Revista Colombiana de Computación
Vol. 27, No. 1. Enero – Junio 2026, pp. 23-40
e-ISSN: 2539-2115, https://doi.org/10.29375/25392115.5620

REVISTA COLOMBIANA DE COMPUTACIÓN

Aprender haciendo tiene un costo: Patrones organizacionales y desviaciones del *SDLC* en un semillero universitario de software

Learning by doing has a cost: Organizational patterns and SDLC deviations in a university software research group

Miguel Ángel Becerra Solano[1], Jheison Alexander De la Rosa Pineda[1], Angie Alejandra Hernández León[1], Michael Fernney Pérez Ochoa[1], Crisóstomo Barajas-Solano[1,2]

[1] Semillero Máster Digital, Universidad de Investigación y Desarrollo – UDI.
[2] Director Máster Digital, Universidad de Investigación y Desarrollo – UDI, Bucaramanga, Colombia.

mbecerra3@udi.edu.co, jdelarosa1@udi.edu.co, ahernandez46@udi.edu.co, mperez27@udi.edu.co, cbarajas4@udi.edu.co

(Recibido: 1 Abril 2026; aceptado: 25 Junio 2026; publicado online: 30 Junio 2026)

Resumen. Los semilleros de investigación universitarios combinan objetivos pedagógicos con desarrollo real de software, introduciendo tensiones organizacionales que el *SDLC* estándar no contempla. Este trabajo presenta un análisis retrospectivo del desarrollo de la versión *1.0* del Aplicativo del Banco de Ideas – ABI, desarrollado por el semillero Máster Digital de la Universidad de Investigación y Desarrollo – UDI, Bucaramanga, Colombia. A partir de evidencia documental –reportes semanales, artefactos de diseño, registros *Git* y actas– se identificaron cinco dimensiones de desviación recurrente: gobernanza y dirección académica, gestión de proyecto y coordinación, coherencia metodológica y definición temprana del *stack* tecnológico, documentación y gestión del conocimiento, y responsabilidad individual. La ausencia de mecanismos formales de gobernanza generó un atraso acumulado de entre siete y nueve meses, concentrando la carga operativa en un subconjunto reducido de integrantes. Se proponen recomendaciones organizacionales adaptadas a entornos formativos para fortalecer la trazabilidad, continuidad técnica y estabilidad operativa de proyectos similares en semilleros universitarios latinoamericanos.

Palabras clave: Semilleros de investigación, ingeniería de software, lecciones aprendidas, gobernanza, trazabilidad.

Abstract. University research seedbeds combine pedagogical objectives with real software development, introducing organizational tensions that standard SDLC frameworks do not address. This paper presents a retrospective analysis of the development of version *1.0* of the Ideas Bank Application – ABI, built by the *Máster Digital* research group at the Universidad de Investigación y Desarrollo – UDI, Bucaramanga, Colombia. Drawing on documentary evidence –weekly reports, design artifacts, Git records, and meeting minutes– five dimensions of recurrent deviation were identified: academic governance and direction, project management and coordination, methodological coherence and early stack definition, documentation and knowledge management, and individual accountability. The absence of formal governance and knowledge transfer mechanisms produced a cumulative delay of seven to nine months, concentrating the operational workload on a small subset of team members. Organizational recommendations adapted to formative environments are proposed to strengthen traceability, technical continuity, and operational stability in similar projects across Latin American university research groups.

Keywords: Research seedbeds, software engineering, lessons learned, governance, traceability.

1 Introducción

Los semilleros de investigación constituyen un mecanismo ampliamente utilizado en universidades latinoamericanas para vincular a estudiantes de pregrado con procesos de investigación aplicada y de desarrollo tecnológico (Flores y otros, 2019; Quintero Corzo y otros, 2008). En el área de ingeniería de software, estos espacios permiten participar en proyectos reales de desarrollo como parte de la formación investigativa (Brew, 2006; Healey y Jenkins, 2009; Lopatto, 2007). Sin embargo, las condiciones estructurales de estos entornos –heterogeneidad técnica, disponibilidad fragmentada y rotación frecuente entre cohortes– no son defectos ocasionales del modelo formativo: son su característica definitoria, y precisamente eso convierte a los semilleros en un objeto de estudio organizacional subestimado.

La ingeniería de software dispone de marcos maduros para estructurar el desarrollo de sistemas: el *SDLC* – *Software Development Life Cycle*, ciclo de vida de desarrollo de software– organiza el proceso en fases de planificación, diseño, implementación, validación y mantenimiento (Sommerville, 2016). Estos marcos asumen

©2026 Universidad Autónoma de Bucaramanga. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY-NC-SA 4.0) license (https://creativecommons.org/licenses/by-nc-sa/4.0/)

condiciones que los entornos formativos rara vez cumplen: equipos estables, roles definidos y mecanismos formales de transferencia de conocimiento (Iacob y Faily, 2019; Vanhanen y Lehtinen, 2014). Estudios previos han documentado dificultades de coordinación en proyectos académicos de software (Iacob y Faily, 2019; Vanhanen y Lehtinen, 2014), la importancia de la documentación técnica (Forward y Lethbridge, 2002; Robillard, 1999) y la trazabilidad de decisiones arquitectónicas (Tyree y Akerman, 2005), pero sin abordar sistemáticamente cómo las condiciones propias de los semilleros se traducen en desviaciones respecto a fases concretas del *SDLC*. La dimensión técnica de estos proyectos ha sido igualmente subestimada: Fenton y Bieman (2014) establecen la importancia de las métricas de proceso y producto como mecanismos de objetivación de la calidad técnica, mientras que el concepto de deuda técnica –introducido por Cunningham (1992) y sistematizado por Kruchten y otros (2012)– ofrece un marco analítico para entender los costos diferidos de las decisiones de diseño tomadas bajo presión de tiempo, condición estructural en los semilleros universitarios. Zazworka y otros (2011) documentan que la presión de entrega y la rotación de equipos son los predictores más fuertes de acumulación de deuda técnica en proyectos académicos de software, independientemente del nivel de experiencia de los desarrolladores.

Este trabajo analiza el desarrollo de la versión *1.0* del Aplicativo del Banco de Ideas – ABI, construido por el semillero Máster Digital de la Universidad de Investigación y Desarrollo – UDI, Bucaramanga, Colombia. El caso reúne condiciones analíticamente adecuadas: se desarrolló durante múltiples periodos académicos, experimentó un atraso acumulado de entre siete y nueve meses, y generó evidencia documental suficiente para una reconstrucción retrospectiva sistemática. Más que un caso de fracaso, es un caso de aprendizaje: la versión *1.0* fue estabilizada y el proyecto continúa activo en su versión *1.5*, lo que permite analizar las desviaciones sin el sesgo de un proyecto abandonado.

Los resultados se organizan en cinco dimensiones: gobernanza y dirección académica, gestión de proyecto y coordinación, coherencia metodológica y definición temprana del *stack* tecnológico, documentación y gestión del conocimiento, y responsabilidad individual. El análisis de cada dimensión incluye su efecto sobre la calidad técnica del sistema desarrollado, sistematizado en la Sección 4.6 mediante una tabla de trazabilidad entre desviaciones organizacionales y consecuencias técnicas concretas sobre el producto de software. El resto del artículo se organiza así: Sección 2 presenta el contexto; Sección 3 la metodología; Sección 4 discute las dimensiones contrastándolas con el *SDLC*; Sección 5 propone recomendaciones; Sección 6 presenta conclusiones.

2 Contexto del proyecto

2.1 El semillero como entorno organizacional

El semillero de investigación Máster Digital, adscrito al programa de Ingeniería de Sistemas de la UDI, contaba a diciembre de 2025 con 35 estudiantes organizados en cuatro niveles, S1-S4, según su permanencia y participación activa dentro del grupo. Los estudiantes S1 corresponden a miembros recientemente vinculados; el paso de un nivel al siguiente depende de la asistencia sostenida y el compromiso demostrado durante cada semestre académico. Los estudiantes S4 cuentan con al menos cuatro semestres continuos de participación y asumen progresivamente funciones de liderazgo técnico y acompañamiento formativo sobre los niveles inferiores. Sin embargo, el semillero ha contado con la participación de 48 estudiantes, considerando las bajas que se han dado a lo largo de la ventana de observación.

Esta arquitectura multinivel es constitutiva del modelo formativo: la coexistencia de estudiantes con trayectorias técnicas y niveles de experiencia muy distintos es una condición de operación permanente, no una anomalía circunstancial (Healey y Jenkins, 2009; Lopatto, 2007). En entornos de este tipo, la dinámica organizacional tiende a estructurarse como una comunidad de práctica (Lave y Wenger, 1991; Wenger, 1998), en la que el conocimiento técnico y metodológico circula principalmente a través de la interacción entre pares y la observación de prácticas existentes, más que mediante mecanismos formales de transferencia.

El semillero se reunía semanalmente los sábados entre las 2:00 p.m. y las 6:00 p.m., complementando este espacio con trabajo individual estimado entre cuatro y seis horas adicionales durante la semana –una dedicación aproximada de 8 a 10 horas semanales por estudiante. Esta disponibilidad es fragmentada y asíncrona: el sábado constituye el único punto formal de sincronización colectiva, mientras que el trabajo restante se desarrolla en momentos dispersos e independientes entre sí.

Una tercera condición estructural, complementaria a la rotación entre cohortes y a la disponibilidad fragmentada, es la heterogeneidad técnica del equipo. La arquitectura multinivel implica que, en un mismo proyecto, conviven integrantes que apenas se inician en programación, S1, con otros que ya dominan el *framework* de trabajo, los patrones de diseño y las prácticas de control de versiones, S3-S4. Esta disparidad no se distribuye uniformemente entre módulos: cada equipo de trabajo reunía perfiles técnicos dispares, lo que favorecía criterios

de implementación divergentes, estilos de código inconsistentes y una capacidad desigual para asumir tareas de integración. A diferencia de la rotación –que compromete la continuidad en el tiempo– y de la disponibilidad –que limita el tiempo efectivo de trabajo–, la heterogeneidad técnica afecta la coherencia simultánea del producto, y su efecto se materializa en las dimensiones organizacionales analizadas en la Sección 4 –especialmente metodología y *stack*, y responsabilidad individual–.

2.2 El problema institucional: Gestión del banco de ideas

En paralelo a estas condiciones organizacionales, la Universidad de Investigación y Desarrollo enfrentaba un problema operativo en la gestión de su banco de ideas de proyectos de grado. Antes del desarrollo de ABI, la Dirección de Investigaciones – DI administraba este proceso de manera predominantemente manual.

Las propuestas de ideas eran enviadas por docentes y estudiantes en documentos PDF. El personal administrativo de la DI debía consolidar estos archivos, distribuirlos a los programas académicos para su evaluación, recibir conceptos de aprobación o rechazo, asignar estudiantes y directores a los proyectos, y elaborar reportes para las directivas institucionales. Las herramientas de gestión utilizadas eran principalmente hojas de cálculo en *MS Excel* o *Google Sheets*.

Este esquema implicaba una alta carga operativa para el personal administrativo, trazabilidad limitada sobre el estado de las propuestas y una fuerte dependencia del procesamiento manual para la consolidación de información. La ausencia de una plataforma centralizada dificultaba el seguimiento institucional de las ideas propuestas, su evaluación y la asignación posterior de estudiantes y directores. Estas limitaciones motivaron la iniciativa de desarrollar una plataforma digital que sistematizara y automatizara el proceso.

La Figura 1 ilustra el flujo completo de este proceso y la centralidad operativa de la DI en cada una de sus etapas.

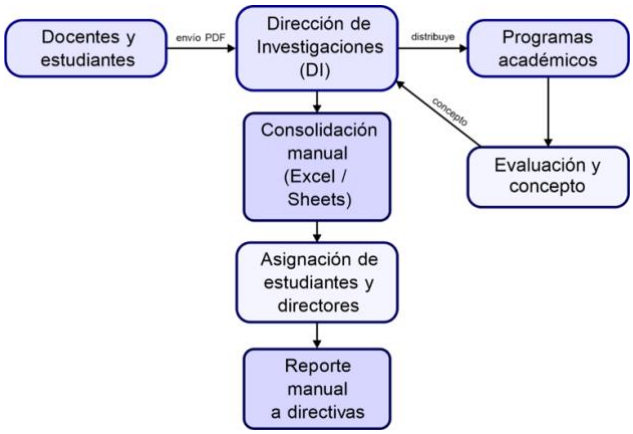


Figura 1. Proceso manual de gestión del banco de ideas previo al desarrollo de ABI. La DI operaba como nodo central de consolidación, distribución y seguimiento, con dependencia del procesamiento manual en cada etapa

2.3 El sistema ABI: Alcance, actores y marco de desarrollo

En este contexto organizacional e institucional, el semillero asumió el diseño e implementación de un aplicativo web orientado a sistematizar la gestión del banco de ideas de proyectos de grado. Este sistema fue denominado ABI –Aplicativo del Banco de Ideas– y su desarrollo inició en 2024. El alcance funcional de la versión *1.0* comprendía:

- Recepción de propuestas de ideas por parte de docentes y estudiantes
- Gestión del proceso de evaluación y aprobación institucional
- Postulación de estudiantes a ideas disponibles
- Asignación de estudiantes y directores de proyectos aprobados
- Generación de reportes institucionales

Los actores formales del proyecto eran tres: la DI en el rol de cliente y *Product Owner* institucional, la dirección del semillero como responsable técnico del desarrollo, y los estudiantes organizados en equipos según su nivel, S1-S4, con los integrantes de niveles superiores asumiendo funciones de coordinación interna.

Desde el punto de vista metodológico, el desarrollo de ABI se planteó bajo un modelo incremental (Sommerville, 2016), con inspiración en prácticas ágiles similares a Scrum (Schwaber y Sutherland, 2020), e

iteraciones previstas de una a dos semanas. La adopción de metodologías ágiles en proyectos de software académicos ha sido documentada como una estrategia pedagógicamente pertinente para entornos formativos (Mahnic, 2012; Paasivaara y otros, 2014), en la medida en que permite alinear los ciclos de entrega con los periodos académicos y fomentar prácticas colaborativas entre estudiantes con distintos niveles de experiencia.

El proyecto generó evidencia documental continua a lo largo de su ejecución, cuyas fuentes y esquema de triangulación se describen en la Sección 3. La versión *1.0* fue estabilizada y el proyecto continúa activo en su versión *1.5*, lo que hace del caso ABI un objeto de análisis con trayectoria completa y documentada.

2.4 Arquitectura e implementación técnica

El sistema ABI fue desarrollado sobre una arquitectura *MVC –Model-View-Controller–* implementada en Laravel *10*, *PHP 8.1+*, con *MySQL/MariaDB* como sistema gestor de base de datos relacional. La capa de presentación combina plantillas *Blade* con *Bootstrap 5.3.1* y el *framework* de componentes *Tablar*; la capa de interactividad incorpora *jQuery*, *ApexCharts* para visualización de datos y *FilePond* para gestión de archivos adjuntos. El procesamiento de documentos es soportado por las bibliotecas *DomPDF, TCPDF* y *PhpSpreadsheet*. El control de autenticación y autorización se implementó mediante Laravel Sanctum, con protección *CSRF* y *middleware* de roles. El entorno de despliegue utiliza *Railway* como plataforma en la nube para el ambiente de producción y *XAMPP* para los ambientes locales de desarrollo.

La arquitectura lógica del sistema sigue el flujo estándar del *framework*: la solicitud del usuario es recibida por el enrutador de Laravel, procesada por el controlador correspondiente, que interactúa con los modelos *Eloquent ORM* para acceder a la base de datos, y retorna la respuesta a través de las vistas Blade. Esta separación de responsabilidades fue adoptada como decisión de diseño desde la fase de análisis del proyecto, aunque –como se documenta en la Sección 4– su implementación práctica presentó desviaciones atribuibles a las condiciones organizacionales del semillero.

2.4.1 Módulos funcionales

La versión *1.0* del sistema ABI comprende nueve módulos funcionales: 1) Gestión de usuarios y perfiles; 2) banco de ideas y registro de propuestas; 3) evaluación de comité; 4) versionamiento e historial de proyectos; 5) gestión de *frameworks* investigativos; 6) reportes institucionales; 7) módulo de postulaciones; 8) gestión de carga docente y demanda de ideas; y 9) aseguramiento de calidad y corrección de errores –*Quality Assurance*, *QA*–. Cada módulo fue desarrollado por un equipo de entre tres y seis estudiantes, con un integrante S3-S4 como líder técnico responsable de la integración y revisión por *pull requests*.

2.4.2 Roles y control de acceso

El sistema implementa un modelo de cuatro roles con permisos diferenciados: *student*, crear y consultar propuestas; *profesor*, proponer y asesorar proyectos; *committee_leader*, evaluar propuestas ante el comité; y *research_staff*, administrar el sistema, gestionar carga docente y generar reportes institucionales. La matriz de permisos fue formalizada como entregable de la fase de análisis y operacionalizada mediante *middleware* de rol en *Laravel*.

2.4.3 Modelo de datos

La base de datos relacional del sistema organiza sus entidades en cuatro dominios: a) usuarios – tablas *users*, *students, professors, research_staff*; b) académico –*departments*, *cities*, *programs*, *research_groups*, *investigation_lines, thematic_areas*–; c) proyectos –*projects*, *versions*, *content_version*, *project_statuses*–; y d) marcos investigativos – *frameworks*, *contents*, *content_frameworks*–. Las relaciones principales incluyen $1:N$ entre *projects* y *versions*, y $N:N$ entre proyectos, estudiantes y profesores. La tabla *versions* –rediseñada durante el desarrollo de la versión *1.0*– incorpora los campos *snapshot* –*JSON*– y *created_by_user_id* para soportar el historial inmutable de cambios requerido por los requisitos funcionales RF04 y RF05.

3 Metodología

3.1 Diseño del estudio

Este trabajo adopta un diseño de estudio de caso instrumental cualitativo (Stake, 1995; Yin, 2014), orientado a la generación de conocimiento transferible a partir de un caso particular con condiciones organizacionales representativas de entornos formativos similares. A diferencia de los estudios de caso descriptivos, el caso ABI no se analiza como fin en sí mismo sino como vehículo para comprender un fenómeno más amplio: las tensiones estructurales entre las dinámicas propias de los semilleros de investigación y las prácticas esperadas del *SDLC*.

Durante los dos años de desarrollo de ABI *v1.0* participaron en el semillero Máster Digital un total acumulado de 48 estudiantes, de los cuales algunos se retiraron de manera voluntaria del semillero. Al final de la ventana de observación se contaba con 35 estudiantes con aval semestral activo, organizados en los cuatro niveles de permanencia, S1-S4, descritos en la Sección 2.1 y cuya composición longitudinal se documenta en la Figura 2. Además del equipo estudiantil, participaron un docente en el rol de director del semillero y, por parte de la Dirección de Investigaciones – DI, como contraparte institucional, dos integrantes del nivel operativo y la directora de la DI de entonces, esta última mediante entrevistas cortas pero sustanciosas en hitos específicos del proyecto.

Una caracterización nominal exhaustiva de los 35 estudiantes durante los dos años del proyecto no resulta apropiada en este estudio por tres razones que conviene hacer explícitas. Primero, una razón ética: codificar nominalmente la trayectoria académica de estudiantes que no participaron en la sistematización analítica ni firmaron este manuscrito implicaría usar sus datos como insumo de investigación sin que hayan otorgado consentimiento informado para ese uso específico. Segundo, una razón de trazabilidad retrospectiva: el semillero no operó con un sistema formal de registro nominal de actividades por estudiante; los reportes semanales eran insumos de avance del proyecto, no instrumentos diseñados para una reconstrucción biográfica posterior, y reconstruir esa información dos años después introduciría reconstrucciones de memoria con alto riesgo de error. Tercero, una razón de pertinencia analítica: la unidad de análisis del estudio no es el individuo sino los patrones organizacionales del proyecto, y una caracterización nominal de los 35 estudiantes desplazaría el foco hacia un dato que no se moviliza en el resto del manuscrito.

En consecuencia, la codificación nominal se acota a los participantes que sí intervinieron directamente en la sistematización analítica que sustenta este trabajo: los cuatro estudiantes coautores –que cumplen tres condiciones simultáneas: estuvieron activos en los cuatro semestres del proyecto, ejercieron roles de liderazgo técnico durante los niveles S3 y S4, y realizaron la sistematización retrospectiva–, el director del semillero, y la contraparte institucional de la DI que participaron en la validación cruzada de los hallazgos. La Tabla 1 sintetiza este esquema de codificación.

Tabla 1. Esquema de codificación de los participantes en la sistematización analítica

Código	Periodo activo	Tipología	Rol en el proyecto	Rol en el análisis
E-01	2024-1 a 2025-2	Estudiante –S4 al cierre–	Líder técnico de codificación	Coautor / sistematizador
E-02	2024-1 a 2025-2	Estudiante –S4 al cierre–	Líder técnico de base de datos	Coautor / sistematizador
E-03	2024-1 a 2025-2	Estudiante –S4 al cierre–	Líder técnico de análisis y *QA*	Coautor / sistematizador
E-04	2024-1 a 2025-2	Estudiante –S4 al cierre–	Líder técnico de análisis y *QA*	Coautor / sistematizador
D-01	2024-1 a 2025-2	Docente	Director del semillero	Coautor / sistematizador
DI-01	2024-1 y 2025-2	Administrativo institucional	Contraparte operativa de la DI	Validación cruzada externa, 3.3

La Figura 2 ilustra la dinámica de participación del semillero durante los cuatro semestres de desarrollo de ABI *v1.0*. El panel a) muestra la composición del equipo por nivel de permanencia, S1-S4, al cierre de cada semestre, evidenciando la progresión de la cohorte fundacional desde S1 2024-1 hasta S4 2025-2. El panel b) presenta el número de estudiantes con aval semestral activo y el número de abandonos por semestre. En conjunto, ambos paneles documentan las dos condiciones estructurales del semillero descritas en la Sección 2.1 –rotación entre cohortes y disponibilidad fragmentada– y permiten dimensionar el universo de participantes del que se deriva la evidencia analizada en este trabajo.

La selección del caso responde a tres condiciones que lo hacen analíticamente adecuado: 1) Se desarrolló a lo largo de cuatro semestres académicos consecutivos, 2024-2025, lo que permitió observar patrones recurrentes más allá de incidentes aislados; 2) generó evidencia documental continua y sistemática a lo largo de su ejecución; y 3) alcanzó una versión funcional estable, lo que permite analizar las desviaciones sin el sesgo interpretativo de un proyecto abandonado.

La transferibilidad de los hallazgos no se sustenta en representatividad estadística sino en la similitud estructural entre el caso estudiado y otros semilleros universitarios latinoamericanos: equipos estudiantiles con rotación entre cohortes, disponibilidad fragmentada, heterogeneidad técnica y ausencia de mecanismos formales de gobernanza. En la medida en que estas condiciones se replican en otros contextos formativos, las lecciones identificadas conservan pertinencia analítica más allá del caso particular (Yin, 2014).

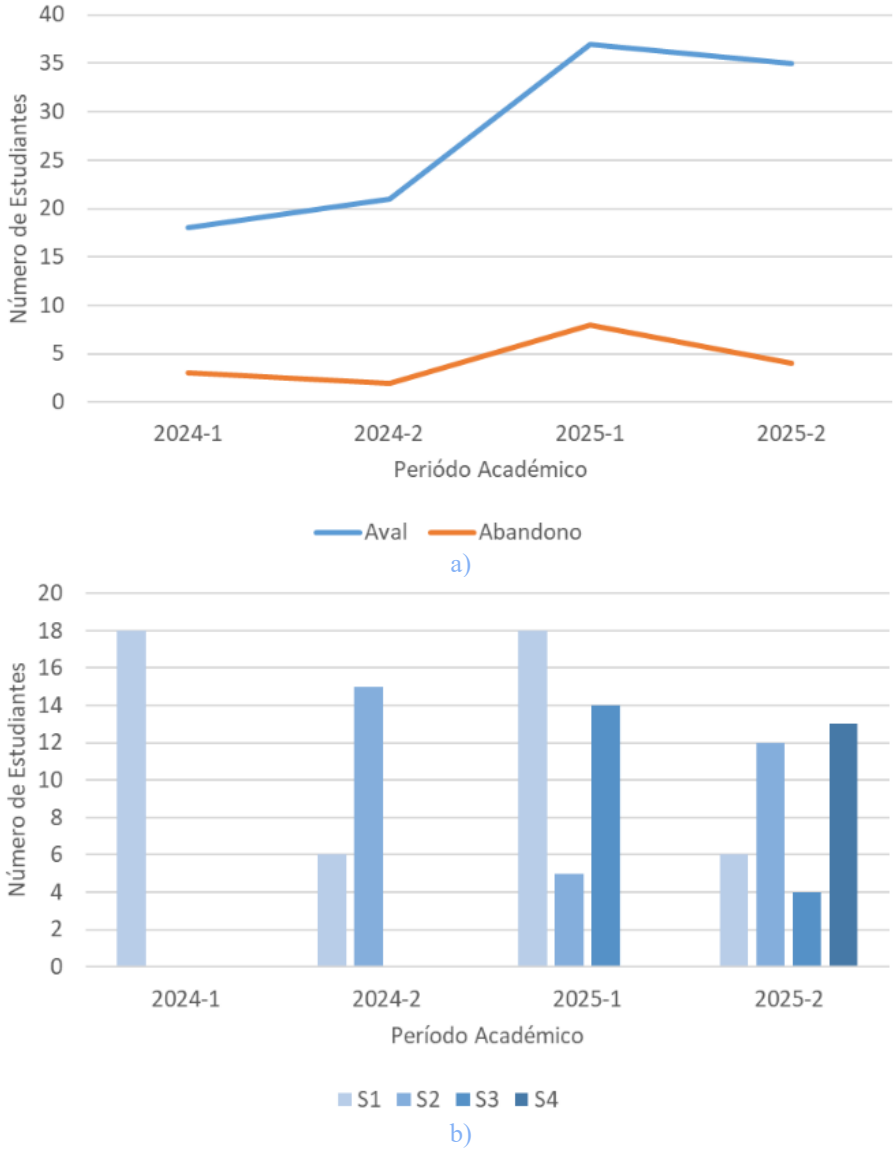


Figura 2. Dinámica de participación en el semillero Máster Digital durante el desarrollo de ABI *v1.0*. a) Composición del equipo por nivel de permanencia, S1-S4**,** al cierre de cada semestre. b) Estudiantes con aval semestral activo y abandonos por semestre.

3.2 Fuentes de evidencia y triangulación

La principal fuente de información fueron las plataformas institucionales *Google Classroom* y *Google Drive*, utilizadas semanalmente por los miembros del semillero para reportar las actividades asignadas durante las reuniones presenciales de los sábados. En dichos reportes, cada estudiante consignaba el detalle de las tareas realizadas, las dificultades encontradas y el estado de avance de los compromisos adquiridos. Paralelamente, se recopilaron evidencias técnicas asociadas a cada *sprint*: *commits* en repositorios *Git*, capturas de interfaces, diagramas *UML* y documentos de especificación.

Como insumo adicional para la sistematización analítica, los cuatro estudiantes coautores –con la contribución documentada de los nueve estudiantes S4 restantes del semestre 2025-2 bajo su dirección– elaboraron al cierre del proyecto un documento colectivo de lecciones aprendidas, que consolida retrospectivamente la perspectiva del equipo sobre el desarrollo de ABI *v1.0*. Todas las citas textuales que se presentan entrecomilladas en la Sección 4

provienen de este documento colectivo y, por su naturaleza grupal y la firma compartida de los autores, no se atribuyen a un autor individual.

La unidad de análisis fue cada iteración de trabajo –*sprint*– y los hitos asociados al cronograma del proyecto. Al cierre de uno o dos *sprints*, o ante el vencimiento de un hito relevante, se realizaba una revisión global del estado del proyecto mediante el contraste de tres fuentes complementarias, lo que permitió una triangulación básica entre discurso, planificación y ejecución real. La Tabla 2 sintetiza las fuentes utilizadas y su rol en este esquema de triangulación.

Tabla 2. Fuentes de evidencia y esquema de triangulación

Fuente	Tipo de evidencia	Frecuencia	Rol en la triangulación
Google Classroom / Drive	Reportes escritos de avance, dificultades y compromisos	Semanal	Discurso declarado del equipo
Repositorio *Git*	*Commits*, ramas, historial de cambios	Continua	Ejecución técnica real
Artefactos de diseño	Diagramas *UML*, modelos de datos, especificaciones	Por hito	Coherencia entre diseño e implementación
Sustentaciones orales	Exposiciones presenciales de avance	Cada 1-2 *sprints*	Contraste entre discurso oral y evidencia escrita
Actas de reunión	Decisiones, compromisos y observaciones del director	Semanal	Trazabilidad de decisiones organizacionales

El director del semillero, con el apoyo del equipo interno de *QA*, efectuaba una medición global del avance en cada *sprint*, mientras que el seguimiento detallado de tareas recaía principalmente en estudiantes de niveles superiores, S3-S4, responsables de los diferentes equipos de trabajo. Se llevó un registro continuo del número de semanas de atraso acumuladas respecto a la planeación inicial y sus posteriores ajustes, aunque sin implementar una métrica formal de complejidad organizacional.

3.3 Criterios de análisis, sesgo y validación

La sistematización explícita de lecciones aprendidas se llevó a cabo al finalizar el periodo académico 2025-2, a partir de la revisión integral de las fuentes descritas en la sección anterior. Para orientar este proceso, se estableció previamente –antes de iniciar la revisión retrospectiva– un criterio operacional para distinguir patrones estructurales de incidentes circunstanciales: se consideró que un evento negativo constituía una lección crítica cuando cumplía al menos uno de los siguientes criterios, definidos al inicio de la sistematización: 1) se repetía en múltiples iteraciones del proyecto, o 2) generaba un atraso apreciable, definido como tres o más semanas adicionales respecto al cronograma previsto.

Dado que los autores participaron activamente en el desarrollo del proyecto –los cuatro estudiantes coautores– E-01 a E-04 en la Tabla 1, como líderes técnicos de los equipos de codificación, base de datos, análisis y *QA*, y –el director del semillero– D-01, como responsable académico, el proceso de sistematización incorporó dos mecanismos para gestionar el sesgo de autoevaluación. En primer lugar, las lecciones fueron construidas mediante revisión cruzada entre todos los autores del manuscrito, contrastando interpretaciones individuales antes de consolidar cada categoría. En segundo lugar, al concluir la sistematización, los resultados fueron socializados con la Dirección de Investigaciones – DI mediante un conversatorio, en el que participó la contraparte operativa codificada como DI-01, y en el que se presentó el proceso de desarrollo de la versión *1.0* y las lecciones identificadas. En su condición de cliente institucional del sistema ABI, la DI aportó una perspectiva externa que permitió contrastar la interpretación del equipo con la percepción del actor que recibió el producto, contribuyendo a matizar algunas de las lecciones consolidadas.

El procedimiento de consolidación de categorías analíticas siguió tres pasos secuenciales. En primer lugar, cada autor revisó de forma independiente el corpus documental completo e identificó eventos candidatos a lección crítica según los criterios operacionales definidos. En segundo lugar, se realizó una sesión de contraste entre todos los autores en la que cada categoría propuesta fue presentada con su evidencia de respaldo; las discrepancias –definidas como aquellos eventos en los que al menos un autor no identificaba cumplimiento de los criterios– se resolvieron por consenso explícito, documentando el argumento que determinó la inclusión o exclusión de cada caso. En tercer lugar, las categorías consolidadas fueron presentadas a la DI como actor externo al equipo de desarrollo, cuya perspectiva como cliente institucional permitió contrastar la interpretación interna con la percepción del receptor del producto. Este proceso produjo ajustes en la formulación de dos lecciones de la dimensión de gobernanza, que fueron moderadas a partir de la retroalimentación de la DI.

4 Discusión: patrones organizacionales y desviaciones del *SDLC*

Las lecciones identificadas durante el desarrollo de la versión *1.0* de ABI no se interpretan aquí como incidentes aislados propios de un equipo inexperto, sino como manifestaciones de tensiones estructurales entre las condiciones organizacionales del semillero y las prácticas que el *SDLC* asume como dadas. En esta sección cada dimensión se analiza en tres capas: el patrón observado en la evidencia documental, la voz directa de los estudiantes que lo vivieron –presentada mediante citas textuales que provienen del documento colectivo de lecciones aprendidas descrito en la Sección 3.2 y que, por su naturaleza grupal, no se atribuyen a un autor individual–, y su implicación respecto a una fase concreta del ciclo de vida del software, contrastada con la literatura relevante. Donde las dimensiones interactúan entre sí, esa interacción se hace explícita.

Las cinco dimensiones se presentan en el siguiente orden: gobernanza y dirección académica, gestión de proyecto y coordinación, coherencia metodológica y definición temprana del *stack* tecnológico, documentación y gestión del conocimiento, y responsabilidad individual.

4.1 Gobernanza y dirección académica

La ausencia de una estructura de gobernanza explícita fue el patrón de mayor alcance sistémico observado en el desarrollo de ABI. Durante las etapas iniciales del proyecto no existía una definición formal de roles, responsabilidades, entregables, lineamientos, procedimientos ni criterios de aceptación. La dirección académica recaía en el facilitador del semillero, mientras que las decisiones operativas emergían de manera distribuida y sin autoridad claramente asignada. Este vacío se manifestó en ambigüedad recurrente sobre quién debía tomar determinadas decisiones técnicas y en incertidumbre sobre el alcance de los entregables en cada fase. El patrón fue reportado de forma consistente a lo largo de los cuatro semestres del proyecto y está documentado en actas, reportes de atraso y sustentaciones orales. En el plano de la gobernanza, esta dimensión comprende la exigencia formal de que todo lineamiento, procedimiento o marco de trabajo –incluido el marco tecnológico– quede definido de manera precisa desde el inicio; la definición técnica concreta del *stack*, en cambio, se aborda en la dimensión de metodología de desarrollo –Sección 4.3–.

4.1.1 Voz del equipo

A continuación, se presentan las afirmaciones que realiza el equipo:

- “Los entregables deben quedar completamente definidos por los docentes y presentados a los estudiantes desde el comienzo.”
- “También es fundamental definir claramente las fases y las actividades del desarrollo desde el inicio.”
- “Cada fase debería contar con una evaluación final ante el comité, con el fin de verificar si cumple con los estándares mínimos de calidad establecidos.”
- “Definir claramente los criterios de aceptación de cada uno de los entregables y de las actividades, fases y documentaciones.”
- “Si dentro del proyecto se elaboran marcos, como el marco tecnológico, es fundamental definirlos de manera precisa.”
- “Dado el creciente número de estudiantes que participan en el semillero, es necesario fortalecer el acompañamiento docente.”

Las afirmaciones anteriores convergen en una preocupación compartida: la claridad normativa del proceso no puede asumirse como implícita en un entorno formativo; debe construirse deliberadamente desde el inicio.

4.1.2 Impacto sobre el SDLC

En el *SDLC* estándar, la fase de planificación establece precisamente estos elementos: estructura de gobernanza, definición de roles, alcance de entregables y criterios de aceptación (Sommerville, 2016). Su ausencia no es un problema menor de organización interna; es una desviación respecto a la fase que condiciona todas las demás. Iacob y Faily (2019) documentan un fenómeno análogo en proyectos grupales de pregrado: la brecha entre las expectativas de los estudiantes sobre la estructura del proyecto y la realidad operativa que encuentran genera fricciones que se amplifican a medida que el proyecto avanza. En ABI esta amplificación fue observable: la ausencia de gobernanza temprana no solo generó ambigüedad en la planificación, sino que contribuyó directamente a las dificultades de documentación y trazabilidad reportadas en la dimensión de documentación y

gestión del conocimiento; dado que sin una estructura de gobernanza que lo exigiera, el registro sistemático de decisiones nunca adquirió carácter obligatorio. Esta interacción entre dimensiones se retoma en la subsección correspondiente.

4.2 Gestión de proyecto y coordinación

La coordinación operativa del proyecto dependió casi exclusivamente de interacciones informales durante las reuniones de los sábados e intercambios *ad hoc* entre estudiantes durante la semana. En ausencia de mecanismos sistemáticos para registrar el avance de tareas, documentar decisiones operativas y sincronizar el trabajo entre equipos, la visión compartida del estado del proyecto era fragmentaria. El patrón fue recurrente a lo largo de los cuatro semestres: en múltiples iteraciones se reportaron inconsistencias entre componentes desarrollados por distintos integrantes, que obligaron a ajustes posteriores de integración. La disponibilidad fragmentada y asíncrona del semillero –descrita en la Sección 2.1– actuó como factor amplificador: sin un punto de sincronización distinto al sábado, los desajustes podían acumularse durante días antes de ser detectados.

4.2.1 Voz del equipo

A continuación, se presentan las afirmaciones que realiza el equipo:

- "Cuando un grupo de estudiantes trabaja en un mismo proyecto, es fundamental establecer reuniones periódicas que permitan unificar criterios y evitar la dispersión de la información."
- "Todas las actividades, reuniones y decisiones deberían quedar registradas en actas, incluyendo fechas, responsables y validaciones."
- "Para fortalecer el compromiso, sería necesario ser más estrictos con la asistencia a las reuniones de los sábados."
- "Se debe definir desde un principio si los estudiantes deben contar con disponibilidad para actividades adicionales durante la semana."
- "Cuando los equipos superan los seis integrantes, se dificulta la coordinación y el seguimiento del aporte individual."

La última afirmación es especialmente significativa: no apunta a una falla de actitud sino a un problema de escala. Con 35 estudiantes organizados en equipos, la coordinación informal deja de ser viable por razones estructurales, independientemente del compromiso individual.

4.2.2 Impacto sobre el *SDLC*

En el *SDLC*, la fase de ejecución presupone mecanismos activos de seguimiento del progreso y sincronización de tareas entre integrantes del equipo (Sommerville, 2016). La coordinación informal no es necesariamente un defecto en equipos pequeños y estables; se convierte en un problema cuando el equipo supera cierto umbral de tamaño y cuando la rotación introduce integrantes que no comparten el contexto acumulado. Vanhanen y Lehtinen (2014) reportan patrones similares en equipos de proyectos *capstone*: la ausencia de prácticas explícitas de seguimiento es una de las fuentes más frecuentes de desviación en proyectos académicos de software, y su impacto se agrava proporcionalmente al tamaño del equipo.

Esta dimensión interactúa directamente con la de gobernanza: la ausencia de una estructura formal de roles y responsabilidades –dimensión anterior– dejó sin un responsable claro la función de coordinación operativa. No era un problema de herramientas –*Discord, Git* y *Drive* estaban disponibles– sino de lineamientos que establecieran cómo y con qué frecuencia usarlas. Esta distinción entre infraestructura disponible y gobernanza que la active es relevante para otros semilleros que podrían asumir que dotar al equipo de herramientas digitales es suficiente para garantizar la coordinación.

4.3 Metodología de desarrollo y definición temprana del *stack*

Las decisiones sobre metodología de desarrollo y *stack* tecnológico no fueron establecidas como punto de partida del proyecto, sino que emergieron de manera progresiva durante la implementación. El modelo incremental con inspiración Scrum planteado en la Sección 2.3 nunca fue operacionalizado de forma completa: no se definieron roles formales, no se gestionó un *backlog* priorizado y los entregables por *sprint* variaban según el criterio de cada

equipo. En paralelo, la ausencia de una decisión temprana sobre el *stack* llevó a que distintos integrantes adoptaran herramientas y enfoques de implementación incompatibles entre sí, generando reprocesos de integración que contribuyeron directamente al atraso acumulado de entre siete y nueve meses. Este patrón fue especialmente pronunciado durante las fases de análisis y diseño, donde los reprocesos en diagramas *UML* y el modelo de datos se repitieron en múltiples iteraciones a lo largo de los cuatro semestres.

4.3.1 Voz del equipo

A continuación, se presentan las afirmaciones que realiza el equipo:

- "No basta con definir la metodología; todo el equipo debe comprender exactamente cómo se usará y qué entregables genera en cada fase."
- "Las entrevistas con los actores representan el punto de partida, pero es fundamental establecer una ruta clara de trabajo para las fases de análisis y diseño."
- "Es indispensable definir las tecnologías que se utilizarán desde las primeras etapas del proyecto."
- "La selección del *stack* tecnológico debe realizarse desde las primeras fases del proyecto."
- "Uno de los aspectos más complejos a mejorar es la fase de diseño, especialmente en la elaboración de modelos *UML* con criterios estandarizados."

La primera y la última afirmación son complementarias y apuntan al mismo problema desde ángulos distintos: no es suficiente nombrar una metodología, ni es suficiente producir diagramas. Lo que faltó fue un entendimiento compartido de qué significa cada artefacto, bajo qué criterios se considera aceptable y quién tiene autoridad para aprobarlo.

4.3.2 Impacto sobre el *SDLC*

En el *SDLC*, la fase de diseño es el momento en que las decisiones arquitectónicas y tecnológicas deben quedar fijadas como base para la implementación (Bass y otros, 2012; Sommerville, 2016). Postergar estas decisiones no elimina el costo de tomarlas; lo transfiere a etapas posteriores donde el costo de cambio es significativamente mayor (Boehm, 1981). En ABI este costo se materializó en rediseños de la base de datos con módulos ya codificados y en reasignaciones de tareas ante incompatibilidades de integración, exactamente el escenario que Boehm documentó empíricamente como el más costoso del ciclo de vida del software.

La interacción con la dimensión de gobernanza es directa: sin una autoridad formal que tomara y comunicara decisiones arquitectónicas tempranas, cada equipo operó con su propia interpretación del *stack* y la metodología. Pero hay también una interacción con la dimensión de coordinación: la ausencia de sincronización sistemática entre equipos retrasó el momento en que las incompatibilidades técnicas se hicieron visibles, reduciendo el margen de corrección disponible. Vale señalar además que la adopción parcial de Scrum sin sus mecanismos de control –*backlog*, definición de terminado, revisión de *sprint*– no es un fenómeno exclusivo del semillero: Boehm y Turner (2004) advierten que la implementación selectiva de prácticas ágiles sin su andamiaje completo puede generar más ambigüedad que la ausencia de metodología formal.

4.4 Documentación y gestión del conocimiento

La documentación del proyecto no operó como un sistema estructurado de gestión del conocimiento sino como un conjunto disperso de artefactos sin criterios comunes de organización ni responsables claros de mantenimiento. Las decisiones técnicas, la arquitectura del sistema y la justificación de implementaciones relevantes vivían principalmente en el conocimiento tácito de los integrantes con mayor permanencia. Este patrón se hizo crítico en dos momentos recurrentes a lo largo de los cuatro semestres: la incorporación de estudiantes nuevos a equipos con trabajo en curso, y la reducción de la participación de integrantes S3-S4 por carga académica. En ambos casos, la reconstrucción del contexto técnico dependió de la disponibilidad y memoria de quienes habían participado previamente, introduciendo demoras y riesgos de pérdida irreversible de conocimiento. La conexión con la dinámica de comunidad de práctica descrita en la Sección 2.1 es directa: en ausencia de mecanismos formales, el conocimiento circulaba por participación periférica legítima (Lave y Wenger, 1991), lo que funciona bien para el aprendizaje individual, pero es frágil como mecanismo de continuidad técnica del proyecto.

4.4.1 Voz del equipo

A continuación, se presentan las afirmaciones que realiza el equipo:

- "Se debe centralizar la información en un solo lugar para facilitar la consulta y mantener versiones organizadas de todos los documentos y diagramas."
- "Es indispensable dejar evidencia detallada de entrevistas, acuerdos y validaciones realizadas durante el proceso."
- "Se debe llevar un registro de errores y correcciones, incluyendo análisis de causa raíz y soluciones implementadas."
- "Todas las actividades, reuniones y decisiones deberían quedar registradas en actas, incluyendo fechas, responsables y validaciones."

La tercera afirmación merece atención particular: los estudiantes no solo identificaron la falta de documentación prospectiva, sino la ausencia de un registro retrospectivo de errores con análisis de causa raíz. Esto sugiere una madurez diagnóstica notable –reconocen que el problema no es solo no documentar lo planeado, sino no aprender sistemáticamente de lo que falló–.

4.4.2 Impacto sobre el *SDLC*

La documentación no es una actividad periférica del *SDLC*: atraviesa todas sus fases y es el mecanismo principal mediante el cual el conocimiento técnico se hace independiente de las personas que lo generaron (Forward y Lethbridge, 2002; Robillard, 1999). Su ausencia en ABI no fue neutral: produjo una dependencia estructural del conocimiento tácito que hizo al proyecto vulnerable a cualquier variación en la composición del equipo. Nonaka y Takeuchi (1995) describen precisamente este riesgo en organizaciones que no han desarrollado mecanismos de conversión del conocimiento tácito en explícito: el conocimiento existe, pero no puede ser transferido ni reutilizado de forma sistemática.

Esta dimensión es receptora de efectos de las tres anteriores. La ausencia de gobernanza significó que nadie tenía mandato formal para exigir documentación. La coordinación informal dejó decisiones sin registro. La definición tardía del *stack* generó cambios arquitectónicos que nunca fueron retroactivamente documentados. El resultado fue una deuda documental acumulada que se hizo especialmente visible en las transiciones entre cohortes; el momento en que la memoria organizacional del semillero debería haberse apoyado precisamente en los artefactos que no existían.

4.5 Trabajo colaborativo y responsabilidad individual

La distribución de carga dentro del equipo no fue uniforme a lo largo del proyecto. Hacia el cierre del desarrollo de la versión *1.0*, un subconjunto de entre seis y ocho estudiantes de niveles S3-S4 absorbió la mayor parte de la carga técnica y organizativa, mientras que la contribución efectiva de otros integrantes era difícil de rastrear en ausencia de mecanismos formales de seguimiento individual. Este patrón no es atribuible exclusivamente a diferencias de compromiso: la estructura del semillero, con equipos de más de seis integrantes y sin asignación explícita de responsabilidades por entregable, creaba condiciones objetivas para la concentración de carga. La percepción de inequidad resultante fue reportada de forma recurrente durante las sustentaciones orales del último semestre y es consistente con la disminución de motivación observable hacia finales de 2025.

4.5.1 Voz del equipo

A continuación, se presentan las afirmaciones que realiza el equipo:

- "Sería pertinente implementar una evaluación de los aportes individuales en cada semestre del semillero, de la cual dependa el aval correspondiente."
- "Para fortalecer el compromiso, sería necesario ser más estrictos con la asistencia a las reuniones de los sábados."
- "Cuando los equipos superan los seis integrantes, se dificulta el seguimiento del aporte individual y la gestión de responsabilidades."

Las tres afirmaciones apuntan a soluciones distintas para el mismo problema: una propone consecuencias formales vinculadas al aval académico, otra propone control de asistencia, y la tercera identifica el tamaño del equipo como variable crítica. Que los estudiantes lleguen a estas tres conclusiones de forma independiente sugiere

que la inequidad percibida era suficientemente extendida como para generar múltiples hipótesis diagnósticas dentro del mismo equipo.

4.5.2 Impacto sobre el *SDLC*

A lo largo del *SDLC*, la trazabilidad de contribuciones individuales cumple una función técnica concreta: permite identificar al responsable de un componente cuando aparece un defecto, facilita la revisión de código entre pares y hace posible la transferencia ordenada de responsabilidades (Sommerville, 2016). En ABI, la asignación difusa de responsabilidades no solo generó inequidad percibida sino consecuencias técnicas directas: componentes sin responsable claro fueron los que con mayor frecuencia requirieron intervención tardía por parte de los estudiantes de mayor nivel, amplificando la concentración de carga que el patrón ya tendía a producir.

Brooks (1995) advierte que agregar integrantes a un proyecto tardío lo retrasa aún más, precisamente porque el costo de integración y transferencia de contexto supera el beneficio de la capacidad adicional. En ABI el mecanismo fue análogo pero inverso: no se agregaron integrantes tarde, sino que los integrantes existentes operaban con un contexto compartido insuficiente, forzando a los estudiantes de mayor permanencia a asumir funciones de rescate técnico que no estaban planificadas. Iacob y Faily (2019) documentan un fenómeno similar en proyectos grupales de pregrado: la concentración de carga en un subgrupo reducido es una consecuencia predecible cuando los mecanismos de seguimiento individual son débiles y el tamaño del equipo supera el umbral de coordinación informal.

Esta dimensión cierra el ciclo de interacciones entre las cinco: la gobernanza no exigió responsabilidades individuales formales, la coordinación no las hizo visibles en tiempo real, la metodología no las ancló a entregables concretos, y la documentación no las registró. La concentración de carga en los estudiantes S3-S4 no fue un accidente sino el resultado predecible de cinco dimensiones que se reforzaron mutuamente.

4.6 Trazabilidad técnica: de las dimensiones organizacionales a los efectos sobre el sistema

Las cinco dimensiones analizadas en las secciones anteriores no son únicamente categorías organizacionales: cada una se tradujo en efectos técnicos concretos y medibles sobre el sistema ABI. Esta sección sistematiza esa trazabilidad a partir de la evidencia documental disponible –informes de módulo, registros de *pull requests*, actas de *QA* y documentación técnica oficial– con el propósito de responder una pregunta que el análisis organizacional solo plantea implícitamente: ¿qué le costó al producto de software cada una de estas desviaciones?

4.6.1 Métricas de proceso

El desarrollo de ABI *v1.0* generó la siguiente evidencia técnica cuantificable: a) defectos registrados: el equipo de *QA* consolidó 13 registros formales de bugs reales durante cinco semanas de revisión sistemática, distribuidos en seis categorías: 4 de interfaz y codificación; 3 de navegación y rendimiento; 3 de validación de formularios; 2 de despliegue y configuración; y 1 de gestión de datos y *seeders*; b) proceso de versionado: el módulo de historial de proyectos fue desarrollado en seis semanas con entregas incrementales semanales, revisión por *pull requests*, actas grupales y criterios de aceptación definidos a priori para los requisitos funcionales RF04 y RF05; c) cobertura de pruebas automatizadas: la suite del módulo de versiones ejecutó cuatro pruebas de tipo *feature/integration* –creación de versión, consulta de historial, inmutabilidad y flujo de corrección/reenvío– con resultado exitoso al cierre del módulo; d) pruebas funcionales manuales: se validaron ocho casos de uso del módulo de versionado por roles, incluyendo escenarios de estado vacío y acceso diferenciado por perfil.

La Tabla 3 sistematiza la correspondencia entre cada dimensión organizacional, la desviación observada y su efecto técnico concreto sobre el sistema.

Tabla 3. Trazabilidad entre dimensiones organizacionales y efectos técnicos sobre ABI *v1.0*

Dimensión	Desviación observada	Efecto técnico concreto
Gobernanza y dirección académica	Ausencia de criterios de aceptación formales por fase	RF04 y RF05 no implementados en *v1.0* inicial; rediseño requerido en *v1.5*
Gestión de proyecto y coordinación	Sincronización únicamente semanal entre equipos	Incompatibilidades de integración detectadas tardíamente; ajustes no planificados en controladores y vistas
Metodología de desarrollo y definición temprana del *stack*	Ausencia de decisión temprana sobre *stack* tecnológico	Rediseño de tabla *versions* con módulos ya codificados; migración forzada como corrección tardía
Documentación y gestión del conocimiento	Conocimiento técnico en memoria de integrantes S3-S4	13 bugs de codificación (*UTF*-8/tildes) atribuibles a ausencia de estándares documentados de configuración
Trabajo colaborativo y responsabilidad individual	Asignación difusa de módulos	Módulos sin responsable claro requirieron intervención tardía de S3-S4; carga concentrada en 6-8 de 35 estudiantes

4.6.2 Deuda técnica acumulada

En términos del marco de Cunningham (1992) y Kruchten *et at* (2012), el proceso de desarrollo de ABI generó deuda técnica en dos dimensiones identificables: a) deuda de diseño, materializada en el rediseño de la tabla *versions* con módulos ya implementados y en las incompatibilidades de integración entre componentes desarrollados con *stacks* parcialmente incompatibles; y b) deuda de pruebas, expresada en la ausencia de una suite de pruebas sistemática en los módulos desarrollados durante los primeros dos semestres, subsanada parcialmente en la versión *1.5*. Ambas formas de deuda son consistentes con los patrones documentados por Zazworka y otros (2011) en proyectos académicos de software: la presión de entrega y la rotación de equipos son los predictores más fuertes de acumulación de deuda técnica, independientemente del nivel de experiencia de los desarrolladores.

La trazabilidad presentada en la Tabla 3 no pretende establecer causalidad directa entre cada desviación organizacional y cada efecto técnico –la evidencia disponible es retrospectiva y de naturaleza documental– sino identificar correspondencias sistemáticas que sugieren mecanismos de amplificación entre las dimensiones organizacionales y los costos técnicos del proyecto. Esta interpretación es consistente con el diseño metodológico del estudio y con los límites de validez interna discutidos en la Sección 3.3.

5 Recomendaciones para semilleros de investigación

Las cinco dimensiones analizadas en la sección anterior no solo describen lo que ocurrió en ABI: configuran un mapa de puntos de intervención organizacional para proyectos de software en entornos formativos. Las recomendaciones que siguen no son prescripciones genéricas de gestión de proyectos trasladadas al aula; son respuestas específicas a patrones observados en un semillero real, formuladas en términos operativos adaptados a sus condiciones estructurales. Cada subsección elimina deliberadamente la recapitulación del problema –ya analizado en la sección anterior– y va directamente a la forma concreta que debe tomar la solución.

5.1 Gobernanza y dirección académica

La gobernanza de un proyecto en un semillero no puede depender exclusivamente de la figura del director. Requiere una estructura distribuida con responsabilidades claras en cada nivel jerárquico, S1-S4, y mecanismos institucionales que la sostengan más allá de la permanencia de personas específicas. Se destacan los siguientes:

- Formalización de entregables por fase: Al inicio de cada periodo académico, cada equipo de trabajo debe producir un documento de alcance que especifique sus entregables, criterios de aceptación y responsables. Este documento debe ser revisado por el equipo de *QA* y avalado formalmente por la dirección del semillero y el comité asesor antes de iniciar el desarrollo (Sommerville, 2016).
- Validación de fase como condición de avance: Ningún equipo debe iniciar la fase siguiente sin que sus entregables de la fase actual hayan sido validados por Q*A* y aprobados por la dirección. Esta práctica introduce puntos de control institucionales que reemplazan la validación informal (Schwaber y Sutherland, 2020).

- Comité asesor activo en hitos críticos: El comité asesor no debe limitarse a avalar el inicio del proyecto; debe participar en al menos dos revisiones formales durante el desarrollo –al cierre del análisis y al cierre del diseño– para garantizar la alineación entre las expectativas institucionales y el producto en construcción.
- Clarificación de marcos conceptuales y tecnológicos adoptados: Todo marco de trabajo –metodológico, arquitectónico o tecnológico– debe quedar definido por escrito al inicio del proyecto, con una descripción operativa de cómo se usará y qué artefactos genera en cada fase. No basta con nombrar el marco; el equipo completo debe comprender su aplicación concreta.

5.2 Arquitectura operativa del proyecto

La coordinación informal funciona en equipos pequeños y estables. En un semillero con 35 estudiantes, rotación entre cohortes y disponibilidad fragmentada, la coordinación debe ser diseñada explícitamente, no asumida (Brooks, 1995). Al respecto, se destaca:

- Reuniones de equipo pequeño como unidad base de sincronización: Los equipos de trabajo de 3 a 5 personas deben reunirse con una frecuencia definida, con agenda predefinida y minuta estructurada que registre decisiones, compromisos y responsables. Este formato reemplaza la sincronización única del sábado por puntos de control más frecuentes y manejables (Schwaber y Sutherland, 2020).
- Reunión mensual de presentación general: Una vez al mes, todos los equipos presentan su avance al semillero completo. Este espacio cumple dos funciones: detectar incompatibilidades entre componentes antes de que se acumulen, y mantener una visión compartida del estado global del proyecto.
- Disponibilidad explícita como condición de participación: Al vincularse al proyecto, cada estudiante debe comprometerse formalmente con una disponibilidad semanal mínima definida. Esta condición debe ser conocida antes del ingreso, no descubierta durante el desarrollo.
- Equipos de 3 a 5 personas como límite operativo: Equipos más grandes dificultan la trazabilidad del aporte individual y la coordinación efectiva (Brooks, 1995). Cuando el proyecto requiera más integrantes, la solución es multiplicar equipos pequeños con interfaces bien definidas, no agrandar equipos existentes.

5.3 Coherencia metodológica y decisiones técnicas tempranas

Adoptar una metodología ágil en un semillero es una decisión pedagógicamente pertinente (Mahnic, 2012; Paasivaara y otros, 2014), pero su valor depende de que sea operacionalizada completamente, no seleccionada parcialmente. Un Scrum sin *backlog* priorizado, sin definición de terminado y sin revisión de *sprint* no es Scrum: es ambigüedad con vocabulario ágil (Boehm y Turner, 2004). Se resalta lo siguiente:

- Guía metodológica operativa vinculada a entregables: Antes del primer *sprint*, el equipo debe producir un documento que describa cómo se aplicará la metodología elegida: qué ceremonias se realizarán, con qué frecuencia, qué artefactos generará cada fase y bajo qué criterios se considerará terminado un entregable (Schwaber y Sutherland, 2020).
- Definición del *stack* tecnológico como entregable del análisis: La selección de tecnologías no debe postergarse hasta la implementación. Debe formalizarse como un entregable explícito de la fase de análisis, con justificación documentada de las decisiones adoptadas (Bass y otros, 2012). En proyectos iterativos, este entregable corresponde al inicio de cada ciclo.
- Sesión de alineación metodológica al inicio de cada periodo: Con la rotación entre cohortes, no puede asumirse que los integrantes nuevos conocen la metodología en uso. Una sesión de inducción al inicio de cada semestre, liderada por estudiantes S3–S4, garantiza un punto de partida compartido.
- Estándares mínimos para modelado y diseño: El equipo debe acordar y documentar criterios mínimos de calidad para los artefactos de diseño –diagramas *UML*, modelos de datos, especificaciones de interfaz– antes de iniciar su producción (Sommerville, 2016). Sin estos criterios, la validación de diseños se vuelve subjetiva e inconsistente entre equipos.

5.4 Sistema de trazabilidad y memoria organizacional

En un semillero, el conocimiento técnico del proyecto no puede depender de la memoria de personas específicas. Con rotación entre cohortes, cada transición es una amenaza a la continuidad si no existe una memoria organizacional explícita que la sostenga (Nonaka y Takeuchi, 1995). A continuación, se destaca:

- Plantillas de acta como estándar mínimo de registro: Todas las reuniones de equipo y de semillero deben registrarse en un formato estandarizado que incluya fecha, participantes, decisiones tomadas, responsables y criterios de validación asociados. La existencia de una plantilla reduce la fricción de documentar y garantiza consistencia entre equipos.
- Registro de decisiones arquitectónicas: Cada decisión técnica significativa –elección de tecnología, cambio de arquitectura, modificación del modelo de datos– debe quedar registrada con su justificación, las alternativas consideradas y el responsable de la decisión (Tyree y Akerman, 2005). Este registro es el activo documental más valioso para los integrantes que se incorporan al proyecto en semestres posteriores.
- Bitácora de errores con análisis de causa raíz: Cada error significativo –definido como aquel que generó un atraso de una semana o más– debe registrarse con su descripción, causa identificada y solución implementada. Esta práctica convierte los errores en conocimiento organizacional reutilizable (Forward y Lethbridge, 2002).
- Consolidación documental al cierre de cada fase: Antes de iniciar la fase siguiente, un integrante designado del equipo de *QA* debe verificar que toda la documentación de la fase completada esté centralizada, actualizada y accesible en el repositorio compartido. Este cierre formal evita la acumulación de deuda documental entre fases.

5.5 Mecanismos de responsabilidad individual y equilibrio de carga

La responsabilidad individual en un entorno colaborativo no es opuesta al trabajo en equipo: es su condición de posibilidad. Sin trazabilidad de contribuciones individuales, la carga tiende a concentrarse en quienes tienen mayor permanencia y compromiso, generando inequidad estructural independientemente de la actitud de los integrantes (Iacob y Faily, 2019).

- Asignación de módulos y roles desde la fase de análisis: Al conformarse cada equipo, deben quedar definidos explícitamente los módulos o aspectos del sistema bajo responsabilidad de ese equipo, y los roles de cada integrante dentro de él. Esta asignación debe quedar documentada y ser conocida por toda la dirección del semillero.
- Reporte individual semanal y consolidado mensual: Cada estudiante debe reportar semanalmente sus actividades individuales y su aporte al equipo. Al cierre de cada mes, el líder del equipo consolida estos reportes y los presenta a la dirección. Este mecanismo hace visible la distribución real de carga antes de que la concentración se vuelva crítica.
- Evaluación individual vinculada al aval académico: La continuidad del aval semestral debe depender, al menos parcialmente, de la evidencia de contribución individual documentada. Esta vinculación introduce una consecuencia formal que complementa la motivación intrínseca y reduce el riesgo de participación nominal dentro del equipo.
- Límite operativo de 3 a 5 integrantes por equipo: Por encima de este umbral, el seguimiento del aporte individual se vuelve inmanejable con los mecanismos informales disponibles en un semillero. El límite no es arbitrario: refleja el tamaño máximo en el que un líder de equipo puede mantener visibilidad real sobre las contribuciones de cada integrante (Brooks, 1995).

5.6 Integración sistémica de las dimensiones propuestas

Las cinco dimensiones no son independientes: su valor real emerge de su adopción conjunta, pues una dimensión débil amplifica las deficiencias de las demás. La Figura 3 muestra las relaciones de dependencia e interacción identificadas, donde los nodos representan las cinco dimensiones; las aristas dirigidas indican relaciones de amplificación documentadas en la evidencia del proyecto (dirección: causa → efecto observado).



Figura 3. Mapa de dependencias entre las cinco dimensiones organizacionales

La gobernanza ocupa el nodo de mayor centralidad. La coordinación activa la infraestructura disponible. La coherencia metodológica ancla las responsabilidades a entregables concretos. La documentación preserva el resultado de todas las demás entre cohortes. Y la responsabilidad individual es la señal de alarma más sensible: cuando la carga se concentra, es porque al menos una de las otras cuatro dimensiones ha fallado. Juntas configuran una estructura organizacional mínima adaptada a las condiciones reales de los semilleros, con el nivel de formalidad suficiente para sostener un proyecto de software a lo largo de múltiples cohortes sin perder su carácter formativo.

6 Conclusiones y trabajo futuro

La contribución principal de este trabajo no es el diagnóstico de un caso particular sino el marco analítico que emerge de él. Las cinco dimensiones identificadas configuran una estructura organizacional mínima adaptada a las condiciones reales de los semilleros universitarios latinoamericanos: rotación entre cohortes, disponibilidad fragmentada y heterogeneidad técnica. A diferencia de marcos genéricos de gestión de proyectos, este marco parte precisamente de las limitaciones estructurales de los entornos formativos para proponer prácticas con el nivel de formalidad suficiente para sostener un proyecto de software sin sacrificar el carácter pedagógico del semillero.

Este estudio tiene límites que deben tenerse presentes al interpretar sus resultados. Es un caso único en un contexto institucional específico –semillero Máster Digital, UDI, Bucaramanga, Colombia–, lo que acota su alcance a la transferibilidad conceptual y no a la generalización estadística (Yin, 2014). Las correspondencias documentadas en la Tabla 3 entre dimensiones organizacionales y efectos técnicos son consistentes con la evidencia disponible, pero no constituyen relaciones causales verificadas experimentalmente: son patrones interpretativos derivados de un análisis retrospectivo con los mecanismos de control de sesgo descritos en la Sección 3.3. En particular, la cuantificación de defectos –13 registros formales de *QA*– y de pruebas automatizadas –4 casos en el módulo de versiones– refleja la evidencia documental disponible del proyecto, no una medición sistemática de la calidad técnica del sistema en su totalidad.

El riesgo más significativo de sesgo es el de autoevaluación, gestionado mediante revisión cruzada entre autores y socialización de resultados con la DI como actor externo; sin embargo, ninguno de estos mecanismos constituye una validación externa independiente en sentido estricto, por lo que el sesgo residual debe considerarse una amenaza real. Adicionalmente, la sistematización retrospectiva puede estar influida por el conocimiento del desenlace del proyecto –sesgo de retrospección (Yin, 2014)–, lo que representa una limitación inherente al diseño metodológico adoptado. Las recomendaciones propuestas en la Sección 5 deben interpretarse como hipótesis de intervención organizacional cuya validez empírica depende de su implementación y evaluación sistemática en otros contextos formativos, lo que constituye el objeto del trabajo de seguimiento actualmente en curso.

Como trabajo futuro se identifican tres líneas prioritarias: la evaluación de las recomendaciones en la versión *1.5* de ABI –objeto de un manuscrito de seguimiento–; la validación del marco en otros semilleros latinoamericanos; y la exploración de métricas cuantitativas que permitan operacionalizar las dimensiones propuestas.

Declaración de conflicto de intereses

Los autores declaran no tener conflicto de intereses con respecto a la investigación, autoría o publicación de este artículo.

Agradecimientos

Los autores quieren agradecer a todos los estudiantes integrantes del semillero Máster Digital, del programa de Ingeniería de Sistemas de la Universidad de Investigación y Desarrollo. Su participación, compromiso, errores y aciertos hicieron posible obtener las lecciones aprendidas presentadas en este manuscrito.

ORCID ID

Miguel Ángel Becerra Solano https://orcid.org/0009-0001-3669-8709
Jheison Alexander De la Rosa Pineda https://orcid.org/0009-0002-2968-6053
Angie Alejandra Hernández León https://orcid.org/0009-0001-8498-6094
Michael Fernney Pérez Ochoa https://orcid.org/0009-0002-4542-9110
Crisóstomo Barajas-Solano https://orcid.org/0000-0001-9781-3603

Referencias

Bass, L., Clements, P., y Kazman, R. (2012). *Software Architecture in Practice* (3.ª ed.). Addison-Wesley.

Boehm, B. (1981). *Software Engineering Economics.* Prentice-Hall. https://staff.emu.edu.tr/alexanderchefranov/Documents/CMPE412/Boehm1981%20COCOMO.pdf

Boehm, B., y Turner, R. (2004). *Balancing Agility and Discipline: A Guide for the Perplexed.* Addison-Wesley.

Brew, A. (2006). *Research and Teaching: Beyond the Divide.* (M. E. UK, Ed.) Palgrave Macmillan.

Brooks, F. (1995). *The Mythical Man-Month: Essays on Software Engineering* (2.ª ed.). Addison-Wesley.

Cunningham, W. (1992). The WyCash portfolio management system. *ACM SIGPLAN OOPS Messenger, 4*(2), 29-30. https://doi.org/10.1145/157710.157715

Fenton, N., y Bieman, J. (2014). *Software Metrics: A Rigorous and Practical Approach* (3.ª ed.). CRC Press.

Flores, E., Mendoza, R., y Loaiza, A. (2019). Semilleros de investigación: una práctica para el desarrollo científico de las naciones. *Negotium, 15*(44), 20-34. https://doi.org/http://doi.org/10.5281/zenodo.4781798

Forward, A., y Lethbridge, T. (2002). The relevance of software documentation, tools and technologies: a survey. *Proceedings of the 2002 ACM symposium on Document engineering*, *66.* https://doi.org/10.1016/S0164-1212(02)00087-9

Healey, M., y Jenkins, A. (2009). *Developing Undergraduate Research and Inquiry.* Higher Education Academy.

Iacob, C., y Faily, S. (2019). Exploring the Gap between the Student Expectations and the Reality of Teamwork in Undergraduate Software Engineering Group Projects. *The Journal of Systems and Software, 157*, 110393. https://doi.org/10.1016/j.jss.2019.110393

Kruchten, P., Nord, R., y Ozkaya, I. (2012). Technical debt: From metaphor to theory and practice. *IEEE Software, 29*(6), 18-21. https://doi.org/10.1109/MS.2012.167

Lave, J., y Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation.* Cambridge University Press. https://doi.org/10.1017/CBO9780511815355

Lopatto, D. (2007). Undergraduate Research Experiences Support Science Career Decisions and Active Learning. *CBE—Life Sciences Education, 6*(4), 297-306. https://doi.org/10.1187/cbe.07-06-0038

Mahnic, V. (2012). A Capstone Course on Agile Software Development Using Scrum. *IEEE Transactions on Education, 55*(1), 99-106. https://doi.org/10.1109/TE.2011.2142311

Nonaka, I., y Takeuchi, H. (1995). *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation.* Oxford University Press.

Paasivaara, M., Heikkilä, V., Lassenius, C., y Toivola, T. (2014). Teaching Students Scrum Using LEGO Blocks. *Companion Proceedings of the 36th International Conference on Software Engineering*, (págs. 382-391). New York. https://doi.org/10.1145/2591062.2591169

Quintero Corzo, J., Munévar Molina, R. A., y Munévar Quintero, F. I. (2008). Semilleros de investigación: una estrategia para la formación de investigadores. *Educación y Educadores, 11*(1), 31-42. https://educacionyeducadores.unisabana.edu.co/index.php/eye/article/view/716

Robillard, P. (1999). The Role of Knowledge in Software Development. *Communications of the ACM, 42*(1), 87-92. https://doi.org/10.1145/291469.291479

Schwaber, K., y Sutherland, J. (2020). *Scrum.org & Scrum Inc*. The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. https://www.scrumguides.org/scrum-guide.html

Sommerville, I. (2016). *Software Engineering* (10.ª ed.). Pearson.

Stake, R. (1995). *The Art of Case Study Research.* SAGE Publications.

Tyree, J., y Akerman, A. (2005). Architecture Decisions: Demystifying Architecture. *IEEE Software, 22*(2), 19-27. https://doi.org/10.1109/MS.2005.27

Vanhanen, J., y Lehtinen, T. (2014). Software Engineering Problems Encountered by Capstone Project Teams. *International Journal of Engineering Education, 30*(6), 1461-1475. https://api.semanticscholar.org/CorpusID:114340914

Wenger, E. (1998). *Communities of Practice: Learning, Meaning, and Identity.* Cambridge University Press. https://doi.org/10.1017/CBO9780511803932

Yin, R. (2014). *Case Study Research: Design and Methods* (5.ª ed.). SAGE Publications. https://doi.org/10.3138/CJPE.30.1.108

Zazworka, N., Shaw, M., Shull, F., y Seaman, C. (2011). Investigating the impact of design debt on software quality. *Proceedings of the 2nd Workshop on Managing Technical Debt* (págs. 17-23). New York: ACM. https://doi.org/10.1145/1985362.1985366