectamente, pero carecían de testing automatizado, integración continua o diseño evolutivo acumulaban deuda técnica insostenible. DevOps surgió no como un marco ágil tradicional sino como un conjunto de capacidades técnicas habilitadoras que

permiten la entrega continua. Como se observó en el análisis, el cambio en gestión debe ser acompañado por un cambio técnico, y viceversa.

La rigidez de marcos universales demostró que adoptar propuestas siguiendo fielmente una guía raramente funciona sin adaptación contextual. Esta tensión motivó patrones organizacionales como Team Topologies y unFIX, que ofrecen lenguaje compartido y opciones de diseño sin imponer estructura única, permitiendo que organizaciones maduras construyan soluciones informadas según su contexto específico.

4.4 ¿Qué implicaciones tiene para la práctica actual?

De hacer agilidad a ser ágil

La evolución observada señala una transición fundamental de ejecutar ceremonias hacia internalizar principios. Las organizaciones efectivas no son aquellas que ejecutan rituales ágiles perfectamente, sino las que entregan valor continuamente adaptándose a su contexto. Esta implicación es crítica para evitar la adopción superficial que convirtió movimientos livianos en colecciones de certificaciones.

La pregunta relevante cambia de ¿qué marco debemos usar? a ¿qué problemas necesitamos resolver y qué prácticas nos ayudan en nuestro contexto?. No se trata de elegir el marco correcto en abstracto, sino de desarrollar capacidad de discernimiento contextual.

De ceremonias a entrega de valor

La práctica actual debe enfocarse en resultados más que en conformidad con procesos. La proliferación de certificaciones ha creado la percepción errónea de que seguir procedimientos formales constituye “ser ágil”. Sin embargo, la agilidad genuina se mide por capacidad de respuesta y entrega de valor, no por adherencia a ceremonias específicas o acumulación de credenciales.

Para practicantes y la industria esto implica priorizar entrega frecuente de software funcionando sobre completar sprints correctamente en una herramienta de gestión de tickets o gestión de proyectos. Se busca la colaboración efectiva sobre roles formales, respuesta al cambio sobre seguimiento de planes y mejora continua basada en resultados sobre certificaciones metodológicas.

Excelencia técnica como habilitador fundamental

Como evidenció el análisis de DevOps, no existe agilidad sostenible sin excelencia técnica. Las prácticas técnicas no son secundarias sino habilitadoras fundamentales de la agilidad organizacional. Las organizaciones deben invertir simultáneamente en integración y entrega continua, testing automatizado, diseño evolutivo, y calidad interna del código.

Sin estas capacidades, los equipos pueden ejecutar ceremonias ágiles, pero permanecen incapaces de responder ágilmente a cambios con software funcionando. La excelencia técnica transforma la agilidad de aspiración retórica a realidad operacional. Esta es quizás la lección más importante: el cambio organizacional de gestión debe ser acompañado por un cambio técnico, y viceversa.

Importancia del context sobre la ortodoxia

El análisis demuestra que no existe jerarquía de superioridad entre marcos. Cada propuesta tiene su lugar según el contexto organizacional. Las organizaciones más efectivas no son aquellas que adoptan el framework más reciente ni las que se aferran rígidamente a prácticas establecidas, sino las que desarrollan capacidad de evaluar su contexto específico y seleccionar prácticas apropiadas.

Para practicantes, esto significa desarrollar criterio contextual antes que lealtad metodológica. Como propone Appelo, el diseño organizacional ágil hoy es una lista de opciones representadas en patrones, no una lista de compra única aplicable universalmente.

La coexistencia de propuestas con características distintas refleja no confusión disciplinar sino reconocimiento de heterogeneidad organizacional. Como ilustra el análisis:

- Frameworks fundacionales permanecen relevantes para equipos que inician transformación ágil
- Marcos de escalamiento son necesarios en organizaciones grandes con múltiples dependencias
- Retorno a principios es valioso para organizaciones maduras que buscan simplificar
- Capacidades técnicas son habilitadoras transversales indispensables
- Patrones organizacionales ofrecen adaptabilidad para contextos complejos y cambiantes

5 Conclusiones y Trabajos Futuros

El análisis revela que las propuestas ágiles no constituyen una secuencia evolutiva donde cada nueva familia “reemplaza” a la anterior. En cambio, representan respuestas a problemáticas organizacionales distintas que coexisten. Sin embargo, incluso estos marcos ágiles han experimentado una institucionalización creciente, con múltiples certificaciones formales, que pareciera la única opción para ser ágil. Un enfoque liviano y adaptable, paradójicamente, se transforma en uno de procedimientos y métodos.

Esta investigación revela que la evolución de la agilidad no constituye un progreso lineal desde enfoques prescriptivos hacia enfoques adaptativos, sino más bien una expansión del repertorio metodológico disponible. Cada generación de propuestas surgió como respuesta a limitaciones específicas del contexto industrial de su época, y todas permanecen relevantes en contextos apropiados.

La tensión entre estructura y adaptabilidad no es un problema por resolver eligiendo el marco adecuado, sino una característica inherente del desarrollo de software que requiere equilibrios contextuales. Las organizaciones más efectivas no son aquellas que adoptan el framework más reciente, sino las que desarrollan capacidad de seleccionar y adaptar prácticas según sus necesidades específicas.

La tendencia hacia la ligereza metodológica no debe interpretarse como un abandono de la disciplina, sino como una evolución hacia formas más sutiles y adaptativas de coordinación. La aplicación de marcos de trabajo más livianos, o incluso la adaptación y personalización de las prácticas, requieren, paradójicamente, una mayor madurez organizacional y una comprensión más profunda de los principios subyacentes de la agilidad. En este sentido, la evolución desde RUP a un conjunto de principios y un repertorio de prácticas y patrones refleja no solo un cambio metodológico, sino una transformación cultural hacia organizaciones más ágiles y conscientes de su contexto.

La progresión hacia marcos menos prescriptivos responde a múltiples factores convergentes. Primero, el reconocimiento creciente de que los contextos organizacionales varían significativamente, demandando enfoques adaptativos antes que soluciones universales. Segundo, la maduración de los equipos de desarrollo, que han desarrollado capacidades para autoorganizarse sin requerir estructuras externas elaboradas. Finalmente, la aceleración del ritmo de cambio tecnológico y de mercado ha hecho que la agilidad de respuesta sea más valiosa que la adherencia a procesos predefinidos.

Para profesionales y organizaciones, este análisis ofrece orientación sobre cuándo y cómo aplicar diferentes niveles de prescripción metodológica. Identificar los factores que impulsaron cada transición y permite tomar decisiones informadas sobre qué prácticas adoptar según el contexto específico. Adicionalmente, reconocer que menos prescripción requiere mayor madurez técnica y organizacional ayuda a evitar la adopción superficial de “agilidad sin estructura” que puede resultar contraproducente.

Agradecimientos

Este trabajo ha sido financiado por la Universidad Tecnológica Nacional y el CONICET. Se agradece el apoyo de ambas instituciones.

References

1. B. Boehm, “A view of 20th and 21st century software engineering,” in Proceedings of the 28th international conference on Software engineering, New York, NY, USA: ACM, May 2006, pp. 12–29. doi: 10.1145/1134285.1134288.
2. I. Jacobson, G. Booch, and J. Rumbaugh, The Unified Software Development Process. AddisonWesley, 1999. ISBN 0201571692.
3. K. Beck et al., “Manifiesto for Agile Software Development,” 2001.
4. K. Schwaber and J. Sutherland, “What is scrum,” URL: <http://www.scrumalliance.org/system/resource/file/275/whatIsScrum.pdf>, [Standard: 03.03.2008], 2010.
5. F. Mccaffery, R. V O’connor, and R. Messnarz, “CCIS 364 Definitions of Agile Software Development and Agility,” 2013.
6. A. Cockburn, “The Heart of Agile.” [Online]. Available: <http://static1.1.sqspcdn.com/static/f/702523/26767147/1451886700677/201601>

7. P. Hohl et al., “Back to the future: origins and directions of the ‘Agile Manifesto’ – views of the originators,” *Journal of Software Engineering Research and Development*, vol. 6, no. 1, Dec. 2018, doi: 10.1186/s404110180059z.
8. G. Kim, J. Humble, P. Debois, J. Willis, and N. Forsgren, *The DevOps handbook: How to create worldclass agility, reliability, & security in technology organizations*. IT Revolution, 2021.
9. K. Beck, *Extreme Programming Explained: Embrace Change*. Boston, MA, USA: AddisonWesley Professional, 2000.
10. S. Magistretti and D. Trabucchi, “Agileasatool and agileasaculture: a comprehensive review of agile approaches adopting contingency and configuration theories,” *Review of Managerial Science*, vol. 19, no. 1, pp. 223–253, Jan. 2025, doi: 10.1007/s11846024007451.
11. A. Cockburn, D. Carlson, B. Gallagher, K. Nidiffer, and R. Sega, “Beyond the Agile Manifesto,” *CrossTalk*, 2016.
12. E. T. Rother, “Revisão sistemática X revisão narrativa,” *Acta Paulista de Enfermagem*, vol. 20, no. 2, pp. v–vi, Jun. 2007, doi: 10.1590/S010321002007000200001.
13. H. Alaidaros, M. Omar, and R. Romli, “A Review on the Methods of Evaluating the New Approaches Proposed in the Agile Context,” *International Journal of Computer and Information Technology(22790764)*, vol. 10, no. 1, Jan. 2021, doi: 10.24203/ijcit.v10i1.54.
14. V. Garousi, M. Felderer, and M. V. Mäntylä, “Guidelines for including grey literature and conducting multivocal literature reviews in software engineering,” *Inf Softw Technol*, vol. 106, pp. 101–121, Feb. 2019, doi: 10.1016/j.infsof.2018.09.006.
15. B. Boehm, *Get Ready for Agile Methods, with Care*. 2002.
16. S. Palmer and M. Felsing, *A Practical Guide to FeatureDriven Development*. Pearson Education, 2001.
17. J. Stapleton, “DSDM: Dynamic Systems Development Method,” in *Proceedings Technology of ObjectOriented Languages and Systems. TOOLS 29 (Cat. No.PR00275)*, IEEE Comput. Soc, pp. 406–406. doi: 10.1109/TOOLS.1999.779095.
18. A. Cockburn, *Crystal Clear A HumanPowered Methodology for Small Teams*. AddisonWesley Professional, 2004.
19. K. Schwaber and J. Sutherland, “The Scrum Guide,” *Scrum Alliance*, vol. 21, no. 1, pp. 1–38, 2011, [Online]. Available: <https://www.scrumguides.org/>
20. S. W. Ambler, *The object primer: Agile modeldriven development with UML 2.0*. Cambridge University Press, 2004.
21. D. North, “Introducing behaviour driven development.”
22. S. Misra, V. Kumar, U. Kumar, K. Fantasy, and M. Akhter, “Agile software development practices: Evolution, principles, and criticisms,” Oct. 05, 2012. doi: 10.1108/02656711211272863.
23. M. Poppendieck, “Lean Software Development,” in *29th International Conference on Software Engineering (ICSE’07 Companion)*, IEEE, May 2007, pp. 165–166. doi: 10.1109/ICSECOMPANION.2007.46.
24. M. O. Ahmad, J. Markkula, and M. Oivo, “Kanban in software development: A systematic literature review,” in *2013 39th Euromicro Conference on Software Engineering and Advanced Applications*, IEEE, Sep. 2013, pp. 9–16. doi: 10.1109/SEAA.2013.28.
25. R. Hoda, N. Salleh, and J. Grundy, “The Rise and Evolution of Agile Software Development”, doi: 10.21227/H2T08T.
26. K. Dikert, M. Paasivaara, and C. Lassenius, “Challenges and success factors for largescale agile transformations: A systematic literature review,” *Journal of Systems and Software*, vol. 119, pp. 87–108, Sep. 2016, doi: 10.1016/j.jss.2016.06.013.

27. O. Turetken, I. Stojanov, and J. J. M. Trienekens, "Assessing the adoption level of scaled agile development: a maturity model for Scaled Agile Framework," *Journal of Software: Evolution and Process*, vol. 29, no. 6, Jun. 2017, doi: 10.1002/smr.1796. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016)
28. C. Larman and B. Vodde, *Largescale scrum: More with LeSS*. AddisonWesley Professional, 2016.
29. S. W. Ambler and M. Lines, *Disciplined agile delivery: A practitioner's guide to agile software delivery in the enterprise*. IBM press, 2012.
30. M. Paasivaara, C. Lassenius, and V. T. Heikkilä, "Interteam coordination in largescale globally distributed scrum," in *Proceedings of the ACM/IEEE international symposium on Empirical software engineering and measurement*, New York, NY, USA: ACM, Sep. 2012, pp. 235–238. doi: 10.1145/2372251.2372294. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999)
31. A. Alami, O. Krancher, and M. Paasivaara, "The journey to technical excellence in agile software development," *Inf Softw Technol*, vol. 150, Oct. 2022, doi: 10.1016/j.infsof.2022.106959.
32. C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano, "DevOps," *IEEE Softw*, vol. 33, no. 3, pp. 94–100, 2016.
33. R. Jabbari, N. Bin Ali, K. Petersen, and B. Tanveer, "What is DevOps? A systematic mapping study on definitions and practices," in *Proceedings of the Scientific Workshop Proceedings of XP2016*, 2016, pp. 1–11.
34. R. Hanslo, "The Integration of Agile Methodologies in DevOps Practices The Integration of Agile Methodologies in DevOps Practices within the Information Technology Industry within the Information Technology Industry Part of the Databases and Information Systems Commons, Management Information Systems Commons, and the Software Engineering Commons." [Online]. Available: <https://digitalcommons.kennesaw.edu/acist>
35. C. Alexander, *A pattern language: towns, buildings, construction*. Oxford University Press, 1977.
36. J. Appelo, "The unFIX Model," <https://unfix.work>.
37. A. Ore, O. Kuznecova, and A. Jegorova, "Selfmanaged Organization: A Role of Business Process Management," in *2021 62nd International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS)*, IEEE, Oct. 2021, pp. 1–6. doi: 10.1109/ITMS52826.2021.9615260.
38. J. Appelo, *Management 3.0: Leading Agile Developers, Developing Agile Leaders*. Boston, MA, USA: AddisonWesley / Pearson Education, 2011.
39. M. Skelton and M. Pais, *Team topologies: organizing business and technology teams for fast flow*. IT Revol, 2019. LNCS Homepage, <http://www.springer.com/lncs>, last accessed 2023/10/25

Enfoque de arquitectura orientada a agentes para la mejora de calidad de requerimientos ágiles

Noelia Pinto

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación (CInApTIC), Argentina.
ns.pinto@gmail.com

César J. Acuña

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación (CInApTIC), Argentina.
csr.acn@gmail.com

Gabriela Tomaselli

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación (CInApTIC), Argentina.
gabriela.tomaselli@gmail.com

Nicolás Tortosa

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación (CInApTIC), Argentina.
nicotortosa@gmail.com

Denise Martinez

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación (CInApTIC), Argentina.
denise_utn@gmail.com

Jerónimo Zapata

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación (CInApTIC), Argentina.
jerezapata13@gmail.com

Martín Lopez Soto

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación (CInApTIC), Argentina.
martinlopezsoto08@gmail.com

Ariano Miranda

Universidad Tecnológica Nacional, Facultad Regional Resistencia, Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación (CInApTIC), Argentina.
arianogabriel02@gmail.com

Resumen

La calidad de los requerimientos constituye un aspecto crítico en proyectos ágiles de desarrollo de software, donde la flexibilidad, la iteración continua y la colaboración con los stakeholders pueden dificultar la claridad, completitud y trazabilidad del backlog. Este artículo presenta una propuesta de arquitectura orientada a agentes como marco innovador para brindar soporte inteligente a la gestión de requerimientos en contextos ágiles. La solución se basa en un modelo conceptual de calidad previamente desarrollado, operacionalizado mediante un sistema

multiagente compuesto por cuatro agentes especializados: Clasificador, Validador, Priorizador y Enriquecedor, coordinados por un Kernel Orquestador. Esta arquitectura incorpora capacidades cognitivas basadas en modelos de lenguaje de gran escala y una memoria semántica que almacena requerimientos históricos, plantillas estandarizadas y patrones de calidad. El enfoque se encuentra en fase de diseño inicial y busca establecer las bases para la construcción de un asistente cognitivo que colabore con los equipos de desarrollo, fortaleciendo los requerimientos a fin de mejorar la calidad del proceso ágil y del producto de software resultante.

Palabras clave: Requerimientos ágiles; Calidad de requerimientos; Sistemas multiagente; Modelos de lenguaje de gran escala; Asistencia cognitiva

1. Introducción

La calidad de los requerimientos constituye un factor decisivo para el éxito de los proyectos de software. En entornos ágiles, donde los equipos priorizan entregas incrementales, la colaboración con los stakeholders y la flexibilidad ante los cambios, la gestión de requerimientos enfrenta desafíos persistentes vinculados a su completitud, claridad, priorización y trazabilidad [1, 2]. Estas dificultades impactan directamente en la planificación, la estimación de esfuerzo y la coherencia del backlog, comprometiendo tanto la calidad del proceso como del producto [3, 4].

Estudios previos han demostrado que la incorporación de marcos de evaluación de calidad de requerimientos en metodologías ágiles fortalece la estabilidad de los proyectos y la confianza entre los actores involucrados [5, 6]. Sin embargo, las propuestas existentes suelen carecer de automatización y de mecanismos inteligentes que asistan a los equipos durante el refinamiento continuo de requerimientos. En este contexto, los SMA emergen como un enfoque prometedor, ya que permiten modularizar funciones, delegar responsabilidades en agentes especializados y coordinar procesos complejos mediante un kernel orquestador [7].

Este trabajo propone un enfoque de arquitectura orientada a agentes diseñado para mejorar la calidad de los requerimientos ágiles. La propuesta se centra en la definición de un middleware orquestador que operacionaliza los componentes de gestión de requerimientos a través de agentes especializados, apoyados en técnicas de Inteligencia Artificial y en una memoria semántica capaz de aportar consistencia y aprendizaje incremental.

El resto de este artículo se organiza de la siguiente manera: la Sección 2 presenta el marco teórico que sustenta la propuesta, con énfasis en la ingeniería de requerimientos ágil, la calidad de los requerimientos y el enfoque multiagente. La Sección 3 describe la propuesta de arquitectura y su integración con un modelo de calidad de requerimientos. La Sección 4 analiza las mejoras técnicas y la aplicación práctica en proyectos ágiles. Finalmente, la Sección 5 expone las conclusiones y los trabajos futuros.

2. Marco Teórico

El marco teórico constituye el sustento conceptual de la propuesta, al establecer los antecedentes y enfoques que han orientado la investigación en torno a la ingeniería de requerimientos en contextos de agilidad. En particular, se abordan tres dimensiones principales: (i) la evolución de la Ingeniería de Requerimientos en la agilidad, (ii) la calidad de los requerimientos como factor determinante de éxito y (iii) los SMA como enfoque arquitectónico para asistir procesos complejos de validación, priorización y refinamiento.

2.1. Ingeniería de Requerimientos en la Agilidad

La comunidad científica ha señalado que los principios de la agilidad —colaboración cercana, entregas iterativas y aceptación de cambios tardíos— reconfiguran las prácticas tradicionales de Ingeniería de Requerimientos (RE, por sus siglas en inglés *Requirements Engineering*). Sin embargo, esta flexibilidad introduce nuevos retos: escasa trazabilidad, dificultades en la documentación y subestimación de requisitos no funcionales [2, 3, 4].

Revisiones sistemáticas como la de Inayat et al. [1] y el mapeo de Curcio et al. [8] sintetizan estos desafíos y prácticas emergentes, confirmando que la gestión de requerimientos en la agilidad se ha consolidado como un campo en sí mismo.

Más recientemente, Bjarnason et al. [9] y Hoy y Xu [10] subrayan la necesidad de integrar prácticas clásicas de RE (documentación estructurada, técnicas de validación y trazabilidad) con valores de la agilidad, para mejorar la calidad del producto sin perder flexibilidad.

Estudios empíricos previos también resaltan las tensiones prácticas de esta integración (Paetsch et al. [11]; Ramesh et al. [12]), mostrando que la RE en contextos ágiles requiere soluciones adaptativas y, cada vez más, soporte automatizado.

2.2. Calidad de Requerimientos

La literatura coincide en que la calidad de los requerimientos impacta de manera directa en el éxito del proyecto [13]. Modelos clásicos (IEEE 830) establecen atributos como claridad, completitud, consistencia y verificabilidad [14]. En contextos de agilidad, varios enfoques se han orientado a adaptar estos criterios a artefactos como historias de usuario [15, 16].

Propuestas como el Agile Quality Framework (AQF) [5, 6] y el Quality User Story Framework (QUS) [4] buscan operacionalizar métricas de calidad en historias de usuario, generando indicadores que permiten evaluar el backlog. No obstante, investigaciones recientes señalan la carencia de soporte automatizado y la necesidad de enfoques que integren inteligencia artificial y análisis semántico [17, 18].

2.3. Sistemas Multiagente

Los sistemas multiagente (SMA) han demostrado ser útiles en dominios donde múltiples actores interactúan de manera autónoma y colaborativa para alcanzar objetivos comunes [7]. En ingeniería de software, han sido aplicados en negociación de requerimientos, monitoreo y validación automática [19].

Wooldridge et al. [7] sentaron las bases metodológicas de la orientación a agentes, mientras que trabajos más recientes integran agentes cognitivos con técnicas de procesamiento de lenguaje natural para asistir en la identificación, análisis y refinamiento de requerimientos [17, 20].

El potencial de los SMA reside en su capacidad de combinar autonomía, cooperación y adaptabilidad, cualidades críticas para asistir a los equipos que trabajan bajo principios de agilidad en la construcción de backlogs de mayor calidad.

2.4. Habilidades de los Agentes

En el marco de los SMA, cada agente se caracteriza no solo por su autonomía y capacidad de interacción, sino también por el conjunto de habilidades que le permiten ejecutar funciones específicas [7, 19]. Estas habilidades constituyen la unidad operativa mediante la cual los agentes procesan información, cooperan entre sí y contribuyen al logro de objetivos globales.

De acuerdo con Jennings et al. [17] y Kolp et al. [19], las habilidades de un agente pueden clasificarse en:

- **Habilidades funcionales:** aquellas orientadas a la ejecución de tareas específicas del dominio, como clasificación, validación o priorización de requerimientos.
- **Habilidades cognitivas:** capacidades de razonamiento, interpretación y toma de decisiones, frecuentemente potenciadas mediante modelos de lenguaje o técnicas de inteligencia artificial [17, 20].
- **Habilidades de cooperación:** mecanismos de comunicación y coordinación que permiten a los agentes colaborar entre sí, compartir conocimiento y generar resultados consistentes a nivel de SMA.
- **Habilidades adaptativas:** permiten modificar su comportamiento según cambios en el entorno o en los objetivos, asegurando la flexibilidad del sistema en contextos dinámicos como la agilidad [7].

En el caso particular de la gestión de requerimientos en contextos ágiles, las habilidades de los agentes pueden asociarse directamente a las dimensiones de calidad:

- El agente clasificador posee habilidades para identificar el tipo de requerimiento (funcional, no funcional, técnico, de negocio), estimar su complejidad, mapear dependencias e inferir su impacto.
- El agente validador integra habilidades orientadas a comprobar atributos de calidad, como necesidad, claridad, completitud, factibilidad y verificabilidad, siguiendo estándares como IEEE 830 [14].
- El agente priorizador cuenta con habilidades de análisis de valor de negocio, esfuerzo y urgencia, aplicando técnicas como Planning Poker y MoSCoW [15].
- El agente enriquecedor incorpora habilidades de reformulación semántica y contextualización, proponiendo mejoras textuales y verificando coherencia del backlog frente a requerimientos existentes [4, 16].

La incorporación explícita de estas habilidades no solo garantiza modularidad y trazabilidad en el análisis, sino que habilita un diseño extensible en el cual cada agente puede evolucionar de manera independiente, añadiendo nuevas capacidades sin alterar la arquitectura global.

2.5. Sistemas Multiagente de Inteligencia Artificial Generativa

La reciente irrupción de la Inteligencia Artificial Generativa (IAG) ha potenciado significativamente el diseño y las aplicaciones de los SMA. A diferencia de los enfoques clásicos, en los que los agentes encapsulan reglas o algoritmos predefinidos [7, 19], la IAG introduce modelos de lenguaje de gran escala (LLMs) que dotan a los agentes de capacidades cognitivas avanzadas, como razonamiento contextual, reformulación semántica y generación automática de propuestas de mejora [20].

En este marco, la combinación de arquitecturas multiagente con LLMs permite concebir sistemas cognitivos distribuidos, donde cada agente no solo ejecuta funciones específicas, sino que además interactúa con el modelo generativo para ampliar su capacidad de interpretación y acción. Así, se logra un doble nivel de inteligencia:

- Nivel individual: cada agente utiliza el modelo generativo para enriquecer su habilidad (por ejemplo, un agente validador que detecta ambigüedades en el lenguaje natural de un requerimiento).
- Nivel colectivo: el SMA aprovecha la cooperación entre agentes para producir un resultado integrado, manteniendo trazabilidad y consistencia histórica a través de memorias semánticas compartidas [18].

Diversos estudios recientes han explorado esta convergencia. Arulmohan et al. [21] destacan el potencial de los LLMs para extraer modelos de dominio a partir de requerimientos textuales, mientras que Herwanto et al. [20] muestran su aplicación en la detección de requisitos de privacidad mediante técnicas de procesamiento de lenguaje natural. Más ampliamente, frameworks de agentes generativos como AutoGen y LangChain han evidenciado la viabilidad de coordinar múltiples LLMs para tareas complejas de razonamiento y planificación [22, 23].

En el contexto de la gestión de requerimientos en la agilidad, esta sinergia permite concebir agentes inteligentes capaces de reformular requerimientos, detectar inconsistencias, enriquecer backlogs y proponer trazabilidad automática, convirtiéndose en asistentes cognitivos que colaboran con los equipos humanos. Esta perspectiva inaugura una línea de investigación prometedora en la que los SMA de IAG actúan como mediadores entre el juicio humano y la automatización inteligente en la RE.

3. Sistema multiagente para la gestión de requerimientos ágiles

La presente sección describe el diseño de un SMA orientado a mejorar la calidad de los requerimientos en contextos de agilidad. El enfoque integra un modelo conceptual de calidad previamente definido con una arquitectura distribuida basada en agentes especializados, que operan de manera coordinada a través de un kernel orquestador.

Este sistema no busca reemplazar el juicio humano, sino complementarlo con capacidades inteligentes de análisis, validación y enriquecimiento, apoyándose en técnicas de Inteligencia Artificial, procesamiento de lenguaje natural y memoria semántica. De esta forma, se provee a los equipos ágiles de un asistente cognitivo capaz de fortalecer la calidad del backlog, optimizar la planificación y mejorar la trazabilidad de los requerimientos.

El enfoque inteligente que se presenta en este trabajo y que se ha diagramado en la Figura 1, constituye un SMA cuyas características se describen en las próximas secciones.



Figura 1. Diagrama de componentes del SMA propuesto

3.1 Modelo de Calidad: Una primera aproximación

La base de la propuesta se sustenta en un modelo de calidad de requerimientos ágiles, concebido como una respuesta a los problemas recurrentes de claridad, completitud y priorización identificados en estudios empíricos y revisiones sistemáticas [1, 2, 8, 24]. Este modelo no se limita a evaluar requerimientos de manera aislada, sino que los considera dentro del backlog ágil, donde la interacción entre ítems y la coherencia global son determinantes para la calidad del producto.

El modelo se organiza en cuatro dimensiones complementarias, que actúan como capas de análisis progresivo:

- **Clasificación del requerimiento** El primer paso consiste en identificar la naturaleza del requerimiento (funcional, no funcional, técnico o de negocio). Esta clasificación no solo facilita la organización del backlog, sino que también permite aplicar métricas de calidad específicas según el tipo detectado. A su vez, el modelo contempla atributos como complejidad, dependencias e impacto, que son fundamentales para anticipar riesgos y planificar el desarrollo.
- **Validación de atributos de calidad** Inspirado en estándares como IEEE 830 [14] y en conceptos de diversos enfoques ágiles [4, 16], el modelo define un conjunto de criterios de validación: claridad, completitud, consistencia, factibilidad, verificabilidad y necesidad. Cada requerimiento debe cumplir con estos atributos antes de avanzar en el ciclo de vida, reduciendo así la probabilidad de retrabajos posteriores.
- **Priorización transparente y colaborativa** El modelo integra técnicas reconocidas como MoSCoW y Planning Poker [15], combinando criterios de valor de negocio, esfuerzo estimado y urgencia. Esto permite alinear la planificación con las expectativas de los stakeholders, evitando que la priorización se convierta en un proceso arbitrario o dependiente de decisiones unilaterales.
- **Enriquecimiento y coherencia contextual** Finalmente, el modelo incorpora un proceso de enriquecimiento que apunta a mejorar la redacción, completar información faltante, detectar redundancias y verificar la coherencia del requerimiento con el backlog existente. Este componente resulta clave para evitar duplicaciones, inconsistencias y solapamientos entre ítems, favoreciendo la construcción de un backlog robusto y mantenible.

En conjunto, estas dimensiones conforman un ciclo iterativo de aseguramiento de calidad, que comienza con la captura del requerimiento y culmina en su reformulación enriquecida, lista para ser incorporada en el backlog. La originalidad del modelo radica en su integración progresiva: cada dimensión añade valor incremental y habilita la intervención de agentes especializados que operacionalizan estas tareas en la arquitectura propuesta.

3.2 Arquitectura general

La arquitectura del SMA se diseñó con el propósito de operacionalizar el modelo de calidad de manera modular, trazable y adaptable a contextos de agilidad. Se estructura bajo un enfoque en capas que refleja la separación de responsabilidades y favorece la escalabilidad del sistema. En ella, los procesos de análisis, validación y enriquecimiento de requerimientos se distribuyen entre agentes especializados, coordinados por un núcleo central que actúa como orquestador inteligente.

El principio rector de este diseño es la colaboración entre agentes autónomos y componentes cognitivos, que trabajan de forma coordinada para transformar un requerimiento inicial —ingresado por el usuario o proveniente de una herramienta externa— en un requerimiento enriquecido, acompañado de métricas e indicadores de calidad. De este modo, la arquitectura combina la eficiencia del procesamiento distribuido con las capacidades interpretativas y generativas de la inteligencia artificial, dando lugar a un sistema dinámico, evolutivo y centrado en la mejora continua de la calidad de los requerimientos.

La arquitectura se organiza en cuatro capas principales, tal como se observa en la Figura 2:

- **Interfaz de Entrada y Salida:** constituye el punto de interacción entre los usuarios y el sistema. Permite el registro de requerimientos a través de formularios o integraciones con herramientas de gestión ágiles (por ejemplo, Jira o Trello). Además, proporciona la salida enriquecida, que incluye métricas de calidad, observaciones y sugerencias de reformulación generadas por los agentes.
- **Kernel Orquestador:** actúa como el componente central encargado de regular la interacción entre los agentes. Implementa una lógica de control adaptativa capaz de decidir dinámicamente cuáles deben intervenir según el estado y las características del requerimiento. Este módulo también administra la secuencia de ejecución, pudiendo habilitar tanto flujos secuenciales como paralelos, lo que garantiza eficiencia y flexibilidad operativa.
- **Agentes Especializados:** cada agente traduce uno de los componentes del modelo conceptual en un conjunto de funciones operativas y habilidades específicas (ver Sección 2.4). En conjunto, representan el núcleo funcional del sistema:
- **Componentes Cognitivos y Contextuales:** complementan el trabajo de los agentes mediante capacidades de razonamiento y soporte de conocimiento.
 - **Modelos de Lenguaje de Gran Escala:** constituyen el núcleo cognitivo del sistema, proporcionando a los agentes la capacidad de interpretar lenguaje natural, razonar sobre dependencias y generar propuestas de mejora textual.
 - **Memoria Semántica:** funciona como repositorio contextual, almacenando requerimientos históricos, plantillas estandarizadas y patrones de calidad. Garantiza consistencia en los resultados y permite aprendizaje incremental a lo largo del tiempo.

Así, estas capas conforman una arquitectura híbrida, donde la autonomía de los agentes se complementa con la inteligencia contextual de los modelos de lenguaje y la memoria semántica, posibilitando un procesamiento adaptativo y orientado a la mejora continua de la calidad de los requerimientos ágiles.

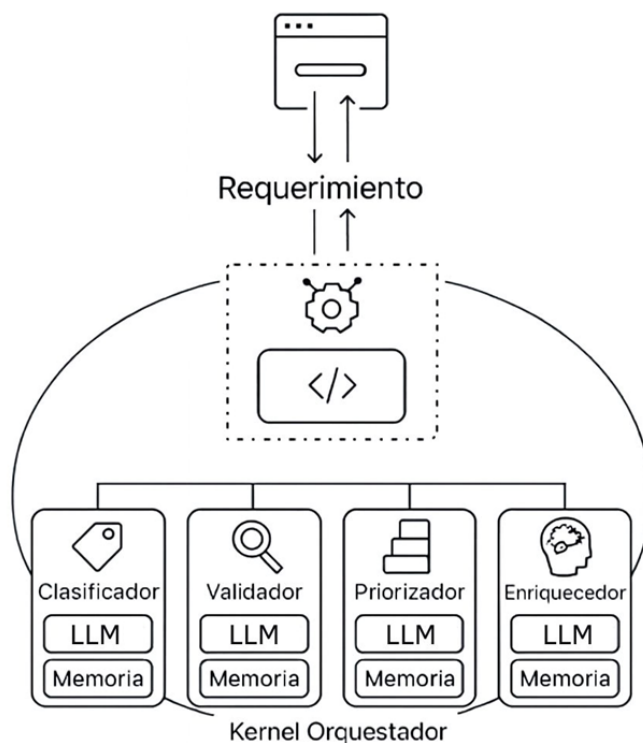


Figura 2. Arquitectura general

3.3 Flujo operativo

El flujo operativo del SMA refleja la dinámica de colaboración entre los distintos componentes que conforman la arquitectura, permitiendo transformar un requerimiento inicial en una versión enriquecida y evaluada bajo criterios de calidad definidos. Este proceso no se limita a una secuencia lineal de tareas, sino que se concibe como un ciclo adaptativo y cognitivo, capaz de ajustar su comportamiento según las características particulares de cada requerimiento y de aprender de experiencias previas para mejorar su desempeño a lo largo del tiempo.

El proceso se inicia con la captura del requerimiento, que puede ser ingresado directamente por el usuario o recibido mediante integraciones con herramientas de gestión ágiles. Una vez registrado, el requerimiento es transferido al Kernel Orquestador, componente central encargado de analizar su estado inicial, identificar qué información ya posee y definir qué agentes deben intervenir. Esta decisión no se basa en una ruta fija, sino en un esquema de control adaptativo que permite diferentes trayectorias de ejecución según la completitud y el tipo del requerimiento.

A partir de esa evaluación inicial, el Kernel activa la intervención de los agentes especializados, los cuales comienzan a operar de manera coordinada. Cada agente consulta la memoria semántica, que almacena un registro estructurado de requerimientos previos, patrones de calidad y plantillas validadas, con el fin de recuperar contexto y garantizar consistencia histórica. Este componente permite, por ejemplo, comparar un requerimiento actual con versiones anteriores de características similares o identificar incoherencias con otros elementos del backlog. Así, la memoria actúa como un mecanismo de trazabilidad y aprendizaje organizacional, asegurando que las mejoras introducidas se acumulen en el tiempo y fortalezcan la madurez del sistema.

Simultáneamente, los LLMs operan como núcleo cognitivo del sistema, asistiendo a los agentes en tareas de razonamiento semántico, interpretación de ambigüedades y generación de alternativas de mejora. Los LLMs aportan una dimensión de inteligencia generativa que potencia la capacidad de los agentes para realizar análisis más profundos y contextualizados. Gracias a su integración, el sistema no se limita a ejecutar validaciones sintácticas o reglas predefinidas, sino que puede identificar matices lingüísticos, sugerir reformulaciones más precisas, detectar redundancias y proponer una redacción más alineada con los objetivos del proyecto o con el dominio del producto.

El flujo de procesamiento se caracteriza, además, por su naturaleza no secuencial. En lugar de seguir una ruta

preestablecida, el sistema permite itinerarios dinámicos y adaptativos. Por ejemplo, un requerimiento que ya haya sido clasificado puede pasar directamente al proceso de validación sin repetir etapas innecesarias, mientras que otro que cuente con información parcial sobre valor y esfuerzo puede ser priorizado en paralelo con la validación de su completitud o consistencia. Esta flexibilidad posibilita un aprovechamiento eficiente de los recursos computacionales y evita redundancias en el análisis, contribuyendo a un ciclo de refinamiento más ágil y coherente con los principios de la agilidad.

Durante la ejecución, los resultados parciales generados por cada agente son consolidados por el Kernel Orquestador, que actúa como mediador entre las diferentes etapas. En esta instancia, se integran los hallazgos provenientes de la clasificación, validación, priorización y enriquecimiento, y se construye una versión enriquecida del requerimiento, acompañada de indicadores de calidad objetivos y observaciones contextualizadas. Estos indicadores permiten visualizar de manera cuantitativa el nivel de madurez y adecuación del requerimiento respecto de los atributos definidos en el modelo de calidad, mientras que las observaciones brindan un espacio cualitativo de reflexión y mejora continua.

El resultado final del flujo operativo se materializa en un requerimiento refinado y asistido, que no reemplaza el juicio ni la decisión de los equipos humanos, sino que actúa como un asistente cognitivo para Product Owners, desarrolladores y stakeholders. Este asistente contribuye a reducir la carga cognitiva en tareas repetitivas, evita discusiones improductivas durante la planificación y aporta evidencia sistemática que respalda las decisiones de priorización y refinamiento. De este modo, el SMA se convierte en una herramienta de apoyo para la gestión ágil de requerimientos, al mismo tiempo que promueve la calidad del software como resultado emergente de un proceso colaborativo e inteligente.

La propuesta de diseño asegura que la arquitectura no solo implemente el modelo de calidad, sino que también aporte un mecanismo inteligente y flexible de soporte a la gestión de requerimientos en la agilidad, contribuyendo a reducir ambigüedades, inconsistencias y sobrecarga en la toma de decisiones.

4. Implementación práctica de la propuesta

La implementación práctica del SMA se orienta a demostrar la viabilidad operativa y el impacto cualitativo del enfoque propuesto dentro de contextos ágiles reales. El objetivo no es únicamente validar la arquitectura desde un punto de vista técnico, sino también evaluar su contribución a la mejora del proceso de gestión de requerimientos y, en consecuencia, a la calidad del software resultante.

El entorno de aplicación elegido se corresponde con marcos de trabajo ágiles como Scrum o Kanban, en los cuales la gestión del backlog y la priorización de historias de usuario constituyen actividades críticas. En este contexto, el SMA actúa como un middleware inteligente, que se integra con las herramientas de gestión existentes (por ejemplo, Jira, Trello o Azure DevOps) mediante APIs o conectores REST. Esta integración permite que los agentes operen sobre los requerimientos en tiempo real, analizando, validando y enriqueciendo las historias de usuario a medida que son registradas o modificadas por los equipos de desarrollo.

Durante las pruebas iniciales de implementación, se diseñó un entorno controlado que simula un flujo de trabajo típico de Sprint Planning. En este escenario, los agentes realizan tareas específicas de acuerdo con sus roles: el Agente Clasificador identifica el tipo y complejidad de cada historia, el Agente Validador evalúa su conformidad con los atributos de calidad definidos, el Agente Priorizador establece un orden de ejecución basado en el valor de negocio y esfuerzo, y el Agente Enriquecedor reformula o completa la información para mejorar la coherencia del backlog.

Estos procesos se ejecutan de manera semiautomatizada, bajo la supervisión del Kernel Orquestador, que coordina la secuencia de agentes y determina las rutas más eficientes en función del estado de cada requerimiento. Las decisiones del Kernel se apoyan en la memoria semántica, que conserva trazas de ejecuciones anteriores, y en los LLMs, que proporcionan análisis de contexto y sugerencias de mejora. El resultado es un flujo de refinamiento más ágil, preciso y documentado, donde cada requerimiento es devuelto al backlog con indicadores de calidad cuantitativos y observaciones cualitativas.

Desde un punto de vista técnico, el sistema fue implementado como un conjunto de microservicios distribuidos, cada uno correspondiente a un agente especializado, desplegados en contenedores Docker y orquestados mediante Kubernetes. Este enfoque de infraestructura permite escalabilidad horizontal, asegurando que la cantidad de agentes activos se ajuste dinámicamente a la carga de trabajo. La comunicación entre componentes se realiza a través de un message broker (RabbitMQ), que garantiza asincronía, resiliencia y trazabilidad en las interacciones.

En la capa cognitiva, los agentes se comunican con LLMs mediante APIs (como OpenAI o HuggingFace), empleando prompts estructurados que encapsulan las reglas de análisis y los criterios de validación definidos en el modelo de calidad. Para la persistencia y recuperación de conocimiento contextual, la memoria semántica se implementó utilizando una base de datos orientada a grafos (Neo4j), que permite modelar las relaciones entre requerimientos, atributos de calidad y decisiones históricas. Este diseño posibilita la búsqueda semántica, la inferencia de patrones y la generación de recomendaciones personalizadas.

Este enfoque proporciona un nuevo paradigma de asistencia cognitiva en la gestión ágil de requerimientos, en el que la inteligencia artificial no sustituye la decisión humana, sino que la potencia mediante análisis contextual, evidencia empírica y aprendizaje continuo. Su carácter innovador radica en la combinación de: la modularidad y autonomía de los SMA, las capacidades de razonamiento generativo de los LLMs, y la memoria semántica como eje de aprendizaje organizacional.

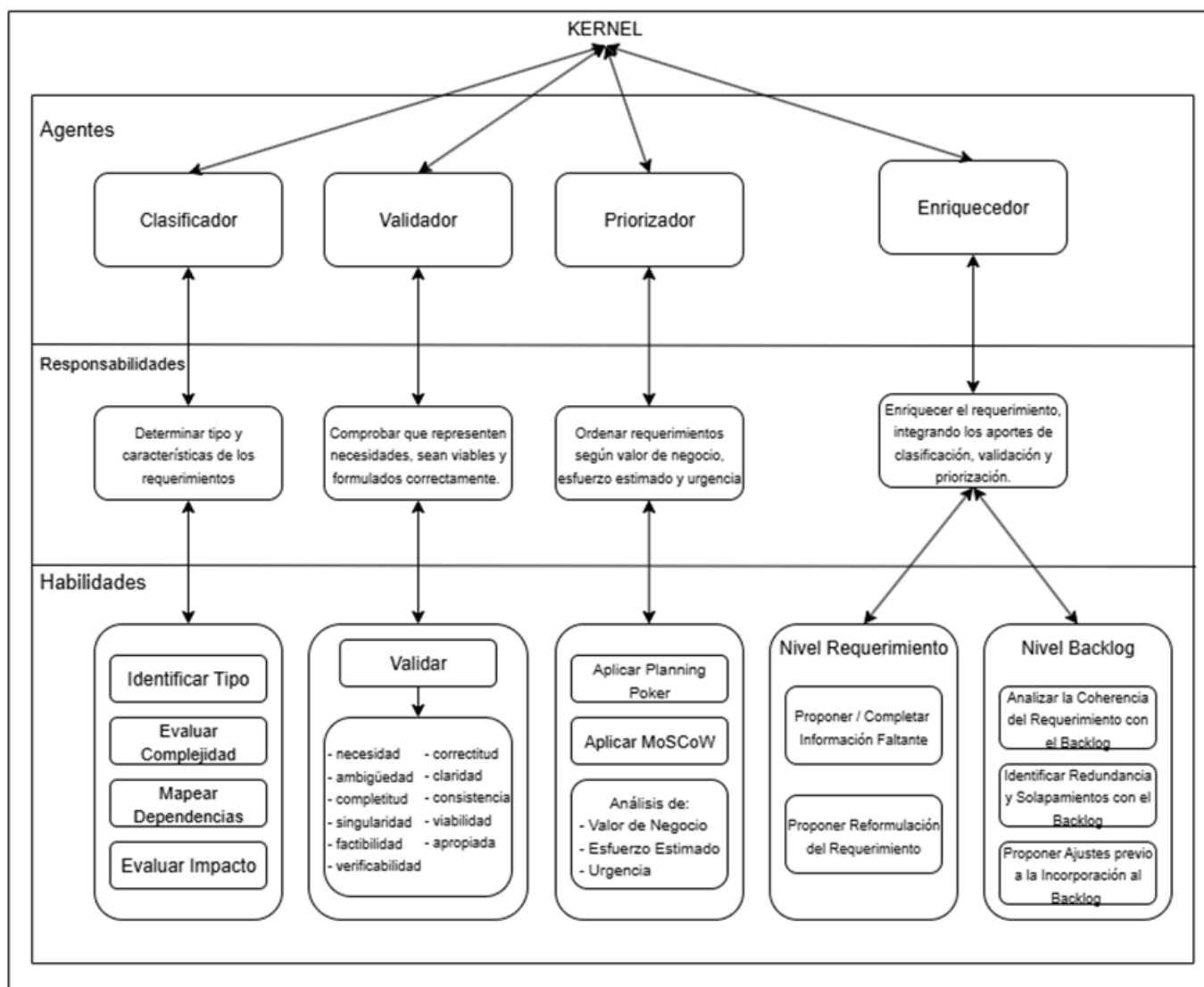


Figura 3. Resumen de Flujo operativo de la propuesta

De esta forma, como se ve en la Figura 3, la propuesta no solo automatiza tareas rutinarias de verificación y priorización, sino que también habilita un entorno de mejora continua y adaptativa, en el que cada interacción incrementa la base de conocimiento del sistema y refuerza la consistencia de los procesos de ingeniería de requerimientos. En última instancia, la implementación práctica de este enfoque demuestra la posibilidad de evolucionar desde modelos ágiles asistidos hacia verdaderos ecosistemas inteligentes de gestión de requerimientos, donde la colaboración entre humanos y agentes cognitivos se traduce en calidad, eficiencia y trazabilidad sostenibles.

5. Conclusiones y Trabajos Futuros

El trabajo presentado aborda la problemática de la calidad de los requerimientos en entornos ágiles, mediante una arquitectura orientada a agentes como una solución innovadora para el soporte inteligente en la gestión del backlog. La contribución central radica en el diseño de un SMA que operacionaliza un modelo de calidad previamente propuesto, a través de la colaboración de cuatro agentes especializados: Clasificador, Validador, Priorizador y Enriquecedor, que trabajan de forma coordinada.

El enfoque introduce un Kernel Orquestador, que permite integrar la modularidad y autonomía de los SMA con las capacidades cognitivas de los Modelos de Lenguaje de Gran Escala y con una memoria semántica diseñada para almacenar requerimientos históricos, plantillas estandarizadas y patrones de calidad, contribuyendo así a mejorar la coherencia del backlog y la completitud de las historias.

En este sentido, el sistema actúa como un asistente cognitivo capaz de generar sugerencias de mejora contextualizadas, colaborando con los equipos en el refinamiento de requerimientos a la vez que brinda soporte en la gestión de requerimientos en entornos ágiles.

No obstante, esta investigación se encuentra aún en una fase de diseño inicial y requiere de instancias posteriores, con una agenda empírica y tecnológica concreta:

- (i) Validación en Contextos Reales: Aplicación del modelo en estudios de caso controlados que permitan verificar su impacto cualitativo y cuantitativo en la mejora de la calidad del proceso de gestión de requerimientos en proyectos ágiles reales.
- (ii) Implementación del Patrón RAG (Retrieval-Augmented Generation): Integración de un patrón RAG para que los agentes accedan y utilicen el backlog completo y los requerimientos históricos en la memoria semántica como contexto extendido antes de generar respuestas o sugerencias.
- (iii) Optimización de Agentes: Ajuste detallado y optimización de los prompts utilizados por los agentes, así como el desarrollo de nuevas habilidades para mejorar su desempeño y capacidad de adaptación dentro del flujo operativo.
- (iv) Integración de Feedback: Diseño e implementación de un módulo que permita a los usuarios retroalimentar el sistema, fortaleciendo el aprendizaje continuo de la memoria semántica y la capacidad de adaptación de los agentes.

En síntesis, el marco de arquitectura orientada a agentes propuesto constituye un paso inicial determinante hacia la mejora sistemática de los requerimientos ágiles. Su consolidación mediante validaciones empíricas y desarrollos tecnológicos permitirá evaluar con mayor precisión cómo este soporte cognitivo incide tanto en la calidad del proceso como en la del producto de software, contribuyendo a cerrar la brecha entre investigación académica y práctica industrial.

6. Agradecimientos

Este trabajo se enmarca en las actividades relacionadas con el Proyecto de Investigación “Métodos, técnicas y herramientas para mejorar y evaluar la calidad de requisitos en proyectos ágiles de Software” (PID SIECRE0008643) financiado por la Secretaría de Ciencia y Tecnología de la Universidad Tecnológica Nacional, en la órbita del Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación (CInApTIC) de la Facultad Regional Resistencia de la Universidad Tecnológica Nacional.

Referencias

- [1] Inayat, I., Salim, S.S., Marczak, S., Daneva, M., Shamshirband, S.: A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior* 51, 915–929 (2015).
- [2] Schön, E.M., Winter, D., Escalona, M.J., Thomaschewski, J.: Key challenges in agile requirements engineering. In: *XP 2017, LNBP 283*, pp. 37–51. Springer (2017).
- [3] Franch, X., Palomares, C., Quer, C., Chatzipetrou, P., Gorschek, T.: The state-of-practice in requirements specification: An extended interview study at 12 companies. *Requirements Engineering* 28(3), 377–409 (2023).
- [4] Lucassen, G., Dalpiaz, F., van der Werf, J.M.E., Brinkkemper, S.: Improving agile requirements: the quality user story framework and tool. *Requirements Engineering* 21(3), 383–403 (2016).

- [5] Pinto, N.S., Tortosa, N., Cabas Geat, B., Ibáñez, L., Bollati, V.: Quality evaluation of agile processes: Measurement of requirements management using AQF v2. In: QUATIC 2018, pp. 15–20. IEEE (2018).
- [6] Pinto, N.S., Tortosa, N., Cabas Geat, B., Ibáñez, L., Acuña, C.J.: Validación de la reingeniería aplicada sobre la primera versión de Agile Quality Framework. *Electronic Journal of SADIO* 18(1), 93–109 (2019).
- [7] Wooldridge, M., Jennings, N.R., Kinny, D.: A methodology for agent-oriented analysis and design. In: *Autonomous Agents 1999*, pp. 69–76. ACM (1999).
- [8] Curcio, K., Navarro, T., Malucelli, A., Reinehr, S.: Requirements engineering: A systematic mapping study in agile software development. *Journal of Systems and Software* 139, 32–50 (2018).
- [9] Bjarnason, E., Lang, F., Mjöberg, A.: An empirically based model of software prototyping: a mapping study and a multi-case study. *Empirical Software Engineering* 28, 115 (2023).
- [10] Hoy, Z., Xu, M.: Agile Software Requirements Engineering Challenges-Solutions—A Conceptual Framework from Systematic Literature Review. *Information* 14(6), 322 (2023).
- [11] Paetsch, F., Eberlein, A., Maurer, F.: Requirements engineering and agile software development. In: *WET ICE 2003*, pp. 308–313. IEEE (2003).
- [12] Ramesh, B., Cao, L., Baskerville, R.: Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal* 20(5), 449–480 (2010).
- [13] Sommerville, I.: *Ingeniería del Software*. 10^a ed. Pearson (2016).
- [14] IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society (2011).
- [15] Cohn, M.: *User Stories Applied: For Agile Software Development*. Addison-Wesley (2004).
- [16] Hallmann, D.: “I Don’t Understand!”: Toward a Model to Evaluate the Role of User Story Quality. In: *XP 2020, LNBP 383*, pp. 103–112. Springer (2020).
- [17] Jennings, N.R., Sycara, K., Wooldridge, M.: A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems* 1(1), 7–38 (1998).
- [18] Peixoto, M., Silva, C., Araújo, J., Gorschek, T., Vasconcelos, A., Vilela, J.: Evaluating a privacy requirements specification method by using a mixed-method approach. *Requirements Engineering* 28(2), 229–255 (2023).
- [19] Kolp, M., Giorgini, P., Mylopoulos, J.: Multi-agent architectures as organizational structures. *Autonomous Agents and Multi-Agent Systems* 13(1), 3–25 (2006).
- [20] Herwanto, G.B., Quirchmayr, G., Tjoa, A.M.: Leveraging NLP techniques for privacy requirements engineering in user stories. *IEEE Access* 12, 22167–22189 (2024).
- [21] Arulmohan, S., Meurs, M. J., Mosser, S.: Extracting domain models from textual requirements in the era of large language models. In: *2023 ACM/IEEE international conference on model driven engineering languages and systems companion (MODELS-C)*, pp. 580–587. IEEE (2023).
- [22] Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., Awadallah, A.H., White, R.W., Burger, D., Wang, C.: AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversations. In: *Proceedings of the First Conference on Language Modeling (COLM 2024)*. (2024). <https://openreview.net/forum?id=BAakY1hNKS>
- [23] LangChain: Building applications with LLMs through composability. <https://www.langchain.com> (2024).
- [24] Tomaselli, G.P., Pinto, N.S., Acuña, C.: Requirements management methods and practices for improving the quality of agile software development processes: a review of the literature. *Requirements Engineering* (2025). <https://doi.org/10.1007/s00766-025-00448-3>

JCSA 2025

Jornadas de Calidad de Software y Agilidad



UNIVERSIDAD
TECNOLÓGICA NACIONAL
FACULTAD REGIONAL
RESISTENCIA



Universidad Nacional del Nordeste
Facultad de Ciencias
Exactas y Naturales y Agrimensura



Universidad Nacional de Misiones
Facultad de Ciencias
Exactas, Químicas y Naturales