

PRIMERA EDICIÓN



INGENIERÍA DE SOFTWARE EN LA EDUCACIÓN UNIVERSITARIA:

Experiencias didácticas para una formación integral

AUTORÍA

Mariuxi Vinuela-Morales
Delia Isabel Carrión León
Jessica Janina Cabezas Quinto
Denis Darío Mendoza Cabrera

Ingeniería de software en la educación universitaria: experiencias didácticas para una formación integral

Autores

Mariuxi G. Vinueza-Morales
Delia Isabel Carrión León
Jessica Janina Cabezas Quinto
Denis Darío Mendoza Cabrera

© Ediciones RISEI, 2025.

Todos los derechos reservados.

Este libro se distribuye bajo la licencia Creative Commons Atribución CC BY 4.0 Internacional.

Las opiniones expresadas en esta obra son responsabilidad exclusiva de sus autores y no reflejan necesariamente la posición de la editorial.

Editorial: Ediciones RISEI.

Colección Sistematización de Experiencias Educativas.

Título del libro: Ingeniería de software en la educación universitaria: experiencias didácticas para una formación integral.

Autoría: Mariuxi G. Vinueza-Morales / Delia Isabel Carrión León / Jessica Janina Cabezas Quinto / Denis Darío Mendoza Cabrera.

Edición: Primera edición.

Año: 2025.

ISBN: 978-9942-596-28-4.

DOI: <https://doi.org/10.63624/risei.book-978-9942-596-28-4>

Coordinación editorial: Jorge Maza-Córdova y Tomás Fontaines-Ruiz.

Diagramación y diseño: Unidad de Diseño.

Revisión por pares: Sistema doble ciego de revisión externa.

Machala — Ecuador, diciembre de 2025.

Este libro fue diagramado en L^AT_EX.

Disponible en: <https://editorial.risei.org/>

Contacto: info@risei.org

Prólogo

La educación universitaria atraviesa, en este inicio del siglo XXI, un proceso de transformación profunda que interpela no sólo las prácticas docentes y los modelos de enseñanza, sino también la manera en que las instituciones comprenden su misión formativa, científica y social. En ese contexto, la **Red Institucional de Sistematización e Innovación Educativa (RISEI)** surge como una respuesta concreta, reflexiva y articulada a los desafíos contemporáneos que enfrentan las universidades al buscar integrar la investigación, la docencia y la innovación en un mismo horizonte de sentido.

Este primer volumen, **RISEI I**, reúne el trabajo de un conjunto de docentes - investigadores que, desde diferentes disciplinas, experiencias y sedes de la **Universidad Católica de Cuyo**, han asumido el compromiso de revisar críticamente su práctica y de reconstruir, desde la sistematización, los procesos pedagógicos que dieron lugar a innovaciones significativas en la enseñanza universitaria. El libro no sólo documenta experiencias, sino que las analiza, las interpreta y las proyecta, configurando así un corpus valioso para comprender las nuevas dinámicas del aprendizaje, la enseñanza y la investigación en la educación superior contemporánea.

La génesis del programa RISEI

El Programa RISEI nació como una iniciativa impulsada por la **Secretaría de Investigación y Vinculación Tecnológica**, en articulación con las unidades académicas, con el propósito de fortalecer la cultura investigativa y el desarrollo de la innovación pedagógica en la universidad. Desde su creación, el programa se propuso consolidar un espacio común donde la reflexión docente se traduzca en conocimiento institucional, sistemático y transferible.

RISEI se sustenta en una premisa central: la mejora de la enseñanza universitaria no puede depender únicamente de la adopción de nuevas tecnologías o de la actualización curricular, sino que requiere de un **proceso reflexivo y colectivo** en el que los docentes se reconozcan como **productores de conocimiento** sobre su propia práctica.

A través de esta red se promueve la integración de tres dimensiones esenciales:

1. **La innovación pedagógica**, entendida como la capacidad de crear, adaptar y experimentar estrategias didácticas que respondan a las transformaciones del contexto.
2. **La investigación educativa**, orientada a producir evidencia y comprensión teórica sobre los procesos de enseñanza y aprendizaje.

3. **La sistematización de experiencias**, como metodología que permite reconstruir el recorrido vivido, otorgándole significado, coherencia y posibilidad de transferencia.

Este marco dio lugar a una serie de proyectos interdisciplinarios que, en su conjunto, conforman los capítulos de este primer volumen.

El sentido institucional del libro

El presente libro constituye un **producto colectivo** del proceso de investigación-acción desarrollado dentro del marco del programa RISEI. Su sentido no es sólo académico, sino también institucional: busca consolidar una cultura de la reflexión pedagógica documentada y evaluable, que permita a la universidad **aprender de su propia experiencia**.

En tiempos donde las universidades enfrentan desafíos vinculados a la digitalización acelerada, la evaluación por competencias, la diversificación de perfiles estudiantiles y la necesidad de generar vínculos más estrechos con la comunidad, RISEI representa un esfuerzo por **recuperar la dimensión humanista de la innovación**. No se trata únicamente de incorporar recursos tecnológicos, sino de reconfigurar el modo en que se enseña, se aprende y se investiga, reconociendo al docente como un **actor transformador y reflexivo**, y al estudiante como un **sujeto activo en la construcción del conocimiento**.

Este volumen es también un testimonio del compromiso institucional por **profesionalizar la docencia universitaria**, entendiendo que la enseñanza de calidad requiere formación permanente, trabajo interdisciplinario y apertura al cambio. Cada capítulo que lo integra es el resultado de un proceso de indagación situado, donde las preguntas emergen de la práctica y vuelven a ella como propuestas de mejora.

La estructura del volumen

El libro se organiza en cuatro capítulos que, aunque distintos en su temática y enfoque, comparten una lógica común: la búsqueda de estrategias efectivas para mejorar el aprendizaje a través de la reflexión crítica sobre la práctica docente.

El **Capítulo 1**, titulado “*Del problema a la formulación: pensar la enseñanza como objeto de investigación*”, introduce el marco conceptual del programa y las metodologías de sistematización aplicadas. Expone los fundamentos teóricos que orientaron la construcción de las experiencias, destacando la necesidad de una docencia basada en la evidencia y la articulación entre teoría y práctica.

El **Capítulo 2**, “*Del problema a la solución: estrategias para enseñar análisis y diseño de software en primer nivel*”, constituye un ejemplo concreto de cómo la investigación educativa permite rediseñar las estrategias pedagógicas en áreas de alta complejidad técnica. El capítulo combina innovación didáctica con rigor metodológico, mostrando cómo el aprendizaje activo, el trabajo colaborativo y la evaluación formativa pueden mejorar la comprensión de conceptos abstractos en la formación tecnológica.

El **Capítulo 3**, “*De la práctica a la teoría: reflexiones sobre la enseñanza universitaria desde la experiencia*”, amplía el enfoque al analizar los procesos de mediación, acompañamiento y construcción colectiva de conocimiento que surgen en la docencia cotidiana. A través de la voz de los docentes participantes, se visibilizan los desafíos de enseñar en contextos híbridos y los modos de generar comunidades de aprendizaje sostenibles.

Finalmente, el **Capítulo 4**, “*De la innovación a la institucionalización: hacia una cultura universitaria de la investigación educativa*”, plantea las conclusiones generales del proceso, las proyecciones del programa RISEI y los lineamientos para su consolidación como política institucional. Este cierre invita a repensar el lugar de la innovación no como un hecho aislado o episódico, sino como parte del ADN institucional que orienta el desarrollo académico, científico y ético de la universidad.

Una mirada sobre la docencia universitaria en transformación

El libro es testimonio de un cambio de paradigma en la enseñanza universitaria. Las experiencias aquí sistematizadas demuestran que la innovación educativa no depende exclusivamente de grandes reformas curriculares ni de costosas infraestructuras tecnológicas, sino de la capacidad de los docentes para **mirar críticamente su propia práctica**, reconocer los problemas que emergen en el aula y construir soluciones contextualizadas.

Esa mirada investigativa de la docencia, promovida por RISEI, contribuye a transformar el rol del profesor universitario: de transmisor de conocimientos a **facilitador del aprendizaje, diseñador de experiencias y productor de saber pedagógico**. Así, la investigación se convierte en una herramienta para el cambio, y la sistematización, en un puente entre la práctica cotidiana y la producción científica.

La universidad, en este sentido, asume su responsabilidad social al generar conocimiento sobre la enseñanza, contribuyendo al mejoramiento continuo de sus procesos formativos y a la formación integral de los estudiantes. La investigación educativa adquiere entonces una doble función: **explicar y transformar** la realidad universitaria.

RISEI como política de innovación y formación docente

El programa RISEI se inserta en un marco más amplio de desarrollo institucional, vinculado al **Plan Estratégico de Investigación (PEI)** y a la **Ordenanza General de Investigación**, que promueven la articulación entre la docencia, la investigación y la vinculación tecnológica.

La creación de esta red ha permitido que los docentes universitarios encuentren un espacio de formación, acompañamiento y producción académica donde compartir experiencias, problematizar sus prácticas y consolidar proyectos de mejora.

El valor del programa radica en su capacidad para **transformar el conocimiento tácito en conocimiento explícito**, contribuyendo al acervo institucional y fortaleciendo los procesos de acreditación, evaluación y mejora continua. RISEI no sólo documenta experiencias, sino que construye evidencias de impacto pedagógico y propone criterios de replicabilidad que pueden ser transferidos a otros contextos de la universidad o del sistema de educación superior.

Además, la articulación de RISEI con las líneas de investigación institucional y con los programas de desarrollo docente consolida una **comunidad académica reflexiva**, capaz de sostener en el tiempo procesos de innovación significativos y de generar una identidad compartida en torno al valor de la investigación educativa.

Proyección y sostenibilidad

El desafío futuro del programa, y por extensión de este libro, consiste en consolidar un **modelo institucional sostenible** de innovación pedagógica que trascienda las experiencias individuales. La meta no es únicamente publicar resultados, sino asegurar que las prácticas exitosas se integren en la planificación académica, la formación docente y la cultura organizacional de la universidad.

De este modo, RISEI se proyecta como una política universitaria de largo alcance, orientada a fortalecer la calidad educativa, fomentar la investigación aplicada a la docencia y promover la formación de docentes-investigadores.

Cada experiencia sistematizada, cada análisis de resultados, cada reflexión incluida en este volumen es una evidencia concreta de ese compromiso colectivo. Las innovaciones que aquí se presentan no son meros ejercicios académicos, sino **acciones transformadoras** que impactan en el aprendizaje real de los estudiantes, en la gestión curricular y en la construcción de una universidad más reflexiva, crítica y humana.

Una invitación a continuar el camino

Este libro no busca clausurar una etapa, sino abrir un camino. Los capítulos que lo componen son, al mismo tiempo, punto de llegada y punto de partida: llegada de un proceso institucional que logró articular docencia, investigación y reflexión pedagógica; y punto de partida para nuevas líneas de trabajo que amplíen el horizonte de la innovación universitaria.

La invitación es a que cada lector —sea docente, investigador o directivo universitario— encuentre en estas páginas una **fuente de inspiración y de guía metodológica** para repensar su práctica, documentar sus avances y compartirlos con la comunidad académica.

La sistematización de experiencias se presenta, así, como un camino posible hacia la construcción de un conocimiento pedagógico propio, situado y significativo.

El programa RISEI, y este primer volumen en particular, constituyen una muestra de que la innovación educativa no es un acto individual, sino un proceso colectivo que se alimenta de la colaboración, la reflexión y la convicción de que **enseñar es también investigar**.

En esa convergencia entre práctica y teoría, entre aula y laboratorio, entre experiencia y evidencia, se configura la verdadera identidad de una universidad comprometida con su tiempo.

Índice general

Prólogo	i
1. Fundamentos de ingeniería de software: de la práctica docente reflexiva a la formación por competencias	1
1.1. Fundamentos de Ingeniería de Software: del aula a la práctica profesional reflexiva	4
1.1.1. Apertura contextual	4
1.1.2. Problematización	5
1.1.3. Propósito de la sistematización	5
1.1.4. Criterios de valor	6
1.1.5. Delimitación del objeto de estudio	7
1.2. Fundamentación conceptual y operativa de la experiencia	8
1.2.1. Transición hacia la fundamentación	8
1.2.2. Identificación de conceptos estructurantes	9
1.2.3. Formulación de dimensiones	10
1.2.4. Construcción de indicadores	10
1.2.5. Fuentes y métodos de verificación	13
1.2.6. Justificación teórica del conjunto	13
1.2.7. Recapitulación y proyección	14
1.3. Vínculo con el currículo y el perfil de la Carrera	15
1.3.1. Transición hacia la fundamentación	15
1.3.2. Identificación de competencias del perfil	15
1.3.3. Resultados de aprendizaje vinculados	16
1.3.4. Actividades y evidencias de aprendizaje	16
1.3.5. Reflexión sobre la alineación curricular	18
1.3.6. Síntesis final del módulo	19
1.4. Ecosistema estratégico (estrategias y relaciones)	20
1.4.1. Puente de Transición hacia el Análisis Estratégico	20
1.4.2. Estrategias de Soporte	21
1.4.3. Estrategias de Contingencia	22
1.4.4. Integración Estratégica y Proyección hacia la Evaluación	23
1.5. Evaluación, indicadores, instrumentos, análisis	24
1.5.1. Puente de Evaluación: Instrumentos, Indicadores y Criterios de Validez	24
1.5.2. Rúbrica analítica de desempeño	25
1.5.3. Cuestionario de percepción estudiantil	25
1.5.4. Entrevista semiestructurada	25
1.5.5. Registro de observación sistemática	26
1.5.6. Indicadores de evaluación y criterios de validez	26
1.5.7. Justificación y cierre	27
1.6. Reflexión crítica y transferencia de la experiencia	31

Tabla de Contenidos

2. Del problema a la solución: estrategias para enseñar análisis y diseño de software en primer nivel	38
2.1. Apertura contextual y problematización de la experiencia docente	42
2.1.1. El problema formativo	42
2.1.2. Propósito de la sistematización	44
2.1.3. Criterios de valor	45
2.1.4. Delimitación del objeto de estudio	46
2.2. Fundamentación conceptual y operativa: del pensamiento computacional al aprendizaje activo	47
2.2.1. Transición hacia la fundamentación conceptual y operativa	47
2.2.2. Aprendizaje activo: del estudiante receptor al estudiante protagonista	49
2.2.3. Práctica colaborativa-reflexiva: aprender con otros para construir significado	50
2.2.4. Formulación de dimensiones	50
2.2.5. Dimensión colaborativa-reflexiva: aprender con otros para construir conocimiento	51
2.2.6. Construcción de indicadores	52
2.2.7. Dimensión pedagógica-didáctica: enseñar a analizar antes de programar	53
2.2.8. Dimensión colaborativa-reflexiva: aprender con otros para construir conocimiento	54
2.2.9. Cierre proyectivo	55
2.2.10. Fuentes y métodos de verificación	55
2.2.11. Fuentes de verificación	56
2.2.12. Métodos de verificación	56
2.2.13. Ejemplo de aplicación	57
2.2.14. Cierre proyectivo	57
2.2.15. Justificación teórica del conjunto	57
2.2.16. Integración	60
2.3. Vínculo curricular y resultados de aprendizaje en la enseñanza del análisis y diseño de software	60
2.3.1. Transición al vínculo curricular	60
2.3.2. Identificación de competencias del perfil	61
2.3.3. Resultados de aprendizaje vinculados	64
2.3.4. Actividades y evidencias	66
2.3.5. Reflexión sobre la alineación curricular	68
2.3.6. Integración curricular del Módulo 3	70
2.4. Del problema a la solución: estrategias para enseñar análisis y diseño de software en primer nivel	73
2.4.1. Transición hacia la operacionalización estratégica	73
2.4.2. Estrategias núcleo en acción	74

2.4.3. Estrategias de soporte aplicadas	76
2.4.4. Estrategias de contingencia desplegadas	79
2.4.5. Arquitectura del ecosistema estratégico	81
2.4.6. Integración: Justificación de la validez curricular del ecosistema estratégico	84
2.5. Evaluación e indicadores de logro: instrumentos, validez y análisis de evidencias	86
2.5.1. Transición hacia la evaluación	86
2.5.2. Instrumentos de evaluación aplicados	87
2.5.3. Indicadores de evaluación y criterios de validez	89
2.5.4. Indicadores aplicados	90
2.5.5. Criterios de validez y confiabilidad	91
2.5.6. Análisis preliminar de evidencias	93
2.5.7. Tipos de evidencias y organización del análisis	93
2.5.8. Hallazgos preliminares	94
2.5.9. Interpretación de patrones emergentes	94
2.5.10. Síntesis y proyección	95
2.5.11. Reflexión sobre validez, sesgos y factibilidad	95
2.5.12. Validez del proceso evaluativo	95
2.5.13. Sesgos identificados y estrategias de mitigación	96
2.5.14. Factibilidad y aprendizajes derivados	97
2.5.15. Síntesis y proyección	98
2.5.16. Integración: Síntesis de la evaluación	98
2.6. Del problema a la solución: estrategias para enseñar análisis y diseño de software en primer nivel	100
2.6.1. Transición hacia la reflexión final	100
2.6.2. Reflexión crítica sobre la experiencia	101
2.6.3. Aportes de la experiencia	101
2.6.4. Tensiones y resistencias encontradas	102
2.6.5. Aprendizajes personales, colectivos e institucionales	103
2.6.6. Síntesis reflexiva y proyección	104
2.6.7. Integración final: Reflexión y transferencia	104
2.6.8. Aportes y aprendizajes globales	105
2.6.9. Tensiones, desafíos y aprendizajes emergentes	106
2.6.10. Proyección y transferencia	106
2.6.11. Cierre del capítulo	107
3. La enseñanza del código limpio: estrategias para formar desarrolladores con estándares profesionales	111
3.1. Buenas prácticas de programación y refactorización	114
3.1.1. Contextualización de la experiencia pedagógica	114

Tabla de Contenidos

3.1.2.	Identificación del problema pedagógico: hacia las buenas prácticas de programación y refactorización	115
3.1.3.	El rol del docente como revisor de código pedagógico	116
3.1.4.	Cierre integrador	117
3.2.	Del código al pensamiento: enseñanza del código limpio en la formación inicial de ingenieros en software	117
3.2.1.	Bisagra Textual	118
3.2.2.	Identificación de conceptos estructurantes	118
3.2.3.	Formulación de dimensiones	119
3.2.4.	Construcción de indicadores	120
3.2.5.	Dimensión Cognitivo–Conceptual: Comprensión de la lógica y la abstracción algorítmica	121
3.2.6.	Dimensión Técnico–Procedimental: Aplicación de buenas prácticas y calidad del código	122
3.2.7.	Fuentes y métodos de verificación	123
3.2.8.	Modelar y abstraer procesos lógicos	124
3.2.9.	Comprensión de la secuencia algorítmica	124
3.2.10.	Aprendizaje significativo y contextualizado	125
3.2.11.	Fuentes complementarias y métodos asociados	125
3.2.12.	Síntesis integradora	125
3.2.13.	Justificación teórica del conjunto	126
3.2.14.	Justificación de los indicadores	127
3.2.15.	Justificación de las fuentes y métodos de verificación	127
3.2.16.	Síntesis final del conjunto	128
3.2.17.	Recomendaciones para profundizar	129
3.3.	Aprender gestionando: experiencias innovadoras en proyectos de software universitarios	129
3.3.1.	Transición al vínculo curricular	129
3.3.2.	Identificación de competencias del perfil	130
3.3.3.	Resultados de aprendizaje vinculados	133
3.3.4.	Actividades y evidencias	135
3.3.5.	Reflexión sobre la alineación curricular	137
3.4.	Transición hacia la operacionalización estratégica	139
3.4.1.	Recomendaciones para profundizar	139
3.4.2.	Clase 1: Estrategias núcleo en acción	140
3.4.3.	Clase 2: Estrategias de soporte aplicadas	143
3.4.4.	Clase 3: Estrategias de contingencia desplegadas	146
3.4.5.	Clase 4: Arquitectura del ecosistema estratégico	149
3.4.6.	Síntesis final: El ecosistema como sistema vivo	151
3.5.	Transición hacia la evaluación	151
3.5.1.	Recomendaciones para profundizar	152
3.5.2.	Clase 1: Instrumentos de evaluación aplicados	152

3.5.3. Recomendaciones para profundizar	155
3.5.4. Clase 2: Indicadores de evaluación y criterios de validez	155
3.5.5. Clase 3: Análisis preliminar de evidencias	158
3.5.6. Clase 4: Reflexión sobre validez, sesgos y factibilidad	160
3.6. Transición hacia la reflexión final	163
3.6.1. Clase 1: Reflexión crítica sobre la experiencia	164
3.6.2. Recomendaciones para profundizar	166
4. Aprender gestionando: experiencias innovadoras en proyectos de software universitarios	171
4.1. Aprender gestionando: experiencias innovadoras en proyectos de software universitarios	175
4.1.1. Apertura contextual	175
4.1.2. Problematización	175
4.1.3. Las consecuencias de este déficit se manifiestan en tres niveles	176
4.1.4. Propósito de la sistematización	176
4.1.5. Criterios de valor	177
4.1.6. Delimitación del objeto de estudio	178
4.2. Fundamentación teórico-metodológica de la experiencia docente	179
4.2.1. Bisagra textual	179
4.2.2. Identificación de conceptos estructurantes	180
4.2.3. Aprendizaje activo	180
4.2.4. Pensamiento crítico	181
4.2.5. Autonomía en el aprendizaje	181
4.2.6. Aprendizaje colaborativo	182
4.2.7. Mentoría pedagógica	182
4.2.8. Relación entre los conceptos	183
4.2.9. Formulación de dimensiones	183
4.2.10. Dimensión pedagógica	184
4.2.11. Dimensión cognitiva-formativa	185
4.2.12. Dimensión socioafectiva-colaborativa	185
4.2.13. Construcción de indicadores	186
4.2.14. Indicadores de la dimensión pedagógica	187
4.2.15. Indicadores de la dimensión cognitiva-formativa	188
4.2.16. Indicadores de la dimensión socioafectiva-colaborativa	188
4.2.17. Fuentes	190
4.2.18. Métodos de verificación	191
4.2.19. Análisis comparativo de productos	191
4.2.20. Análisis de contenido	192
4.2.21. Triangulación de evidencias	192
4.2.22. Justificación teórica del conjunto	193

Tabla de Contenidos

4.3. Integración curricular y desarrollo de competencias en la formación del ingeniero de software	195
4.3.1. Transición al vínculo curricular	195
4.3.2. Identificación de competencias del perfil	196
4.3.3. Resultados de aprendizaje vinculados	198
4.3.4. Actividades y evidencias	200
4.3.5. Reflexión sobre la alineación curricular	202
4.4. Diseño e implementación del ecosistema estratégico de aprendizaje en ingeniería de software	204
4.4.1. Transición hacia la operacionalización estratégica	204
4.4.2. Estrategias núcleo en acción	205
4.4.3. Estrategias de soporte aplicadas	207
4.4.4. Estrategias de contingencia desplegadas	209
4.4.5. Arquitectura del ecosistema estratégico	211
4.5. Evaluación integral de la experiencia: instrumentos, evidencias y validez del proceso formativo	213
4.5.1. Transición hacia la evaluación	213
4.5.2. Instrumentos de evaluación aplicados	214
4.5.3. Indicadores de evaluación y criterios de validez	216
4.5.4. Análisis preliminar de evidencias	218
4.5.5. Reflexión sobre validez, sesgos y factibilidad	220
4.6. Reflexión final y proyección institucional de la experiencia docente	222
4.6.1. Transición hacia la reflexión final	222
4.6.2. Reflexión crítica sobre la experiencia	223
4.6.3. Tensiones y resistencias encontradas	224
4.6.4. Aprendizajes personales, colectivos e institucionales	225
4.6.5. Síntesis reflexiva	226

1

Fundamentos de ingeniería de software: de la práctica docente reflexiva a la formación por competencias

Mariuxi G. Vinueza-Morales ¹

El capítulo sistematiza una experiencia pedagógica desarrollada en la Universidad Estatal de Milagro (UNEMI) en la asignatura Introducción a la Ingeniería de Software. A través de seis etapas, se analiza la transición conceptual de los estudiantes desde una visión reduccionista centrada en la programación hacia una comprensión integral del proceso de desarrollo de software. La propuesta combina fundamentos teóricos, estrategias activas y herramientas de evaluación válidas y confiables, evidenciando mejoras en el aprendizaje, la reflexión docente y la coherencia curricular. La experiencia demuestra la viabilidad de un modelo formativo basado en competencias, reflexión y mejora continua.

¹Universidad Estatal de Milagro, mvinuezam@unemi.edu.ec.

Índice

1.1. Fundamentos de Ingeniería de Software: del aula a la práctica profesional reflexiva	4
1.1.1. Apertura contextual	4
1.1.2. Problematización	5
1.1.3. Propósito de la sistematización	5
1.1.4. Criterios de valor	6
1.1.5. Delimitación del objeto de estudio	7
1.2. Fundamentación conceptual y operativa de la experiencia	8
1.2.1. Transición hacia la fundamentación	8
1.2.2. Identificación de conceptos estructurantes	9
1.2.3. Formulación de dimensiones	10
1.2.4. Construcción de indicadores	10
1.2.5. Fuentes y métodos de verificación	13
1.2.6. Justificación teórica del conjunto	13
1.2.7. Recapitulación y proyección	14
1.3. Vínculo con el currículo y el perfil de la Carrera	15
1.3.1. Transición hacia la fundamentación	15
1.3.2. Identificación de competencias del perfil	15
1.3.3. Resultados de aprendizaje vinculados	16
1.3.4. Actividades y evidencias de aprendizaje	16
1.3.5. Reflexión sobre la alineación curricular	18
1.3.6. Síntesis final del módulo	19
1.4. Ecosistema estratégico (estrategias y relaciones)	20
1.4.1. Puente de Transición hacia el Análisis Estratégico	20
1.4.2. Estrategias de Soporte	21
1.4.3. Estrategias de Contingencia	22
1.4.4. Integración Estratégica y Proyección hacia la Evaluación	23

1.5. Evaluación, indicadores, instrumentos, análisis	24
1.5.1. Puente de Evaluación: Instrumentos, Indicadores y Criterios de Validez	24
1.5.2. Rúbrica analítica de desempeño	25
1.5.3. Cuestionario de percepción estudiantil	25
1.5.4. Entrevista semiestructurada	25
1.5.5. Registro de observación sistemática	26
1.5.6. Indicadores de evaluación y criterios de validez	26
1.5.7. Justificación y cierre	27
1.6. Reflexión crítica y transferencia de la experiencia	31

1.1. Fundamentos de Ingeniería de Software: del aula a la práctica profesional reflexiva

1.1.1. Apertura contextual

La experiencia que aquí se sistematiza tuvo lugar en la Universidad Estatal de Milagro, en la asignatura *Introducción a la Ingeniería de Software*, dentro de la carrera de Ingeniería de Software. Se trata de un curso inicial en el que convergen estudiantes jóvenes, en su mayoría hombres y provenientes de diferentes provincias, que buscan insertarse en un campo profesional de alta demanda. El aula se convierte en un espacio de diversidad cultural y académica, donde confluyen expectativas, motivaciones y vacíos conceptuales que condicionan los procesos de aprendizaje.

Durante una sesión inicial, al plantear un ejercicio práctico sobre cómo abordar un pedido de cliente, la mayoría de los estudiantes propuso comenzar directamente a programar sin considerar posibles cambios de requerimientos. Esta respuesta reflejó una tendencia general: asociar la ingeniería de software exclusivamente con la escritura de código, sin visualizar los procesos de planificación, diseño y calidad. Este episodio constituyó un punto de inflexión, pues mostró con claridad la necesidad de trabajar en la transición conceptual desde una mirada reduccionista hacia una comprensión integral de la disciplina.

Las condiciones que favorecieron el desarrollo de la experiencia incluyen la apertura de los estudiantes a reflexionar sobre su propia práctica, así como el interés institucional por fortalecer la formación de base en ingeniería de software. Entre las limitaciones encontradas se destacan la resistencia inicial a cuestionar creencias previas y la escasez de materiales didácticos adaptados al contexto local. Este escenario se convierte en el cimiento para la sistematización, ya que visibiliza el desafío central: diferenciar entre “programar” y “construir software”, situando este contraste como el núcleo de la experiencia formativa.

Me parece clave que piensen en términos de diagnóstico. En la epistemología del sur se viene cuestionando el uso del término “diagnóstico” hacia un análisis situacional. Ustedes ayer presentaron distintos autores que los alimentan teóricamente. Yo les traigo uno que cuestiona el uso de la palabra “diagnóstico” porque dice que es solo formulativo. Hacer un análisis situacional implica ir más allá de solo datos, es poder intrincarme con el otro.

1.1.2. Problematización

El problema central identificado en esta experiencia es la visión limitada que los estudiantes tienen sobre la ingeniería de software, restringida casi exclusivamente al acto de programar. Este enfoque reduccionista no solo empobrece su formación académica, sino que también genera implicaciones profesionales, ya que limita la capacidad de abordar proyectos con calidad, sostenibilidad y visión de procesos.

La relevancia de este problema se refleja en la práctica profesional: los egresados que priorizan únicamente la codificación suelen enfrentar dificultades para responder a cambios de requerimientos, gestionar equipos o implementar procesos de calidad (Pressman, 2014). En el aula, esta tendencia se traduce en propuestas desorganizadas, donde los estudiantes privilegian soluciones rápidas sin prever la evolución del producto. Tal desconexión entre formación académica y necesidades del campo evidencia la urgencia de una intervención pedagógica.

Desde una perspectiva educativa, Schön (1992) sostiene que la práctica reflexiva permite a los profesionales replantearse sus marcos de acción, lo que resulta clave para enfrentar problemas complejos. En la experiencia descrita, el quiebre inicial al confrontar a los estudiantes con la pregunta sobre cómo responder a un cliente mostró que existía un vacío conceptual profundo. Esta evidencia práctica, combinada con estudios como los de McConnell (2004), que advierten sobre los riesgos de no formar en procesos desde etapas tempranas, confirma la pertinencia de situar este problema como núcleo de la sistematización.

La problematización, por tanto, no se limita a describir una dificultad puntual, sino que revela un desafío formativo estructural: cómo guiar a los estudiantes en la transición de una visión técnica fragmentada hacia una concepción integral de la ingeniería de software. Este planteamiento será la base para definir el propósito y orientar las decisiones narrativas del capítulo.

1.1.3. Propósito de la sistematización

El propósito de esta sistematización es documentar y analizar la estrategia pedagógica que favoreció la transición conceptual de los estudiantes desde una visión reduccionista hacia una comprensión procesual e integral de la ingeniería de software. Con ello se busca formalizar una propuesta didáctica que, además de ser válida para el contexto de la Uni-

versidad Estatal de Milagro, pueda inspirar a otros docentes e instituciones interesadas en fortalecer la formación inicial en ingeniería.

Esta intención surge de la convicción de que enseñar ingeniería de software no puede reducirse a transmitir técnicas, sino que debe acompañar la construcción de un marco conceptual sólido que prepare a los estudiantes para los retos de la práctica profesional. En este sentido, la sistematización se convierte en una herramienta de reflexión y comunicación, alineada con lo que Carlino (2005) y Hyland (2009) plantean respecto a la escritura académica como medio para compartir procesos invisibilizados y generar redes de aprendizaje.

El propósito tiene, además, una dimensión proyectiva. Como docentes, buscamos transformar una experiencia situada en conocimiento replicable, con la capacidad de incidir en la cultura pedagógica de la disciplina. Este propósito se enlaza con lo propuesto por Somerville (2011), al destacar la importancia de comprender la ingeniería de software como un conjunto de procesos interrelacionados, y con los principios del *Manifiesto Ágil* (Beck et al., 2001), que enfatizan la adaptabilidad y la mejora continua. En consecuencia, el propósito se constituye en brújula que orienta tanto la narrativa del capítulo como el aporte a la comunidad académica.

1.1.4. Criterios de valor

El valor de esta experiencia radica en demostrar que la enseñanza de la ingeniería de software puede convertirse en un espacio de innovación pedagógica cuando integra reflexión crítica y aprendizaje técnico. No se trata solo de transmitir conocimientos, sino de acompañar un proceso formativo que transforma la forma en que los estudiantes entienden la disciplina.

La innovación se fundamenta en haber vinculado la metodología de sistematización educativa (Jara, 2018a) con la enseñanza de fundamentos técnicos, generando un modelo pedagógico reflexivo. En lugar de limitarse a prácticas centradas en la programación, la experiencia promovió que los estudiantes asumieran una visión integral basada en procesos. Esto se inscribe en la lógica de la investigación-acción que Elliott (1993) describe, donde el docente convierte su práctica en objeto de análisis y conocimiento compartido.

El impacto se observó en varios niveles. En los estudiantes, se produjo una evolución desde la lógica del código hacia la valoración de la calidad y los procesos (McConnell, 2004). En los docentes, se consolidó un marco para abordar brechas conceptuales recu-

rrentes. Para la institución, los resultados ofrecen insumos validados para la mejora curricular, en sintonía con estándares internacionales como ABET (2020). Esta dimensión reflexiva remite a Schön (1992), quien subraya la importancia de aprender a partir de la práctica.

La transferibilidad es otro criterio clave. El contraste entre desarrollo ad hoc y modelos formales, las actividades de reflexión guiada y los instrumentos de evaluación conceptual pueden adaptarse a otros contextos. Stenhouse (1987) señala que el conocimiento pedagógico cobra sentido cuando es compartido y puesto a prueba en nuevos escenarios. En este caso, la coherencia con los principios del *Manifiesto Ágil* (Beck et al., 2001) y marcos de calidad como ANEAES (2020) refuerza la validez de la propuesta.

En conclusión, la experiencia merece ser considerada valiosa porque conjuga innovación, impacto y transferibilidad. Como sostiene Jara (2018a), la sistematización adquiere sentido cuando trasciende la práctica individual y se convierte en conocimiento compartido. Con este cierre se abre camino a la delimitación del objeto de estudio, paso que otorga foco y coherencia a la narrativa del capítulo.

1.1.5. Delimitación del objeto de estudio

El objeto de estudio de esta sistematización se centra en el proceso de transición conceptual de los estudiantes desde una visión centrada en la programación hacia una comprensión procesual de la ingeniería de software. El propósito es analizar cómo la estrategia pedagógica implementada favoreció este cambio, documentando tanto las dinámicas de aula como las evidencias producidas por los estudiantes.

La elección de este foco responde a la necesidad de destacar el aspecto más significativo de la experiencia, evitando dispersión. No se abordan de manera exhaustiva los efectos institucionales ni la totalidad de resultados docentes, sino que se delimitan como eje principal las prácticas didácticas y el aprendizaje estudiantil. Flick (2014a) sostiene que toda investigación cualitativa requiere un recorte definido para alcanzar profundidad y coherencia, criterio que guía esta delimitación.

El recorte incluye a los estudiantes de primer semestre de la carrera de Ingeniería de Software de la Universidad Estatal de Milagro, así como el rol docente como mediador. Las evidencias centrales consideradas son registros de aula, producciones de los estudiantes en actividades de contraste y evaluaciones que reflejan la comprensión conceptual. Se

excluyen seguimientos longitudinales y valoraciones externas posteriores, al exceder los límites temporales y temáticos del análisis.

Este objeto de estudio parte del supuesto de que comprender los procesos de ingeniería desde etapas tempranas constituye un factor decisivo para la formación profesional. Tal elección responde a lo señalado por Jara (2018a), para quien sistematizar no es abarcarlo todo, sino convertir una práctica particular en conocimiento comunicable y útil.

En síntesis, el capítulo se construirá en torno al análisis de esta transición conceptual, con un marco definido que asegura coherencia y pertinencia. Con ello se cierra el primer tramo de escritura, sobre el cual se edificará el desarrollo posterior.

1.2. Fundamentación conceptual y operativa de la experiencia

La reflexión desarrollada en el capítulo anterior permitió delimitar el objeto de estudio y fundamentar la relevancia de analizar la transición conceptual de los estudiantes desde una visión reducida de la ingeniería de software —centrada exclusivamente en la programación— hacia una comprensión integral del proceso de desarrollo. Sobre esa base, el presente capítulo se orienta a establecer los fundamentos teóricos y metodológicos que sustentan dicha sistematización, articulando los conceptos estructurantes, dimensiones, indicadores y métodos que consolidan el marco conceptual de la experiencia formativa desarrollada en la Universidad Estatal de Milagro (UNEMI).

1.2.1. Transición hacia la fundamentación

La sistematización de la experiencia desarrollada en la Universidad Estatal de Milagro (UNEMI) surge como una estrategia formativa orientada al fortalecimiento de la escritura académica de los docentes universitarios. Este proceso busca promover la reflexión pedagógica, la mejora continua y la consolidación de una cultura institucional de investigación. El Módulo 2, “Fundamentación conceptual y operativa de la experiencia”, constituye el corazón del proceso, ya que en él se articulan los conceptos, dimensiones, indicadores, fuentes y métodos que sostienen teóricamente la práctica sistematizada.

De acuerdo con Jara (2018b), sistematizar es “volver sobre la experiencia para comprenderla y comunicar”, lo cual exige una estructura conceptual clara y coherente. En

este sentido, la UNEMI, comprometida con la formación de docentes investigadores y con la innovación pedagógica, encuentra en este ejercicio una oportunidad para construir conocimiento colectivo desde su propia práctica.

1.2.2. Identificación de conceptos estructurantes

La identificación de los conceptos estructurantes fue el punto de partida para construir el marco teórico de esta sistematización. En el contexto de la UNEMI, los conceptos escritura académica, acompañamiento docente, práctica reflexiva e identidad profesional emergieron de las experiencias formativas y de los procesos de tutoría desarrollados con los docentes. Estos conceptos permiten comprender la experiencia desde una perspectiva integral, al combinar el desarrollo de competencias comunicativas con la transformación del rol docente.

La escritura académica, como plantean Carlino (2005) y Hyland (2009), es una práctica social situada, en la que los sujetos construyen conocimiento, identidad y pertenencia disciplinar. No se trata solo de producir textos correctos, sino de aprender a pensar, argumentar y comunicar en los lenguajes de la ciencia. En la UNEMI, este concepto se traduce en el fomento de la alfabetización académica como parte del desarrollo profesional docente, en concordancia con las líneas de investigación institucionales.

El acompañamiento docente, por su parte, se entiende como una mediación pedagógica orientada al desarrollo reflexivo y colaborativo. Schön (1992) describe este proceso como “reflexión en la acción”, una práctica donde el docente aprende al analizar sus propias decisiones y experiencias. En la UNEMI, este acompañamiento ha adoptado un enfoque dialógico, que combina tutorías personalizadas, talleres de escritura y espacios de coevaluación, generando comunidades de aprendizaje entre pares.

Finalmente, la práctica reflexiva y la identidad profesional actúan como ejes transversales del proceso. Day (2006) sostiene que la identidad docente se construye en la interacción entre la biografía personal, el contexto institucional y los valores profesionales. En este sentido, las tutorías de escritura no solo fortalecen habilidades técnicas, sino que contribuyen a la consolidación de una identidad académica comprometida con la calidad y la investigación.

1.2.3. Formulación de dimensiones

La construcción de dimensiones permitió organizar y profundizar el análisis de la experiencia. Según Flick (2014b), las dimensiones actúan como categorías intermedias que permiten traducir la complejidad de una práctica en un sistema interpretativo coherente. En este proceso, se definieron tres dimensiones: pedagógica, institucional y subjetiva, las cuales articulan los aspectos formativos, estructurales y personales de la experiencia.

La dimensión pedagógica aborda las estrategias, metodologías y recursos que sustentan el acompañamiento docente en la escritura académica. Incluye la planificación de talleres, la retroalimentación formativa y la integración de la escritura en las prácticas docentes. Como plantea Zabalza (2011), esta dimensión permite valorar la competencia didáctica del profesor universitario no solo en su conocimiento disciplinar, sino también en su capacidad para guiar procesos de aprendizaje autónomo. En la UNEMI, esta dimensión cobra relevancia dentro de su modelo educativo basado en competencias y aprendizaje activo.

La dimensión institucional refleja el compromiso de la UNEMI con la profesionalización docente y la mejora continua. Esta universidad ha impulsado políticas de fortalecimiento académico, como la formación continua, la vinculación con la investigación y la integración de tecnologías en los procesos educativos. Fullan y Hargreaves (2012) destacan que las instituciones que promueven el desarrollo profesional colectivo logran consolidar comunidades de práctica sostenibles, lo que coincide con los propósitos del acompañamiento desarrollado.

Por su parte, la dimensión subjetiva da cuenta de los cambios en la percepción, motivación y autoconfianza del profesorado. Freire (1996) recuerda que todo acto educativo implica una dimensión ética y emocional; por tanto, el crecimiento profesional se acompaña de una transformación personal. En este sentido, la experiencia en la UNEMI evidenció cómo la escritura académica se convierte también en un espacio de empoderamiento y resignificación del rol docente.

1.2.4. Construcción de indicadores

Los indicadores representan el paso del plano conceptual al empírico, constituyendo herramientas concretas para evaluar la transformación de las prácticas. Yin (2014) explica que los indicadores son los puentes entre las categorías teóricas y la realidad observada; su función es permitir la verificación y comprensión de los fenómenos estudiados. En esta

sistematización, los indicadores se construyeron a partir de los registros de tutorías, los textos producidos y las reflexiones docentes.

En la dimensión pedagógica, los indicadores incluyeron la frecuencia de integración de actividades de escritura en el aula, la calidad de la retroalimentación entre pares y la adopción de estrategias de revisión. En la dimensión institucional, se consideraron la participación en programas de capacitación, el uso de plataformas académicas y el acceso a recursos institucionales. En la dimensión subjetiva, se evaluó la autopercepción del cambio, el nivel de motivación y la apropiación del rol académico. Stake (1995) subraya que los indicadores solo adquieren sentido cuando están acompañados de evidencias verificables; por ello, su construcción estuvo guiada por criterios de claridad, pertinencia y observabilidad.

La matriz de dimensiones–indicadores–fuentes–métodos constituye la síntesis del trabajo desarrollado en el Módulo 2 y cumple la función de articular el marco conceptual con la práctica observada. En ella se organizan los aspectos pedagógicos, institucionales y subjetivos de la experiencia, mostrando cómo se tradujeron en indicadores observables y verificables, respaldados con evidencias concretas y métodos de análisis rigurosos.

En la dimensión pedagógica, los indicadores se reflejaron en la calidad progresiva de los textos académicos de los docentes, evidenciada en borradores, versiones finales y registros de tutoría. Estos materiales se analizaron mediante la comparación entre productos y el análisis documental, lo que permitió constatar mejoras en coherencia, cohesión y uso de referencias.

En la dimensión institucional, los indicadores se verificaron en registros formales como actas de tutoría, certificaciones y documentos del plan de desarrollo docente. Su análisis documental y la triangulación con otras fuentes demostraron el compromiso de la universidad para respaldar y legitimar el proceso.

Finalmente, en la dimensión subjetiva, los testimonios recogidos en entrevistas, grupos focales y relatos escritos permitieron verificar cambios en la confianza, la motivación y la identidad profesional de los participantes. Para esta fuente se aplicó el análisis de contenido y la codificación temática, lo que facilitó interpretar las narrativas como evidencia de transformación personal.

En conjunto, la matriz muestra cómo las dimensiones analíticas no quedaron en el plano abstracto, sino que se vincularon con indicadores claros, sustentados en fuentes pertinentes y validadas mediante métodos cualitativos sólidos. De este modo, se garantiza

Tabla 1.1: Matriz de dimensiones, indicadores, fuentes, métodos

Dimensión	Indicadores	Fuentes de verificación	Métodos de verificación
Pedagógica	1. Los docentes entregan borradores semanales con estructura académica básica. 2. Incorporan citas y referencias en sus textos. 3. Mejoran la coherencia y cohesión entre versiones iniciales y finales.	- Borradores de textos - Versiones finales - Observaciones de tutoría	- Comparación entre versiones - Análisis textual y documental
Institucional	1. La universidad certifica la participación de los docentes. 2. Se asignan horas institucionales para el proceso de formación. 3. Se reconoce formalmente la capacitación dentro del plan de desarrollo docente.	- Certificaciones - Actas de tutoría - Plan de desarrollo docente	- Análisis documental - Triangulación con otras fuentes
Subjetiva	1. Los docentes expresan mayor confianza en sus escritos. 2. Participan voluntariamente en espacios de retroalimentación. 3. Reconocen avances en su identidad como autores de conocimiento.	- Entrevistas - Grupos focales - Relatos personales escritos	- Análisis de contenido - Codificación temática - Triangulación

Fuente: elaboración propia.

que la sistematización se base en evidencias confiables y se fortalezca su credibilidad académica

1.2.5. Fuentes y métodos de verificación

La validez del proceso depende de la rigurosidad en la selección de fuentes y métodos. Flick (2014a) recomienda la triangulación de fuentes como estrategia fundamental para garantizar la credibilidad del análisis cualitativo. En este caso, se utilizaron diversas fuentes: borradores de escritura, entrevistas semiestructuradas, diarios reflexivos y documentos institucionales. Esta diversidad permitió observar la experiencia desde múltiples perspectivas.

El método de análisis de contenido facilitó la identificación de patrones y categorías emergentes en los textos de los docentes; mientras que el análisis comparativo ayudó a contrastar los avances en distintas etapas del proceso. Además, el análisis narrativo permitió reconstruir las trayectorias formativas de los participantes, otorgando profundidad interpretativa. Yin (2014) señala que el valor de un estudio de caso radica en su capacidad para conectar las evidencias con el marco conceptual, y eso se logró al relacionar cada fuente con los indicadores definidos.

1.2.6. Justificación teórica del conjunto

La coherencia del conjunto conceptual y operativo responde a una lógica fundamentada en la teoría educativa y la metodología cualitativa. Jara (2018a) plantea que la sistematización no solo organiza la experiencia, sino que la convierte en conocimiento socialmente útil. En este caso, la articulación entre los conceptos, las dimensiones y los métodos genera una estructura analítica sólida, coherente con la misión institucional de la UNEMI, que busca “formar profesionales competentes con actitud proactiva y valores éticos, desarrollar investigación relevante y ofertar servicios que contribuyan al desarrollo de la sociedad”.

Carlino (2005) y Hyland (2009) aportan el sustento epistemológico al concebir la escritura académica como práctica social que construye conocimiento y comunidad. Flick (2014a) y Yin (2014) sustentan el rigor metodológico, mientras que Freire (1996) y Schön (1992) aportan la mirada ética y reflexiva del proceso docente. Este entramado teórico legitima las decisiones tomadas y permite proyectar el trabajo hacia la mejora continua de la formación académica y profesional del profesorado universitario.

Además, la justificación teórica se fortalece al reconocer que este proceso se inscribe en una política institucional de UNEMI orientada a la innovación educativa y la producción científica. Tal como sostiene Bolívar (2012), la cultura organizativa de una institución determina la sostenibilidad de los cambios pedagógicos; por ello, integrar la escritura académica dentro del desarrollo docente constituye un paso decisivo hacia la consolidación de una comunidad universitaria más reflexiva y generadora de conocimiento. Este marco teórico-operativo, al articular lo individual con lo institucional, reafirma que la sistematización no es un ejercicio aislado, sino una práctica que potencia el sentido de pertenencia y el compromiso con la transformación educativa.

1.2.7. Recapitulación y proyección

Este módulo me permitió articular los fundamentos conceptuales y metodológicos que sustentan la sistematización de mi experiencia en la UNEMI. A través de los cinco puentes recorridos, logré integrar los conceptos de escritura académica, acompañamiento docente y reflexión pedagógica en un marco coherente, que se traduce en dimensiones analíticas con sus respectivos indicadores, fuentes y métodos de verificación. Este proceso no sólo consolidó la estructura del capítulo, sino que fortaleció mi comprensión sobre cómo la práctica puede convertirse en conocimiento comunicable.

La integración de estos elementos me da la seguridad de que el trabajo realizado cuenta con validez académica y sentido formativo. Haber completado esta fundamentación se prepara para avanzar hacia el Módulo 3, centrado en el análisis e interpretación de los hallazgos. Desde la perspectiva institucional de la UNEMI, este avance representa una contribución concreta al fortalecimiento de la cultura investigativa y reflexiva del profesorado, reafirmando que la escritura académica, más que una exigencia, es una práctica de crecimiento profesional y colectivo.

En síntesis, la fundamentación conceptual y operativa presentada en este capítulo permitió consolidar la estructura metodológica que sustenta la sistematización de la experiencia. La articulación entre conceptos, dimensiones, indicadores y métodos dio coherencia al proceso y garantizó su validez académica. Este recorrido teórico-práctico constituye la base para el siguiente capítulo, donde se abordará el análisis e interpretación de los hallazgos empíricos, con el propósito de visibilizar los aprendizajes, transformaciones y desafíos que emergen de la práctica reflexiva y del acompañamiento docente en la UNEMI.

1.3. Vínculo con el currículo y el perfil de la Carrera

1.3.1. Transición hacia la fundamentación

El Módulo 3 representa un punto de inflexión dentro del proceso de sistematización de la experiencia docente, al centrarse en la **integración curricular** como eje articulador entre la práctica pedagógica y los propósitos formativos institucionales. En este módulo, la reflexión se orienta a establecer conexiones claras entre la experiencia desarrollada, las **competencias del perfil de egreso**, los **resultados de aprendizaje**, las **actividades realizadas**, las **evidencias generadas** y la **alineación curricular** que sustenta la calidad de la enseñanza en la Universidad Estatal de Milagro (UNEMI).

La UNEMI, en coherencia con su Modelo Educativo Institucional (Universidad Estatal de Milagro (UNEMI), 2022), promueve un enfoque por competencias que busca formar profesionales con pensamiento crítico, compromiso ético y capacidad de innovación. En este contexto, la sistematización docente se convierte en una herramienta de investigación y mejora, permitiendo documentar cómo las estrategias pedagógicas implementadas en el aula dialogan con el currículo y aportan al perfil de egreso.

El propósito general del Módulo 3 es evidenciar la **coherencia entre lo planificado y lo aprendido**, mostrando que cada acción pedagógica responde a un marco curricular que legitima y potencia su valor formativo. A través de los cinco puentes que lo componen —desde la identificación de competencias hasta la integración final—, se construye una narrativa reflexiva que da cuenta del impacto académico de la experiencia, fortaleciendo tanto el proceso de enseñanza-aprendizaje como el desarrollo institucional.

1.3.2. Identificación de competencias del perfil

El recorrido del Módulo 3 inició con la identificación de las **competencias profesionales y transversales** que sustentan el perfil de egreso de la carrera de Ingeniería de Software de la UNEMI. A partir de ellas, se analizaron los **resultados de aprendizaje** que operativizan dichas competencias, revelando una coherencia entre los propósitos formativos institucionales y la práctica docente desarrollada. Este proceso permitió reconocer que las competencias específicas —como la capacidad de analizar, diseñar e implementar soluciones informáticas eficaces— se complementan con competencias genéricas como el trabajo colaborativo, la comunicación efectiva y el pensamiento crítico.

En este primer puente se estableció que la práctica pedagógica no puede limitarse a la transmisión técnica de conocimientos, sino que debe orientarse al desarrollo de **capacidades integrales**. La articulación entre competencias y resultados de aprendizaje proporcionó una base conceptual sólida que guio todo el proceso de sistematización, situando la experiencia dentro del **marco curricular institucional** (Universidad Estatal de Milagro (UNEMI), 2022). Este ejercicio inicial reafirma la importancia de un currículo dinámico, centrado en el estudiante y coherente con los desafíos tecnológicos y sociales contemporáneos.

1.3.3. Resultados de aprendizaje vinculados

En este segundo momento, la atención se centró en mostrar cómo las **actividades desarrolladas** durante la experiencia docente materializaron los resultados de aprendizaje previstos. Se diseñaron y aplicaron estrategias didácticas como talleres de análisis de casos con requerimientos cambiantes, elaboración progresiva de documentos de diseño, sesiones de retroalimentación entre pares y ejercicios de reflexión guiada sobre el proceso de desarrollo.

Cada una de estas actividades generó **evidencias verificables** —informes técnicos, rúbricas de coevaluación, registros audiovisuales de reflexión— que validaron los logros alcanzados. Esta relación actividad–resultado–evidencia demostró la coherencia pedagógica de la práctica, tal como propone Biggs (2003) en su modelo de Alineamiento Constructivo y Shulman (2005) en su planteamiento sobre la documentación reflexiva de la enseñanza. Así, se comprobó que la experiencia no fue improvisada, sino que se estructuró con propósito curricular, generando aprendizaje profundo y demostrable.

Este puente aportó evidencia empírica al proceso de sistematización, mostrando que la planificación didáctica, cuando se articula con los resultados esperados, se convierte en un motor de calidad educativa (Bolívar, 2016).

1.3.4. Actividades y evidencias de aprendizaje

El tercer puente permitió realizar una mirada crítica sobre la **articulación entre la experiencia y el currículo**. Se evidenció que la práctica docente contribuye significativamente a la calidad formativa de la carrera al vincular teoría y práctica mediante metodologías activas, reflexivas y colaborativas. A la vez, se identificaron tensiones derivadas de la

implementación de enfoques innovadores, como la resistencia inicial de algunos actores académicos o la necesidad de ajustar tiempos y recursos para sostener el trabajo reflexivo.

No obstante, estas tensiones se transformaron en oportunidades de aprendizaje y mejora institucional. La alineación curricular fortaleció la comprensión del perfil de egreso, evidenciando que la práctica docente es un espacio de innovación que puede alimentar la renovación del currículo. Esta reflexión se apoyó en los aportes de Díaz-Barriga (2014), quien señala la importancia de un currículo flexible y contextualizado, y de Bolívar (2016), quien destaca la función formativa del docente investigador de su propia práctica.

Esta práctica promovió la capacidad crítica, la argumentación técnica y la empatía profesional, aspectos fundamentales en la formación de ingenieros reflexivos (Schön, 1992). Las actas de retroalimentación y los comentarios escritos constituyeron evidencias tangibles del desarrollo de habilidades de evaluación y comunicación, en consonancia con la competencia genérica de trabajo colaborativo y comunicación efectiva establecida por la Universidad Estatal de Milagro (UNEMI) (2022).

La última fase de la experiencia incluyó **ejercicios de reflexión guiada sobre el proceso de desarrollo**, en los que los estudiantes analizaron los aprendizajes obtenidos, las dificultades enfrentadas y las decisiones técnicas adoptadas. Este ejercicio permitió generar evidencias cualitativas —como informes de autoevaluación y testimonios audiovisuales— que documentan la interiorización de una práctica profesional consciente. Stake (1995) señala que la triangulación de evidencias, es decir, la combinación de documentos, observaciones y reflexiones otorga mayor validez al proceso de evaluación educativa, al permitir contrastar percepciones y resultados de distintas fuentes.

De manera sintética, la relación entre **actividad–resultado de aprendizaje–evidencia** puede expresarse así:

Tabla 1.2: Relación entre actividad, resultado de aprendizaje y evidencia generada

Actividad	Resultado de aprendizaje trabajado	Evidencia generada
Taller de análisis de casos	Distingue las fases del ciclo de vida del software	Informe técnico con diagramas de proceso
Elaboración de documentos técnicos	Redacta especificaciones y decisiones de diseño	Borradores y entregas finales de documentación
Retroalimentación entre pares	Aplica pensamiento crítico y comunicación técnica	Rúbricas y actas de coevaluación
Reflexión guiada	Evalúa críticamente su proceso de aprendizaje	Informes escritos y testimonios audiovisuales

Nota. La tabla sintetiza la articulación entre actividad, resultado de aprendizaje y evidencia generada.
Fuente: elaboración propia.

Esta trazabilidad demuestra la coherencia pedagógica y curricular de la experiencia, asegurando que los resultados observados no son producto del azar, sino de una secuencia planificada que integra teoría, práctica y reflexión. En consonancia con el modelo educativo de la Universidad Estatal de Milagro (UNEMI) (2022), las evidencias generadas se convierten en insumos para la evaluación integral de los aprendizajes y para la mejora continua de la docencia universitaria. Así, este puente permite cerrar el ciclo formativo de la experiencia, mostrando cómo la práctica concreta del aula se articula con los propósitos institucionales de formación profesional.

1.3.5. Reflexión sobre la alineación curricular

A partir de la experiencia, se identificaron tres dimensiones clave del proceso de alineación: **fortalezas, tensiones y aprendizajes**. Entre las fortalezas, destaca la coherencia lograda entre competencias, resultados, actividades y evidencias, lo cual refleja la madurez pedagógica alcanzada. Entre las tensiones, se reconocen los desafíos asociados a la implementación de metodologías activas en contextos de enseñanza tradicional, lo que demanda acompañamiento institucional y desarrollo profesional docente. Finalmente, los aprendizajes obtenidos confirman que la sistematización es un proceso de construcción colectiva que promueve una cultura de mejora continua (Jara, 2018a).

La experiencia reafirma que la práctica reflexiva es un medio eficaz para integrar el

Tabla 1.3: Dimensiones, descripciones e implicaciones pedagógicas

Dimensión	Descripción	Implicaciones pedagógicas
Fortalezas	Coherencia entre competencias, resultados, actividades y evidencias; articulación efectiva con el currículo institucional.	Fortalece la calidad académica y la pertinencia del aprendizaje.
Tensiones	Resistencias al cambio metodológico; limitaciones de tiempo y recursos; necesidad de ajustes evaluativos.	Requiere apoyo institucional y capacitación docente continua.
Aprendizajes	La sistematización como práctica reflexiva; la importancia de documentar evidencias y compartir experiencias.	Consolida una cultura de mejora y colaboración docente.

Nota: La tabla sintetiza las principales dimensiones analizadas y sus implicaciones pedagógicas.

Fuente: elaboración propia.

saber académico con la realidad profesional, fortaleciendo la formación de ingenieros capaces de responder a los retos tecnológicos y sociales del entorno. Como señala **Schön (1992)**, el profesional reflexivo aprende en y sobre la acción, convirtiendo la práctica en una fuente legítima de conocimiento pedagógico.

1.3.6. Síntesis final del módulo

Desde una perspectiva integradora, el proceso desarrollado en el Módulo 3 constituye una manifestación concreta del **alineamiento constructivo** (Biggs & Tang, 2011), aplicada al contexto universitario ecuatoriano. La experiencia docente sistematizada demuestra que los aprendizajes significativos surgen de la coherencia entre lo que se enseña, cómo se enseña y cómo se evalúa. La **UNEMI**, al promover la docencia por competencias y el enfoque de aprendizaje activo, se consolida como una institución que impulsa la transformación educativa con base en evidencia pedagógica.

Este puente teórico también legitima la experiencia como un **producto académico**, al articular teoría, práctica y reflexión crítica. La práctica sistematizada deja de ser una experiencia aislada y se convierte en **conocimiento transferible**, susceptible de ser compartido con otros docentes e integrado en procesos institucionales de mejora curricular. Además, reafirma que la calidad universitaria depende de la coherencia entre el mode-

lo educativo, las prácticas pedagógicas y la evaluación por competencias (Universidad Estatal de Milagro (UNEMI), 2022).

En conjunto, los cinco puentes conforman un proceso de maduración académica y profesional. La identificación de competencias, la verificación de resultados de aprendizaje, la producción de evidencias y la reflexión sobre la alineación curricular demuestran que la docencia en Ingeniería de Software no solo forma técnicos, sino **profesionales éticos, críticos y reflexivos**.

Este cierre proyecta el trabajo hacia el **Módulo 4**, donde el análisis de resultados permitirá medir el impacto de las estrategias implementadas y formular propuestas de mejora basadas en la evidencia generada, consolidando el ciclo de reflexión–acción–mejora continua en la práctica docente universitaria.

1.4. Ecosistema estratégico (estrategias y relaciones)

1.4.1. Puente de Transición hacia el Análisis Estratégico

Hasta este punto, se ha logrado mostrar la coherencia curricular de la experiencia, evidenciando la conexión entre las competencias del perfil, los resultados de aprendizaje y las actividades implementadas. Sin embargo, comprender una experiencia educativa en su totalidad requiere ir más allá del plano curricular: es necesario explorar las estrategias que la hicieron posible en la práctica. El propósito de esta sección es precisamente establecer la bisagra entre la planeación curricular y la acción estratégica, explicando cómo las decisiones metodológicas se tradujeron en procesos reales que garantizaron el logro de los aprendizajes esperados.

La mirada estratégica permite revelar el entramado operativo que sostuvo la experiencia, evidenciando las dinámicas de acción que convirtieron las intenciones formativas en logros observables. Más que describir qué se enseñó, se trata de explicar **cómo** se desplegó un conjunto de estrategias coherentes, secuenciadas y adaptativas que respondieron a los retos propios del contexto educativo de la UNEMI. En este sentido, la reflexión se orienta hacia el análisis del diseño didáctico como un sistema vivo, compuesto por estrategias núcleo, de soporte y de contingencia, que interactúan de forma articulada para sostener el proceso de enseñanza-aprendizaje. Estrategias Núcleo

Las **estrategias núcleo** constituyen el corazón de la experiencia, aquellas decisiones que definieron la dirección pedagógica y el alcance de los aprendizajes. En el contexto

de la UNEMI, estas estrategias se centraron en la implementación de metodologías activas y reflexivas que favorecieron la construcción significativa del conocimiento en los estudiantes de Ingeniería de Software. Entre las acciones más decisivas se destacan tres:

El análisis de casos con requerimientos cambiantes, que promovió la adaptación de los estudiantes a contextos de incertidumbre, fortaleciendo competencias de resolución de problemas y pensamiento crítico.

La elaboración progresiva de documentos de diseño, en la que los estudiantes pasaron por distintas iteraciones del producto, integrando retroalimentación y demostrando comprensión del ciclo de vida del software.

Las sesiones de retroalimentación entre pares y reflexión guiada, que propiciaron la metacognición y el aprendizaje colaborativo, permitiendo evidenciar la comprensión conceptual y la capacidad argumentativa.

Estas estrategias se desarrollaron siguiendo secuencias planificadas, basadas en el modelo de Alineamiento Constructivo de Biggs (2003), que asegura coherencia entre los objetivos de aprendizaje, las actividades y las evaluaciones. Cada acción fue diseñada para generar evidencias verificables: informes técnicos, rúbricas de coevaluación, y registros de autoevaluación. Gracias a esta ingeniería didáctica intencional, los resultados obtenidos no fueron fortuitos, sino el reflejo de un proceso estructurado y consciente, que convirtió la práctica docente en una experiencia formativa de alto impacto.

1.4.2. Estrategias de Soporte

Ninguna estrategia principal puede sostenerse sin un andamiaje que la respalde. Las **estrategias de soporte** fueron los mecanismos que garantizaron la viabilidad y continuidad de la innovación. Estas incluyeron tanto recursos institucionales como herramientas metodológicas y logísticas. Desde el apoyo del entorno virtual de aprendizaje hasta la disposición de tiempo curricular para la reflexión y la documentación de procesos, cada elemento de soporte contribuyó a que las estrategias núcleo pudieran implementarse de manera efectiva.

Entre las acciones más significativas se destacan la **utilización de la plataforma Moodle de la UNEMI** para centralizar recursos y tareas, la **creación de rúbricas y guías estructuradas** para orientar la evaluación, y el **acompañamiento formativo entre docentes**, que permitió afinar criterios y compartir buenas prácticas. Este conjunto de apoyos conformó una infraestructura pedagógica que fortaleció la coherencia del pro-

ceso, asegurando que la experiencia no dependiera exclusivamente de la improvisación individual, sino de una red institucional que la sostuvo.

Las estrategias de soporte no solo ofrecieron estabilidad, sino que potenciaron la efectividad de las estrategias núcleo. Por ejemplo, el uso de guías estandarizadas permitió a los estudiantes reconocer con claridad las expectativas de desempeño, mientras que el trabajo colaborativo docente garantizó la consistencia metodológica entre grupos y niveles. En consecuencia, el soporte se consolidó como una dimensión indispensable de la sostenibilidad pedagógica.

1.4.3. Estrategias de Contingencia

Toda práctica innovadora se enfrenta a imprevistos que ponen a prueba su solidez. Las **estrategias de contingencia** fueron aquellas implementadas para responder de forma flexible ante obstáculos sin desvirtuar los objetivos formativos. En esta experiencia, los principales desafíos se relacionaron con la gestión del tiempo, la resistencia inicial de algunos estudiantes al cambio metodológico y ciertos inconvenientes técnicos en el uso de plataformas digitales.

Ante estas situaciones, se aplicaron medidas adaptativas, como **ajustes en las secuencias temporales**, la **incorporación de tutorías personalizadas**, y la **habilitación de canales de comunicación alternativos** mediante foros y videollamadas. Estas decisiones permitieron mantener el ritmo del aprendizaje, mostrando que la gestión de la contingencia forma parte de la competencia docente contemporánea.

La importancia de estas estrategias radica en que demuestran la resiliencia pedagógica del proceso. Más que “soluciones de emergencia”, las contingencias se convirtieron en oportunidades de mejora y aprendizaje tanto para el docente como para los estudiantes, evidenciando la capacidad de adaptación ante contextos dinámicos. Integración del Ecosistema Estratégico

Una vez analizadas las tres categorías de estrategias, se puede visualizar la experiencia como un **ecosistema estratégico**. Este ecosistema se organiza en una arquitectura en la que las estrategias núcleo conforman el eje central, las estrategias de soporte actúan como cimientos que brindan estabilidad, y las de contingencia funcionan como mecanismos de ajuste que aseguran la continuidad ante imprevistos.

Esta integración revela la coherencia interna del proceso educativo: las estrategias no actuaron de forma aislada, sino interdependiente. Su interacción generó un sistema equi-

librado que combinó planificación, apoyo y flexibilidad. De este modo, el ecosistema estratégico puede representarse como una red viva de acciones que sostienen la experiencia desde la intención curricular hasta su implementación práctica.

Esta visión sistémica permite entender la innovación educativa como un proceso complejo y articulado, donde cada decisión pedagógica tiene sentido dentro del conjunto. Así, la experiencia de UNEMI se consolida como un modelo de práctica reflexiva que integra teoría, acción y evaluación continua.

1.4.4. Integración Estratégica y Proyección hacia la Evaluación

El recorrido desarrollado permite concluir que las estrategias aplicadas (núcleo, soporte y contingencia) no solo garantizan la coherencia operativa del proceso, sino que también aseguran su pertinencia curricular. Gracias a este ecosistema estratégico, fue posible **alcanzar competencias clave** del perfil de egreso, tales como la capacidad de análisis crítico, la resolución de problemas en contextos reales, el trabajo colaborativo y la gestión integral de proyectos tecnológicos.

Las evidencias obtenidas (documentos de diseño, rúbricas de evaluación, informes de reflexión y proyectos funcionales) respaldan el logro de estos resultados, demostrando que la experiencia docente no solo fortaleció aprendizajes técnicos, sino también dimensiones éticas, comunicativas y reflexivas. El mensaje que debe quedar al lector es que la experiencia representa una propuesta pedagógica válida, transferible y coherente con los fines formativos de la UNEMI.

El ecosistema estratégico dejó como principal aporte la confirmación de que la enseñanza basada en competencias requiere una estructura dinámica, en la que cada tipo de estrategia cumple un rol complementario. Las relaciones entre ellas reflejan una pedagogía de la interdependencia, donde la planificación anticipa, el soporte sostiene y la contingencia adapta. Esta integración fortalece la profesionalización docente y demuestra que la gestión pedagógica puede abordarse como un proceso de ingeniería educativa.

Finalmente, esta visión estratégica prepara el camino hacia el **Módulo 5**, orientado a la evaluación integral de los resultados obtenidos. Desde la perspectiva de la UNEMI, esta transición implica pasar de la descripción de la práctica a su valoración sistemática, midiendo su impacto en los aprendizajes y en la calidad del proceso formativo. De este modo, la experiencia documentada se erige no solo como evidencia de coherencia curri-

Figura 1.1: Modelo de articulación entre ecosistema estratégico, tipos de estrategias y competencias curriculares



Fuente: elaboración propia.

cular, sino también como un referente de mejora continua en la educación universitaria contemporánea.

1.5. Evaluación, indicadores, instrumentos, análisis

1.5.1. Puente de Evaluación: Instrumentos, Indicadores y Criterios de Validez

El proceso de evaluación constituyó una fase clave para comprobar, de manera sistemática, el logro de las competencias curriculares derivadas de la experiencia pedagógica. Evaluar implicó no solo medir resultados, sino también garantizar la transparencia y la credibilidad del proceso mediante instrumentos adecuados y criterios de validez claramente establecidos. En este sentido, se aplicaron cuatro herramientas complementarias: la **rúbrica analítica de desempeño**, el **cuestionario de percepción estudiantil**, la **entrevista semiestructurada** y el **registro de observación sistemática**. Cada instrumento respondió a una dimensión específica del aprendizaje y permitió generar evidencias diversas y trianguladas, fortaleciendo así la confiabilidad del proceso evaluativo.

1.5.2. Rúbrica analítica de desempeño

La rúbrica analítica fue el instrumento central de evaluación, orientado a valorar el nivel de logro de las competencias profesionales, comunicativas y reflexivas de los participantes. Esta herramienta desglosó los criterios de desempeño en indicadores observables, tales como **coherencia argumentativa, aplicación del conocimiento teórico, resolución creativa de problemas y trabajo colaborativo**. Se aplicó durante la revisión de productos académicos y proyectos de aplicación, empleando una escala de cuatro niveles: principiante, básico, competente y destacado.

Cada nivel se acompañó de descriptores cualitativos que facilitaron una interpretación homogénea y justa de los resultados. Este instrumento generó evidencias tanto cuantitativas (puntajes) como cualitativas (comentarios analíticos del evaluador). Su pertinencia radicó en la posibilidad de medir el desempeño real de los estudiantes respecto a los resultados de aprendizaje esperados, asegurando una **evaluación auténtica** alineada con las competencias curriculares (Andrade, 2019).

1.5.3. Cuestionario de percepción estudiantil

El cuestionario de percepción se aplicó al finalizar la experiencia con el propósito de recoger la valoración de los estudiantes sobre la pertinencia de las estrategias pedagógicas y la efectividad del proceso formativo. Este instrumento incluyó **ítems cerrados tipo Likert (de 1 a 5)** para medir aspectos como la claridad de las consignas, la utilidad de las actividades, la retroalimentación docente y la percepción de aprendizaje alcanzado. Además, incorporó **preguntas abiertas** que permitieron captar apreciaciones cualitativas.

Su aplicación fue anónima y digital, lo cual favoreció respuestas sinceras y reflexivas. Los datos obtenidos sirvieron para complementar la evaluación del desempeño mediante la voz de los participantes, fortaleciendo la validez interna del proceso. Este instrumento permitió identificar patrones de satisfacción, niveles de motivación y percepciones sobre la coherencia entre lo enseñado y lo aprendido (Gómez & Cárdenas, 2021).

1.5.4. Entrevista semiestructurada

La **entrevista semiestructurada** fue aplicada a una muestra representativa de estudiantes y docentes tutores con el objetivo de profundizar en las percepciones, aprendizajes y desafíos vividos durante la experiencia. Las preguntas abordaron temas como la aplica-

ción de las estrategias núcleo, la colaboración entre pares, la autonomía en la resolución de problemas y el impacto de la práctica en el desarrollo profesional.

Las entrevistas fueron grabadas, transcritas y analizadas mediante codificación temática, identificando categorías emergentes como “crecimiento reflexivo”, “sentido de logro” y “aprendizaje situado”. Este instrumento generó evidencias cualitativas que complementaron los datos de la rúbrica y el cuestionario, aportando una comprensión más holística del proceso educativo (Kvale & Brinkmann, 2015).

1.5.5. Registro de observación sistemática

El **registro de observación sistemática** permitió documentar, durante las sesiones prácticas y colaborativas, el comportamiento de los estudiantes en relación con indicadores como la participación activa, la interacción en grupo, la toma de decisiones y la aplicación práctica del conocimiento. El registro se estructuró en una ficha con categorías predefinidas (participación, comunicación, liderazgo, resolución de conflictos) y un espacio para observaciones libres.

Este instrumento aportó evidencias directas sobre la conducta observable en el aula y permitió verificar la coherencia entre lo que los estudiantes expresaron en los cuestionarios y lo que realmente se evidenció en la práctica. Su uso continuo facilitó el seguimiento longitudinal de la experiencia y la identificación de patrones de mejora sostenida (Cohen et al., 2018).

1.5.6. Indicadores de evaluación y criterios de validez

Los instrumentos se articularon en torno a un conjunto de **indicadores de evaluación** diseñados para medir el logro de las competencias curriculares. Entre ellos destacaron:

Coherencia conceptual: correspondencia entre la teoría y la práctica aplicada en los productos académicos.

Pertinencia profesional: grado en que las actividades reflejaron situaciones reales del campo formativo.

Calidad reflexiva: capacidad del estudiante para analizar críticamente sus decisiones y resultados.

Participación colaborativa: evidencia de trabajo en equipo, liderazgo y comunicación efectiva.

Autonomía en el aprendizaje: nivel de iniciativa y responsabilidad mostrado en las tareas.

La validez del proceso se garantizó mediante la **triangulación de fuentes** (rúbricas, encuestas, entrevistas y observaciones), la **coherencia interna** entre los instrumentos y los objetivos de aprendizaje, y la **transparencia metodológica** en la aplicación y análisis de los datos. Asimismo, se adoptaron principios de **consistencia y credibilidad**, inspirados en los criterios de rigor cualitativo propuestos por Lincoln y Guba (1985), que incluyen la **credibilidad, transferibilidad y objetividad**.

1.5.7. Justificación y cierre

La pertinencia de estos instrumentos e indicadores radicó en su complementariedad y coherencia con las competencias del perfil de egreso. Cada herramienta aportó un tipo de evidencia que, al integrarse, permitió construir una evaluación válida, confiable y contextualizada. Gracias a este diseño, la evaluación trascendió la mera calificación para convertirse en un proceso reflexivo de comprensión y mejora continua.

En síntesis, los instrumentos aplicados no solo midieron el logro de los resultados de aprendizaje, sino que posibilitaron una lectura integral de la experiencia formativa. Este enfoque consolidó la base empírica y conceptual que, en el siguiente apartado, permitirá analizar los **indicadores en acción** y las **evidencias emergentes** que reflejan los logros alcanzados en el desarrollo de las competencias curriculares.

Tabla 1.4: Instrumentos de evaluación y sus características

Instrumento de evaluación	Objetivo de aplicación	Forma y momento de aplicación	Tipo de evidencia generada	Indicadores de evaluación asociados	Criterios de validez y confiabilidad
Rúbrica analítica de desempeño	Valorar el nivel de logro de las competencias curriculares mediante criterios observables de desempeño académico y profesional.	Aplicada por el docente evaluador durante la presentación de productos académicos y proyectos aplicados (fase final de la experiencia).	Cuantitativa (puntajes) y cualitativa (comentarios analíticos).	Coherencia conceptual; Aplicación del conocimiento teórico; Resolución de problemas; Trabajo colaborativo; Calidad reflexiva	Triangulación con otros instrumentos; consistencia interna en los criterios; revisión por pares docentes para asegurar confiabilidad interevaluador.

Cuestionario de percepción estudiantil	Recoger la valoración de los estudiantes sobre la pertinencia, utilidad y coherencia de las estrategias de aprendizaje.	Aplicado en formato digital y anónimo al finalizar la experiencia.	Cuantitativa (escala Likert) y cualitativa (preguntas abiertas).	Pertinencia profesional; Motivación y satisfacción; Claridad de consignas; Retroalimentación docente	Validación de contenido mediante revisión de expertos; anonimato para evitar sesgos; análisis estadístico básico (frecuencias y medias).
Entrevista semiestructurada	Profundizar en las percepciones de docentes y estudiantes sobre el impacto de las estrategias pedagógicas y los aprendizajes logrados.	Aplicada a una muestra representativa de participantes al concluir la experiencia.	Cualitativa (transcripciones y categorías temáticas).	Calidad reflexiva; Transferencia del aprendizaje; Satisfacción con la metodología; Sentido de logro personal	Criterios de credibilidad y objetividad (Lincoln & Guba, 1985); registro y codificación sistemática; revisión cruzada de categorías.

Registro de observación sistemática	Documentar el desempeño observable de los participantes en contextos reales o simulados de práctica profesional.	Aplicado por el docente durante el desarrollo de actividades colaborativas y sesiones prácticas.	Cualitativa y descriptiva (notas de campo y categorías predefinidas).	Participación activa; Comunicación efectiva; Resolución colaborativa de tareas; Autonomía en el aprendizaje	Observación estructurada mediante fichas; triangulación con rúbrica y entrevistas; coherencia entre criterios de observación y competencias del perfil de egreso.
-------------------------------------	--	--	---	---	---

Fuente: Elaboración propia a partir del diseño metodológico del proyecto.

El conjunto de instrumentos conformó un **sistema de evaluación mixto** que integró evidencias cuantitativas (puntajes, escalas) y cualitativas (narrativas, observaciones, testimonios). Esta triangulación fortaleció la **validez interna y externa** del proceso, garantizando que los resultados obtenidos reflejaran de manera auténtica el nivel de logro de las **competencias curriculares**.

Los **criterios de validez y confiabilidad** se sustentaron en los principios de **rigurosidad metodológica** propuestos por Lincoln y Guba (1985), quienes destacan la objetividad, transferibilidad y consistencia como ejes de credibilidad en la investigación educativa. Asimismo, se cuidó la **transparencia procedimental**, documentando cada etapa de la aplicación y análisis de los instrumentos (Cohen et al., 2018).

1.6. Reflexión crítica y transferencia de la experiencia

La sistematización de la experiencia educativa desarrollada permitió no solo reconstruir un proceso didáctico innovador, sino comprender en profundidad su sentido formativo, sus alcances y sus desafíos. Mirar hacia atrás desde la evaluación y el análisis crítico implica reconocer que cada fase, desde la planificación curricular hasta la evaluación de resultados, fue parte de un ecosistema estratégico de aprendizaje. Esta mirada retrospectiva ofrece una comprensión integral de cómo las estrategias implementadas, los instrumentos aplicados y las evidencias obtenidas se articularon para transformar la práctica docente en una experiencia significativa. La reflexión final, entonces, no se limita a constatar logros, sino a comprender su trascendencia pedagógica y a vislumbrar los caminos futuros que se abren a partir de ellos.

Entre los principales aportes de la experiencia destaca la consolidación de un modelo pedagógico centrado en el desarrollo de competencias, donde la articulación entre estrategias núcleo, de soporte y de contingencia aseguró la coherencia y continuidad del proceso. Este enfoque posibilitó la participación activa de los estudiantes, fortaleció su autonomía y generó aprendizajes más duraderos y contextualizados. Asimismo, la integración de herramientas digitales y metodologías activas contribuyó a dinamizar las interacciones en el aula y a promover una cultura de colaboración. Estos resultados se reflejan en una mejora tangible de la calidad de los productos académicos y en el fortalecimiento de la identidad profesional de los participantes.

Otro logro relevante fue la capacidad institucional de responder de manera flexible ante los desafíos del contexto. En el caso de la Universidad Estatal de Milagro (UNEMI), la experiencia permitió comprobar que las innovaciones pedagógicas pueden sostenerse incluso en entornos de transformación digital o limitaciones logísticas, siempre que se construyan desde un enfoque colaborativo y reflexivo. Este hallazgo refuerza la idea de que las universidades deben concebir la docencia como un espacio de experimentación permanente, donde las prácticas se revisan, ajustan y mejoran a partir de la evidencia y la reflexión colectiva.

Sin embargo, el proceso también enfrentó tensiones significativas. Una de ellas estuvo vinculada a la resistencia inicial de algunos actores al cambio metodológico, especialmente en lo referente a la adopción de estrategias activas y el uso de tecnología educativa. Estas resistencias, aunque naturales, pusieron a prueba la capacidad de liderazgo pedagógico y la gestión emocional del equipo docente. Otra tensión estuvo asociada a la carga de

trabajo y a la necesidad de compatibilizar la innovación con las exigencias institucionales y administrativas, lo que evidenció la importancia de contar con estructuras de apoyo que favorezcan la sostenibilidad de las propuestas.

En el plano metodológico, también se presentaron limitaciones vinculadas a la aplicación de los instrumentos de evaluación. En algunos momentos, las rúbricas o los cuestionarios no lograron captar con precisión la complejidad de los procesos formativos observados, lo que llevó a ajustes y reinterpretaciones. Esta situación, lejos de debilitar la experiencia, permitió aprender sobre la necesidad de diseñar herramientas más flexibles y sensibles a la diversidad de contextos y estilos de aprendizaje. En este sentido, la reflexión crítica se convirtió en un motor de mejora continua.

Desde el punto de vista personal y profesional, el proceso de sistematización constituyó una oportunidad de crecimiento docente. Permitir que la práctica se vuelva objeto de análisis transformó la forma de comprender el rol educativo: ya no como transmisor de conocimiento, sino como mediador de procesos de aprendizaje. La reflexión sobre la propia práctica, apoyada en evidencias, propició un ejercicio de autoconocimiento profesional y consolidó una ética pedagógica basada en la coherencia, la empatía y la responsabilidad social. Además, fortaleció la capacidad de documentar, comunicar y compartir experiencias, lo que incrementa el capital pedagógico institucional.

A nivel colectivo, la experiencia generó aprendizajes valiosos sobre trabajo en equipo, construcción de consensos y gestión compartida del conocimiento. La interacción entre docentes, estudiantes y autoridades permitió construir una visión común sobre la enseñanza y el aprendizaje, fortaleciendo la cultura institucional de innovación. Este aprendizaje colaborativo constituye uno de los legados más potentes de la experiencia, ya que trasciende el aula y se proyecta hacia la gestión académica y la mejora continua de la carrera.

En el ámbito institucional, la sistematización mostró que las innovaciones educativas solo son sostenibles cuando se integran a las políticas y estructuras organizativas. UNEMI demostró que la reflexión pedagógica puede y debe formar parte de los procesos institucionales de evaluación y acreditación, favoreciendo una cultura de evidencia y mejora. Esta conexión entre la práctica docente y la política universitaria abre un horizonte de transferencia que puede inspirar a otras unidades académicas a documentar y reflexionar sus propias experiencias formativas.

La sistematización permitió comprender que cada acción pedagógica tuvo un impacto diferenciado según los contextos y actores involucrados. Este reconocimiento llevó a valorar la flexibilidad del proceso como una fortaleza, ya que permitió ajustar estrategias

en función de las necesidades emergentes del estudiantado. Así, la práctica docente se consolidó como un espacio de aprendizaje adaptativo y reflexivo, donde el error y la retroalimentación se convirtieron en oportunidades para innovar. Esta perspectiva flexible reafirma la idea de que la educación de calidad requiere apertura al cambio y disposición constante para el aprendizaje continuo.

Otro aspecto relevante fue el fortalecimiento del sentido de comunidad académica. La experiencia promovió la colaboración entre docentes y estudiantes, generando redes de apoyo que trascendieron el aula. Este enfoque cooperativo favoreció el intercambio de saberes y experiencias, consolidando una cultura de trabajo interdisciplinario. A nivel institucional, este proceso contribuyó a reforzar la identidad colectiva, evidenciando que la innovación educativa no depende únicamente de la creatividad individual, sino de la capacidad de construir significados compartidos en torno al aprendizaje.

Desde una mirada crítica, también se reconoce que la implementación del ecosistema estratégico implicó enfrentar resistencias y limitaciones estructurales. La gestión del tiempo, la carga laboral y las restricciones tecnológicas fueron factores que tensionaron el proceso, obligando a replantear algunas metas iniciales. Sin embargo, estas dificultades se convirtieron en insumos para la mejora, al permitir identificar áreas donde la institución puede fortalecer su infraestructura y sus políticas de acompañamiento docente. En este sentido, los desafíos enfrentados no debilitaron la experiencia, sino que le otorgaron profundidad y realismo.

Finalmente, la experiencia deja aprendizajes transferibles a otros contextos educativos. Las estrategias utilizadas pueden adaptarse a distintas áreas disciplinarias, siempre que se mantenga el principio de coherencia entre competencias, metodologías y evaluación. Más aún, el enfoque reflexivo adoptado promueve una visión del docente como investigador de su práctica, capaz de generar conocimiento pedagógico desde la acción. Este cambio de paradigma de *enseñar a aprender enseñando* es uno de los aportes más valiosos para el fortalecimiento de la educación superior contemporánea.

En síntesis, la reflexión crítica final reafirma que sistematizar una experiencia no es solo reconstruir un proceso, sino resignificarlo. A través de este recorrido, se comprendió que la innovación educativa implica gestionar tensiones, valorar los logros y reconocer los límites. Pero, sobre todo, implica construir conocimiento colectivo sobre cómo enseñar mejor. Este capítulo, al integrar evidencias, estrategias y reflexiones, deja abierta la invitación a seguir aprendiendo desde la práctica, proyectando sus enseñanzas hacia nue-

vas experiencias docentes que fortalezcan la calidad y la pertinencia de la educación en UNEMI y más allá.

Bibliografía

- ABET. (2020). *Criteria for Accrediting Computing Programs*. Accreditation Board for Engineering; Technology.
- Andrade, H. (2019). Using rubrics to promote thinking and learning. *Educational Leadership*, 57(5), 13-18.
- Biggs, J. (2003). *Teaching for Quality Learning at University*. Open University Press.
- Biggs, J., & Tang, C. (2011). *Teaching for Quality Learning at University* (4.^a ed.). McGraw-Hill Education.
- Bolívar, A. (2016). *La investigación biográfico-narrativa en educación*. Editorial Síntesis.
- Carlino, P. (2005). *Escribir, leer y aprender en la universidad: Una introducción a la alfabetización académica*. Fondo de Cultura Económica.
- Cohen, L., Manion, L., & Morrison, K. (2018). *Research Methods in Education* (8.^a ed.). Routledge. <https://doi.org/10.4324/9781315456539>
- Day, C. (2006). *Pasión por enseñar: La identidad personal y profesional del docente y sus valores*. Narcea.
- Díaz-Barriga, Á. (2014). *Curriculum, enseñanza y aprendizaje: Una mirada constructivista*. Trillas.
- Elliott, J. (1993). *El cambio educativo desde la investigación-acción*. Morata.
- Flick, U. (2014a). *Introducción a la investigación cualitativa* (5.^a ed.). Morata.
- Flick, U. (2014b). *An Introduction to Qualitative Research* (5.^a ed.). SAGE Publications.
- Freire, P. (1996). *Pedagogía de la autonomía: Saberes necesarios para la práctica educativa*. Siglo XXI.
- Fullan, M., & Hargreaves, A. (2012). *Professional Capital: Transforming Teaching in Every School*. Teachers College Press.
- Gómez, M., & Cárdenas, S. (2021). Percepción estudiantil sobre estrategias pedagógicas y aprendizaje significativo en educación superior. *Revista de Estudios Educativos Contemporáneos*, 5(2), 45-63.
- Hyland, K. (2009). *Academic Discourse: English in a Global Context*. Continuum.
- Jara, O. (2018a). *La sistematización de experiencias: Aprendizajes y desafíos para la educación popular*. Alforja.
- Jara, O. (2018b). *La sistematización de experiencias: Práctica y teoría para otros mundos posibles*. Alforja.

- Kvale, S., & Brinkmann, S. (2015). *InterViews: Learning the Craft of Qualitative Research Interviewing* (3.^a ed.). SAGE Publications.
- Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic Inquiry*. SAGE Publications.
- Schön, D. (1992). *La formación de profesionales reflexivos*. Paidós.
- Shulman, L. (2005). Signature Pedagogies in the Professions. *Daedalus*, 134(3), 52-59.
- Stake, R. (1995). *The Art of Case Study Research*. SAGE Publications.
- Stenhouse, L. (1987). *La investigación como base de la enseñanza*. Morata.
- Universidad Estatal de Milagro (UNEMI). (2022). *Modelo Educativo Institucional*. UNE-MI.
- Yin, R. K. (2014). *Case Study Research: Design and Methods* (5.^a ed.). SAGE Publications.
- Zabalza, M. A. (2011). *Competencias docentes del profesorado universitario: Calidad y desarrollo profesional*. Narcea.

2

Del problema a la solución: estrategias para enseñar análisis y diseño de software en primer nivel

Delia Isabel Carrión León ²

El capítulo sistematiza una experiencia pedagógica en Ingeniería de Software (UNEMI) para enseñar análisis y diseño en primer nivel. Parte de un doble problema: brecha digital y debilidades en pensamiento lógico-computacional. Con metodologías activas (aprendizaje basado en problemas y aula invertida) se priorizó comprender el problema antes de programar, fortaleciendo análisis, modelado y razonamiento computacional. La propuesta integra pensamiento computacional, aprendizaje activo, modelado de sistemas y práctica colaborativa-reflexiva, articulando teoría y práctica en un entorno inclusivo. Los resultados muestran mejoras conceptuales, mayor uso de diagramas UML y competencias analíticas transferibles, reduciendo deserción y elevando motivación estudiantil.

²Universidad Estatal de Milagro, dcarrionl@unemi.edu.ec.

Índice

2.1. Apertura contextual y problematización de la experiencia docente .	42
2.1.1. El problema formativo	42
2.1.2. Propósito de la sistematización	44
2.1.3. Criterios de valor	45
2.1.4. Delimitación del objeto de estudio	46
2.2. Fundamentación conceptual y operativa: del pensamiento computacional al aprendizaje activo	47
2.2.1. Transición hacia la fundamentación conceptual y operativa . . .	47
2.2.2. Aprendizaje activo: del estudiante receptor al estudiante protagonista	49
2.2.3. Práctica colaborativa-reflexiva: aprender con otros para construir significado	50
2.2.4. Formulación de dimensiones	50
2.2.5. Dimensión colaborativa-reflexiva: aprender con otros para construir conocimiento	51
2.2.6. Construcción de indicadores	52
2.2.7. Dimensión pedagógica-didáctica: enseñar a analizar antes de programar	53
2.2.8. Dimensión colaborativa-reflexiva: aprender con otros para construir conocimiento	54
2.2.9. Cierre proyectivo	55
2.2.10. Fuentes y métodos de verificación	55
2.2.11. Fuentes de verificación	56
2.2.12. Métodos de verificación	56
2.2.13. Ejemplo de aplicación	57
2.2.14. Cierre proyectivo	57
2.2.15. Justificación teórica del conjunto	57
2.2.16. Integración	60

2.3. Vínculo curricular y resultados de aprendizaje en la enseñanza del análisis y diseño de software	60
2.3.1. Transición al vínculo curricular	60
2.3.2. Identificación de competencias del perfil	61
2.3.3. Resultados de aprendizaje vinculados	64
2.3.4. Actividades y evidencias	66
2.3.5. Reflexión sobre la alineación curricular	68
2.3.6. Integración curricular del Módulo 3	70
2.4. Del problema a la solución: estrategias para enseñar análisis y diseño de software en primer nivel	73
2.4.1. Transición hacia la operacionalización estratégica	73
2.4.2. Estrategias núcleo en acción	74
2.4.3. Estrategias de soporte aplicadas	76
2.4.4. Estrategias de contingencia desplegadas	79
2.4.5. Arquitectura del ecosistema estratégico	81
2.4.6. Integración: Justificación de la validez curricular del ecosistema estratégico	84
2.5. Evaluación e indicadores de logro: instrumentos, validez y análisis de evidencias	86
2.5.1. Transición hacia la evaluación	86
2.5.2. Instrumentos de evaluación aplicados	87
2.5.3. Indicadores de evaluación y criterios de validez	89
2.5.4. Indicadores aplicados	90
2.5.5. Criterios de validez y confiabilidad	91
2.5.6. Análisis preliminar de evidencias	93
2.5.7. Tipos de evidencias y organización del análisis	93
2.5.8. Hallazgos preliminares	94
2.5.9. Interpretación de patrones emergentes	94
2.5.10. Síntesis y proyección	95
2.5.11. Reflexión sobre validez, sesgos y factibilidad	95

2.5.12. Validez del proceso evaluativo	95
2.5.13. Sesgos identificados y estrategias de mitigación	96
2.5.14. Factibilidad y aprendizajes derivados	97
2.5.15. Síntesis y proyección	98
2.5.16. Integración: Síntesis de la evaluación	98
2.6. Del problema a la solución: estrategias para enseñar análisis y diseño de software en primer nivel	100
2.6.1. Transición hacia la reflexión final	100
2.6.2. Reflexión crítica sobre la experiencia	101
2.6.3. Aportes de la experiencia	101
2.6.4. Tensiones y resistencias encontradas	102
2.6.5. Aprendizajes personales, colectivos e institucionales	103
2.6.6. Síntesis reflexiva y proyección	104
2.6.7. Integración final: Reflexión y transferencia	104
2.6.8. Aportes y aprendizajes globales	105
2.6.9. Tensiones, desafíos y aprendizajes emergentes	106
2.6.10. Proyección y transferencia	106
2.6.11. Cierre del capítulo	107

2.1. Apertura contextual y problematización de la experiencia docente

En continuidad con la reflexión iniciada en el capítulo anterior sobre la enseñanza de los fundamentos de la ingeniería de software, esta experiencia se centra en una fase complementaria del proceso formativo: el desarrollo del pensamiento analítico y del diseño estructurado de soluciones. El trabajo se enmarca en la carrera de Ingeniería de Software de la Universidad Estatal de Milagro (UNEMI), y se orienta a los estudiantes que cursan el primer nivel, quienes, tras su acercamiento inicial a los conceptos generales de la disciplina, se enfrentan ahora al desafío de aplicar el análisis y la modelización en la resolución de problemas.

El grupo participante estuvo conformado por jóvenes de entre 18 y 20 años, provenientes de distintas zonas urbanas y rurales, con trayectorias educativas diversas. Aunque comparten el interés por la tecnología, muchos presentan limitaciones en el desarrollo del razonamiento lógico y del pensamiento computacional, competencias necesarias para avanzar en la carrera. Este punto de partida exige un acompañamiento docente que articule teoría y práctica, así como estrategias inclusivas que consideren las desigualdades en el acceso a recursos digitales.

Durante las primeras clases, una actividad introductoria reveló un obstáculo recurrente: al enfrentarse con un problema sencillo de análisis, varios estudiantes admitieron no haber trabajado nunca con diagramas de flujo ni representaciones gráficas de procesos. Este hecho, lejos de ser una dificultad aislada, permitió identificar un patrón: la tendencia a buscar soluciones inmediatas sin comprender plenamente el problema. A partir de allí, se propuso una secuencia de actividades que priorizaran la comprensión, el análisis y la modelización como pasos previos a la programación, fortaleciendo el vínculo entre teoría y práctica.

2.1.1. El problema formativo

El principal problema identificado en la enseñanza de la Ingeniería de Software en el primer nivel radica en dos dimensiones interrelacionadas: la persistente brecha digital y las dificultades en el desarrollo del pensamiento lógico-computacional. Estos factores limitan la capacidad de los estudiantes para comprender el análisis y diseño de sistemas antes de llegar a la etapa de programación. Como señala Wing (2006), el pensamiento

computacional constituye una competencia esencial en la educación en ciencias de la computación, y su ausencia inicial repercute en la manera en que los estudiantes enfrentan los retos de la carrera.

La relevancia de este problema es evidente: cuando los estudiantes no logran comprender la importancia de analizar los problemas antes de proponer soluciones, se genera un sesgo hacia la programación como única meta, dejando de lado procesos fundamentales como la identificación de requerimientos y el modelado. Grover y Pea (2013) destacan que el pensamiento computacional no solo implica programar, sino también descomponer problemas, reconocer patrones y abstraer soluciones. Sin estas bases, el avance en asignaturas posteriores se ve afectado, y la visión de la ingeniería como una disciplina sistemática se debilita.

Las consecuencias de no atender esta problemática son significativas. Si los estudiantes mantienen la idea de que programar equivale a “hacer software”, se enfrentarán a dificultades mayores en asignaturas avanzadas que exigen razonamiento formal, modelado de sistemas y arquitectura de software. Este desfase se traduce en desmotivación y, en muchos casos, en riesgo de abandono de la carrera. Según Tinto (2017), la falta de dominio de competencias básicas y el desajuste entre expectativas y realidades formativas son factores críticos en la deserción estudiantil en educación superior. Resolver este problema no es opcional: constituye una condición necesaria para asegurar la continuidad y el éxito académico.

Una evidencia práctica de este desafío se observó en las primeras actividades de ingeniería de requerimientos y modelado, donde se invitó a los estudiantes a construir un árbol de problemas. Aunque la consigna era identificar causas y efectos, la mayoría se centró en proponer soluciones inmediatas, omitiendo el análisis del problema. Este comportamiento confirma lo señalado por Pressman y Maxim (2020), quienes advierten que el desarrollo de software de calidad requiere un entendimiento profundo de las necesidades antes de la codificación. Incluso docentes en formación tienden a reproducir este enfoque limitado, privilegiando la ejecución sobre la reflexión.

En síntesis, el problema formativo se concreta en la insuficiente preparación de los estudiantes para enfrentar los primeros niveles de la carrera, debido a la brecha digital y a las dificultades en el desarrollo del pensamiento lógico y computacional. Como señalan Van Dijk y Hacker (2018), la brecha digital no se limita al acceso a recursos, sino que incluye competencias de uso y comprensión crítica. Atender este problema constituye, por tanto, el punto de partida para el propósito de esta sistematización.

2.1.2. Propósito de la sistematización

El propósito de esta sistematización es visibilizar la importancia del análisis y diseño como ejes fundamentales en la formación de los futuros ingenieros de software. En los primeros niveles de la carrera, muchos estudiantes asumen que programar es sinónimo de “hacer software”, sin advertir que la comprensión del problema, la identificación de requerimientos y el modelado constituyen la base de cualquier desarrollo tecnológico de calidad. Por ello, esta experiencia busca demostrar que el fortalecimiento del pensamiento lógico y computacional, desde actividades iniciales de análisis, no solo mejora el desempeño académico inmediato, sino que prepara el terreno para afrontar con éxito las asignaturas posteriores de la malla curricular.

Este propósito surge de la necesidad de transformar una práctica docente que, en muchos casos, privilegia la solución inmediata sobre la comprensión profunda del problema. La sistematización permite convertir lo vivido en un relato académico capaz de aportar reflexiones y propuestas a la comunidad universitaria. Como plantea Jara (2018a), toda sistematización debe responder a una intencionalidad clara que oriente la reconstrucción de la experiencia y dé sentido a su difusión. En este caso, la intención es consolidar un recurso que combine teoría y práctica, integrando ejercicios, casos y diagramas que muestren a los estudiantes la relevancia de analizar antes de programar.

La relevancia de este propósito es doble. En primer lugar, responde a la necesidad formativa de los estudiantes de Ingeniería de Software, quienes enfrentan limitaciones derivadas de la brecha digital y de vacíos en el desarrollo del razonamiento lógico. En segundo lugar, ofrece a los docentes un modelo replicable de enseñanza que puede integrarse en diversas asignaturas iniciales de la carrera. De acuerdo con Carlino (2005), narrar las experiencias docentes desde un enfoque académico no solo documenta buenas prácticas, sino que también fortalece la identidad profesional y abre espacios de innovación pedagógica. La proyección hacia el lector se concreta en tres aportes específicos:

1. La posibilidad de acceder a un conjunto de actividades prácticas que ejercitan el análisis de problemas, la construcción de árboles de causas y efectos, y la elaboración de diagramas básicos en UML.
2. La presentación de casos de estudio reales y contextualizados que facilitan el desarrollo de competencias analíticas y críticas.

3. Un modelo didáctico adaptable por otros docentes de la carrera o de programas afines, contribuyendo a reducir la desmotivación y el riesgo de deserción.

En síntesis, el propósito de esta sistematización es transformar una práctica pedagógica situada en la enseñanza del análisis y diseño de software en un recurso académico con proyección institucional y disciplinar (Hyland, 2009).

2.1.3. Criterios de valor

El valor de esta experiencia radica en la posibilidad de replantear cómo se introducen los fundamentos de la ingeniería de software en el primer nivel universitario. En lugar de reproducir un enfoque centrado únicamente en la programación, se priorizó la enseñanza del análisis y diseño como ejes esenciales para comprender la disciplina. Esta perspectiva, poco común en los inicios de la formación, permite reconocer que el desarrollo de software comienza con la comprensión del problema y la identificación rigurosa de requerimientos, y no con la escritura inmediata de código.

La innovación de esta práctica se sustenta en la integración de metodologías activas —como el aprendizaje basado en problemas y el aula invertida— aplicadas a casos reales. Estas estrategias aportaron dinamismo a las clases y permitieron que los estudiantes reconocieran la importancia de analizar las causas y efectos de un problema antes de proponer soluciones técnicas. Como señala Elliott (1993), la investigación-acción educativa adquiere valor cuando genera conocimiento situado que responde a problemas concretos de la práctica docente.

El impacto de la experiencia se refleja en los beneficios observados en estudiantes y docentes. Los jóvenes comenzaron a comprender que la ingeniería de software no se limita al acto de programar, sino que implica un proceso ordenado de análisis, diseño y construcción de soluciones. Al trabajar con árboles de problemas y diagramas de casos de uso, muchos lograron superar su resistencia inicial y reconocer el valor de estas herramientas. Para los docentes, la experiencia abrió un espacio de reflexión sobre la necesidad de fortalecer el pensamiento lógico y computacional desde el inicio de la carrera. Schön (1992) enfatiza que el profesional reflexivo encuentra en la práctica una oportunidad para revisar y mejorar sus estrategias de enseñanza, lo que aquí se tradujo en una transformación en la forma de concebir la introducción a la disciplina.

En cuanto a su transferibilidad, la propuesta ofrece un modelo replicable que puede adaptarse a otros contextos académicos. El uso de ejercicios prácticos, casos de estudio y

diagramas UML constituye un recurso aplicable a diferentes programas de formación tecnológica, especialmente en aquellos que enfrentan limitaciones similares —como la brecha digital o la falta de preparación en lógica matemática—. Como plantea Jara (2018a), la sistematización convierte experiencias particulares en aprendizajes colectivos que enriquecen a la comunidad académica.

2.1.4. Delimitación del objeto de estudio

El objeto de estudio de esta sistematización es el proceso de enseñanza-aprendizaje del análisis y diseño de software en el primer nivel de la carrera de Ingeniería de Software de la UNEMI. El foco se sitúa en cómo enfrentar dos desafíos principales: la brecha digital que afecta a los estudiantes y las dificultades en el desarrollo del pensamiento lógico y computacional. Este recorte es intencional, pues busca atender el núcleo formativo que condiciona la continuidad y el éxito en los niveles posteriores de la carrera.

La elección de este objeto de estudio se justifica en la necesidad de visibilizar la relevancia del análisis y diseño en la formación inicial de los futuros ingenieros. En esta etapa, los estudiantes definen su relación con la carrera y enfrentan los primeros obstáculos que pueden determinar su permanencia o deserción. Sistematizar esta experiencia permite reconstruir la práctica docente, reflexionar sobre sus aciertos y limitaciones, y generar un modelo de enseñanza que integre metodologías activas, recursos digitales y ejercicios prácticos de análisis de problemas. Como señala Jara (2018a), delimitar una experiencia educativa es una forma de darle sentido, orientando su narrativa hacia aprendizajes que trascienden lo individual y se convierten en aporte colectivo.

El alcance contextual se circunscribe al semestre académico 2025-I, en modalidad presencial, en las aulas de la UNEMI. Los actores principales son los estudiantes de primer nivel, con edades entre 18 y 20 años, y los docentes que acompañan el proceso. Las evidencias centrales son los ejercicios de aula (árboles de problemas, ingeniería de requerimientos, diagramas UML), observaciones y reflexiones docentes.

En cuanto a lo que queda fuera, no se incluyen las asignaturas de niveles superiores ni los proyectos integradores avanzados, ya que el objetivo no es evaluar toda la trayectoria académica, sino concentrarse en la etapa inicial. Tampoco se abordan políticas educativas nacionales ni comparaciones externas, puesto que el interés está en la práctica situada en la UNEMI. Este recorte garantiza la coherencia del análisis y evita dispersar la reflexión hacia aspectos que exceden el propósito del capítulo.

Finalmente, el valor de esta delimitación reside en centrar la reflexión en un escenario concreto y manejable, asegurando un hilo conductor claro para el desarrollo del capítulo. Como advierte Flick (2014a), toda investigación o sistematización requiere acotar su objeto de estudio para producir conocimiento comunicable y con validez contextual. En este caso, la delimitación asegura que la narrativa mantenga coherencia y que las conclusiones aporten insumos útiles tanto a estudiantes como a docentes e instituciones.

2.2. Fundamentación conceptual y operativa: del pensamiento computacional al aprendizaje activo

2.2.1. Transición hacia la fundamentación conceptual y operativa

Luego de haber contextualizado el problema formativo y delimitado el objeto de estudio en la primera parte de este capítulo, esta segunda entrega profundiza en los fundamentos conceptuales y metodológicos que sustentan la experiencia sistematizada.

La primera parte de este capítulo permitió reconstruir la experiencia docente desde una mirada narrativa, situando el contexto institucional, el problema formativo y el propósito de la práctica pedagógica desarrollada en el primer nivel de la carrera de Ingeniería de Software de la Universidad Estatal de Milagro (UNEMI). A través de ese recorrido se identificaron los desafíos asociados a la brecha digital y a las dificultades en el desarrollo del pensamiento lógico-computacional, así como la necesidad de fortalecer la comprensión del análisis y diseño de software como fase previa a la programación. Este proceso narrativo sentó las bases para reconocer los aprendizajes y aportes de la experiencia, otorgándole sentido y coherencia dentro de la formación de futuros ingenieros.

Sin embargo, lo narrado hasta aquí requiere ahora ser sustentado teóricamente. Para dotar de mayor rigor académico y validez a la experiencia, se hace necesario integrar los marcos conceptuales que explican sus fundamentos pedagógicos, cognitivos y tecnológicos. En este nuevo apartado se realizará esa fundamentación, articulando las nociones centrales con la literatura especializada. De este modo, el texto transita del relato vivencial a la argumentación académica, mostrando cómo las decisiones didácticas se apoyan en teorías del pensamiento computacional, del aprendizaje activo y del modelado de sistemas.

A partir de esta transición, se presentarán los conceptos estructurantes que organizan la experiencia y orientan su análisis: el pensamiento computacional como competencia clave en la formación de ingenieros de software, el aprendizaje activo como enfoque metodológico, y la práctica colaborativa y reflexiva como medio para construir conocimiento significativo. Estos conceptos servirán de base para la formulación de las dimensiones y los indicadores que permitirán observar con precisión cómo se manifestó la experiencia en el aula.

Identificación de conceptos estructurantes

Entre los conceptos que sustentan teóricamente la experiencia de enseñanza del análisis y diseño de software en el primer nivel de la carrera de Ingeniería de Software de la Universidad Estatal de Milagro (UNEMI), se destacan cuatro ejes fundamentales: pensamiento computacional, aprendizaje activo, modelado de sistemas y práctica colaborativa-reflexiva. Estos conceptos emergen tanto de la práctica docente como de la literatura especializada, y constituyen los pilares conceptuales sobre los cuales se edifica la fundamentación del capítulo. Su identificación permite comprender las decisiones pedagógicas que guiaron la experiencia y facilita, posteriormente, la formulación de dimensiones analíticas e indicadores observables.

1. Pensamiento computacional: base cognitiva del análisis y diseño de software

El pensamiento computacional constituye una competencia esencial para los futuros ingenieros de software, pues permite abordar los problemas desde una lógica estructurada y sistemática. Wing (2006) lo define como el proceso mental que implica descomponer problemas, abstraer patrones y diseñar soluciones algorítmicas transferibles a distintos contextos. En el ámbito educativo, Grover y Pea (2013) subrayan que el pensamiento computacional no se reduce a programar, sino que integra la capacidad de analizar, modelar y representar procesos antes de codificarlos.

En la experiencia desarrollada, este concepto se concretó a través de actividades que exigían a los estudiantes analizar el problema antes de proponer la solución, utilizando herramientas como los árboles de problemas y los diagramas de flujo. Al hacerlo, los estudiantes comprendieron que el desarrollo de software implica una secuencia lógica de pasos orientados a la comprensión del contexto y a la delimitación del alcance del sistema. En este sentido, el pensamiento computacional no solo fue un contenido de enseñanza, sino también una metodología de razonamiento aplicada en el aula.

2.2.2. Aprendizaje activo: del estudiante receptor al estudiante protagonista

El segundo concepto estructurante es el aprendizaje activo, entendido como un enfoque que sitúa al estudiante en el centro del proceso formativo. Bonwell y Eison (1991) lo describen como un método que involucra a los estudiantes en actividades que fomentan el análisis, la evaluación y la aplicación del conocimiento, en lugar de la simple recepción de información. A su vez, Prince (2004) sostiene que las estrategias de aprendizaje activo, como el aprendizaje basado en problemas (ABP) o el aula invertida, incrementan la motivación y la retención del conocimiento al promover la participación directa del estudiante.

En esta experiencia, el aprendizaje activo se evidenció en la resolución colaborativa de enunciados de problemas, donde los estudiantes debían discutir, analizar y consensuar una propuesta de diseño. Esta dinámica permitió que asumieran un papel protagónico, transformando la clase en un espacio de construcción conjunta de conocimiento. Así, el docente pasó a ser un mediador que orienta, guía y retroalimenta los procesos de razonamiento, consolidando una práctica coherente con la enseñanza por competencias.

3. Modelado de sistemas: de la comprensión del problema a la representación formal

El modelado de sistemas constituye otro concepto clave, pues vincula la comprensión del problema con su representación técnica. Pressman y Maxim (2020) señalan que el modelado es una fase esencial en el desarrollo de software, ya que permite visualizar la estructura y el comportamiento del sistema antes de su implementación. Asimismo, Booch et al. (2005) —creadores del Lenguaje Unificado de Modelado (UML)— destacan que el modelado ayuda a abstraer los elementos críticos del sistema y facilita la comunicación entre los miembros del equipo.

Durante las clases, los estudiantes tradujeron sus análisis en diagramas de casos de uso y de flujo, representando los procesos del sistema a partir de la identificación de actores, funcionalidades y relaciones. Este paso fue decisivo para superar la tendencia a “programar sin pensar”, ya que permitió establecer un puente entre el análisis conceptual y la construcción técnica. Por tanto, el modelado de sistemas operó no solo como contenido técnico, sino también como un recurso cognitivo para estructurar el pensamiento analítico.

2.2.3. Práctica colaborativa-reflexiva: aprender con otros para construir significado

El cuarto concepto estructurante es la práctica colaborativa-reflexiva, que reconoce la importancia del trabajo en equipo y la reflexión compartida en la formación de ingenieros. Wenger (1998) introduce el concepto de comunidades de práctica para describir los espacios donde el conocimiento se construye colectivamente a través del diálogo y la interacción. En el ámbito educativo, Schön (1992) plantea que el docente reflexivo revisa sus acciones y decisiones para mejorar su práctica profesional, generando aprendizajes tanto individuales como colectivos.

En la experiencia desarrollada, los estudiantes trabajaron en grupos colaborativos para resolver problemas y diseñar propuestas conjuntas, mientras el docente promovía la discusión y la reflexión sobre los procesos seguidos. Este intercambio fortaleció la comunicación, el pensamiento crítico y la toma de decisiones compartidas, aspectos fundamentales en la formación de profesionales de software que deben trabajar en entornos interdisciplinarios. De esta manera, la práctica colaborativa-reflexiva operó como un medio para construir sentido y consolidar aprendizajes duraderos.

2.2.4. Formulación de dimensiones

A partir de los conceptos estructurantes identificados en el apartado anterior —pensamiento computacional, aprendizaje activo, modelado de sistemas y práctica colaborativa-reflexiva—, la experiencia docente desarrollada en la carrera de Ingeniería de Software de la Universidad Estatal de Milagro (UNEMI) se organiza en tres dimensiones analíticas que permiten comprenderla en profundidad: pedagógica-didáctica, cognitiva-tecnológica y colaborativa-reflexiva. Siguiendo a Flick (2014b), las dimensiones funcionan como re cortes analíticos que ayudan a ordenar la complejidad de la práctica, mientras que Jara (2018a) sostiene que constituyen un medio para traducir las experiencias en conocimiento comunicable. Estas dimensiones, por tanto, operan como un marco de lectura que da densidad conceptual y académica al proceso vivido.

1. Dimensión pedagógica-didáctica: enseñar a analizar antes de programar

Esta dimensión aborda las estrategias y decisiones de enseñanza que priorizan la comprensión del problema como punto de partida para el desarrollo de software. Stenhouse (1984) y Elliott (1993) coinciden en que la práctica docente debe concebirse como un proceso de investigación y reflexión permanente. En esta experiencia, los estudiantes par

ticiparon en actividades que los llevaron a construir árboles de problemas, identificando causas, efectos y relaciones antes de proponer una solución técnica. Esta metodología permitió que comprendieran la importancia de delimitar el alcance del proyecto y definir los objetivos del sistema antes de programar. Así, la dimensión pedagógica-didáctica da cuenta del tránsito desde un enfoque instrumental hacia una enseñanza orientada al pensamiento analítico y a la planificación estratégica.

2. Dimensión cognitiva-tecnológica: fortalecer el pensamiento lógico-computacional
Esta dimensión se relaciona con los procesos mentales y las herramientas técnicas que sustentan el aprendizaje del análisis y diseño de software. Wing (2006) define el pensamiento computacional como la capacidad de descomponer problemas, abstraer patrones y diseñar soluciones algorítmicas transferibles. Grover y Pea (2013) añaden que esta competencia implica la integración del razonamiento lógico con la representación simbólica de los procesos. En la experiencia, los estudiantes elaboraron diagramas de casos de uso y de flujo para traducir su comprensión conceptual en representaciones gráficas del sistema. Este ejercicio integró la cognición con la tecnología, reforzando la idea de que el modelado no es una actividad técnica aislada, sino un medio para estructurar el pensamiento y visualizar soluciones.

De manera integrada, los estudiantes avanzaron desde el análisis del enunciado hasta la construcción de una propuesta técnica completa. Tras elaborar el árbol de problemas, identificaron causas y efectos, y con base en esa comprensión aplicaron los principios de ingeniería de requerimientos, diferenciando las necesidades del usuario de las funcionalidades del sistema. Posteriormente, trasladaron estos requerimientos al modelado UML, mediante diagramas de casos de uso, secuencia y clases, representando la interacción entre actores, procesos y componentes del sistema. Este recorrido formativo evidencia cómo las herramientas del análisis y del diseño se articulan en un proceso continuo de razonamiento, planificación y representación técnica. Así, la dimensión cognitiva-tecnológica se consolida como el eje que integra la comprensión conceptual con la aplicación práctica, asegurando una progresión coherente en el desarrollo del pensamiento lógico-computacional.

2.2.5. Dimensión colaborativa-reflexiva: aprender con otros para construir conocimiento

La tercera dimensión pone énfasis en la interacción social y la reflexión compartida como componentes esenciales del aprendizaje. Schön (1992) plantea que la práctica

reflexiva permite a docentes y estudiantes revisar sus acciones para transformarlas en conocimiento, mientras que Wenger (1998) sostiene que las comunidades de práctica se constituyen como espacios donde el aprendizaje se produce en interacción. En esta experiencia, los estudiantes trabajaron en equipos colaborativos para analizar enunciados de problemas y proponer soluciones conjuntas. Este proceso de diálogo y confrontación de ideas fomentó la argumentación, la toma de decisiones compartidas y la responsabilidad colectiva sobre el aprendizaje. La dimensión colaborativa-reflexiva evidencia que el conocimiento se construye en la interacción y que la reflexión conjunta potencia la comprensión y la autonomía.

Estas tres dimensiones permiten analizar la experiencia docente de manera integral, mostrando cómo se articulan las decisiones pedagógicas, los procesos cognitivos y las dinámicas colaborativas. En conjunto, configuran un marco que explica cómo la enseñanza del análisis y diseño de software puede transformarse en una práctica reflexiva, activa y significativa. Este marco dimensional sienta las bases para el siguiente paso del proceso de sistematización: la construcción de indicadores, a través de los cuales se podrá observar y evidenciar con mayor precisión cómo se manifestó cada dimensión en la práctica y qué impacto tuvo en el aprendizaje de los estudiantes.

2.2.6. Construcción de indicadores

La formulación de dimensiones permitió organizar la experiencia docente en categorías analíticas que dieron coherencia al proceso reflexivo. Sin embargo, para evidenciar cómo esas dimensiones se concretan en la práctica, es necesario traducirlas en indicadores observables y verificables. Como señalan Flick (2014a) y Yin (2014), los indicadores son instrumentos que permiten operacionalizar conceptos teóricos, garantizando la consistencia entre lo que se plantea en la fundamentación y lo que realmente ocurre en el campo. De este modo, cada indicador se convierte en una señal tangible del avance y la efectividad de la experiencia, conectando teoría, práctica y evidencia empírica.

En esta experiencia, los indicadores se construyeron a partir de la observación directa en clase, la revisión de entregas en la plataforma virtual y las evaluaciones de proyectos finales. Los productos elaborados por los estudiantes —árboles de problemas, requerimientos de usuario y diagramas UML— constituyen evidencias verificables de su progreso. Stake (1995) sostiene que la credibilidad de un proceso de sistematización depende

de su vínculo con evidencias reales, razón por la cual los indicadores aquí presentados se apoyan en resultados concretos obtenidos en el aula.

2.2.7. Dimensión pedagógica-didáctica: enseñar a analizar antes de programar

Esta dimensión se expresa en la capacidad de los estudiantes para comprender un problema antes de pasar a la fase de diseño y codificación. Se observan avances cuando los estudiantes aplican estrategias de análisis, delimitan el alcance del sistema y proponen soluciones coherentes con el enunciado.

Indicadores:

1. Los estudiantes elaboran árboles de problemas que identifican causas, efectos y relaciones lógicas del escenario presentado.
2. Los estudiantes definen con claridad los límites y objetivos de la solución antes de proponer los módulos del sistema.
3. En las actividades de aula, los estudiantes sustentan oralmente sus análisis, mostrando comprensión de la problemática y coherencia argumentativa.

Ejemplo de aplicación:

Durante los talleres en clase, los equipos trabajaron con casos como la gestión de sistemas académicos o de pacientes. Antes de modelar, debían representar el problema en un árbol causal y justificar su delimitación. La observación y evaluación de estas actividades mostraron un avance significativo en la comprensión analítica y en la planificación previa al diseño del software.

2. Dimensión cognitiva-tecnológica: fortalecer el pensamiento lógico-computacional
Esta dimensión se evidencia en la aplicación de herramientas y representaciones técnicas que integran el razonamiento lógico con la abstracción del sistema. Se observan logros cuando los estudiantes traducen sus análisis en representaciones formales mediante diagramas UML coherentes y completos.

Indicadores:

1. Los estudiantes aplican la ingeniería de requerimientos diferenciando correctamente las necesidades del usuario y del sistema.

2. Los estudiantes elaboran diagramas UML (casos de uso, secuencia y clases) coherentes con el análisis previo del problema.
3. Los proyectos finales integran tecnologías externas (IA, GPS, reconocimiento facial) correctamente modeladas en las representaciones del sistema.

Ejemplo de aplicación:

En los talleres y proyectos integradores, los estudiantes desarrollaron modelos de sistemas de control de vehículos y roles de pago. A partir de los requerimientos definidos, generaron diagramas de casos de uso, secuencia y clases en UML. La observación directa y la revisión en el aula virtual mostraron mejoras en la correspondencia entre análisis, requerimientos y modelado técnico, demostrando desarrollo del pensamiento lógico-computacional.

2.2.8. Dimensión colaborativa-reflexiva: aprender con otros para construir conocimiento

Esta dimensión se manifiesta en la interacción entre pares, la reflexión conjunta y la construcción colaborativa del conocimiento. Se observa progreso cuando los estudiantes participan activamente en la resolución grupal de problemas y en la exposición de resultados.

Indicadores:

1. Los estudiantes trabajan en equipos colaborativos, discutiendo y consensuando decisiones sobre el diseño del sistema.
2. Los grupos presentan sus resultados en clase, reflexionando sobre los aciertos y dificultades de su proceso de análisis y modelado.
3. El docente facilita instancias de retroalimentación colectiva, promoviendo la revisión de estrategias y la comparación de diferentes enfoques de solución.

Ejemplo de aplicación:

En las sesiones presenciales, los grupos discutieron los resultados de sus análisis, contrastando modelos y proponiendo mejoras. Estas actividades promovieron el diálogo técnico y la argumentación, evidenciando que la práctica colaborativa incrementa la comprensión y la autonomía.

Tabla 2.1: Dimensiones, indicadores observables y evidencias asociadas

Dimensión	Indicadores observables	Evidencias asociadas
Pedagógica-didáctica	1. Elaboración de árboles de problemas. 2. Definición clara del alcance y objetivos del sistema. 3. Sustentación oral del análisis.	Talleres en clase; deberes evaluados; registros de retroalimentación.
Cognitiva-tecnológica	1. Aplicación de ingeniería de requerimientos. 2. Elaboración de diagramas UML coherentes. 3. Integración de tecnologías externas en los modelos.	Diagramas UML; proyectos integradores; entregas en el aula virtual.
Colaborativa-reflexiva	1. Trabajo grupal y toma de decisiones compartidas. 2. Presentación y discusión de resultados en clase. 3. Retroalimentación colectiva con el docente.	Observaciones en clase; presentaciones orales; registros de participación.

Nota: La tabla presenta las dimensiones evaluadas, los indicadores observables y las evidencias asociadas en el proceso de análisis.

Fuente: elaboración propia.

2.2.9. Cierre proyectivo

Los indicadores formulados permiten comprobar el impacto formativo de la experiencia en los tres planos analíticos: pedagógico, cognitivo y colaborativo. Además, proporcionan la base para construir la matriz de verificación, donde cada indicador podrá asociarse con fuentes de evidencia (productos de aula, observaciones, evaluaciones) y métodos de análisis. Así, la sistematización avanza desde la descripción narrativa hacia la validación empírica, asegurando que las dimensiones teóricas puedan demostrarse mediante resultados concretos del proceso educativo.

2.2.10. Fuentes y métodos de verificación

El proceso de sistematización requiere evidencias que respalden de forma verificable las dimensiones y los indicadores definidos. En este sentido, las fuentes y métodos de

verificación permiten demostrar que los avances observados en la práctica docente son reales, medibles y consistentes con los objetivos formativos. Como señala Flick (2014a), la credibilidad de un análisis educativo depende de la definición clara de las fuentes que sustentan las observaciones y del modo en que se las interpreta. En esta experiencia, las fuentes y métodos empleados garantizan la validez del proceso y fortalecen el carácter académico de la sistematización.

2.2.11. Fuentes de verificación

Las fuentes utilizadas para comprobar los indicadores provienen de tres tipos: documentales, testimoniales y de producto. Las fuentes documentales incluyen los talleres, deberes, informes y evaluaciones almacenadas en la plataforma virtual institucional, donde cada actividad queda registrada con calificaciones y retroalimentación. Las fuentes testimoniales se obtienen a través de la observación directa del docente durante las clases y exposiciones, en las cuales los estudiantes discuten y justifican sus decisiones de análisis o modelado. Finalmente, las fuentes de producto se componen de los materiales elaborados por los estudiantes: árboles de problemas, documentos de requerimientos, diagramas UML (casos de uso, secuencia y clases) y proyectos integradores finales.

Además, se incluyen prácticas experimentales adicionales, en las que los estudiantes resuelven varios ejercicios complementarios para consolidar el conocimiento. Estas actividades permiten verificar la comprensión y la transferencia de los aprendizajes. Las rúbricas de evaluación son una fuente clave, ya que registran con precisión los criterios de desempeño en cada actividad y evidencian el logro de los resultados de aprendizaje esperados.

2.2.12. Métodos de verificación

Para garantizar la validez del análisis se aplican varios métodos de verificación. En primer lugar, la observación directa en clase permite registrar el desempeño de los estudiantes durante el desarrollo de talleres y prácticas, especialmente su capacidad para analizar problemas y argumentar decisiones en grupo. En segundo lugar, el análisis comparativo de productos se utiliza para contrastar los resultados de las primeras actividades con los proyectos finales, observando mejoras en la identificación de requerimientos y en la coherencia de los diagramas UML. Yin (2014) sostiene que la comparación entre evi-

dencias fortalece la consistencia interna de los estudios de caso, al mostrar la evolución del aprendizaje.

Asimismo, se emplea el análisis de desempeño mediante rúbricas, que permite verificar de manera objetiva los niveles alcanzados en cada criterio: comprensión del problema, aplicación de metodologías, claridad del modelado y argumentación en la exposición. Finalmente, la triangulación entre observaciones, rúbricas y evaluaciones asegura que los resultados no dependan de una sola fuente, sino que surjan de un conjunto de evidencias convergentes, lo que incrementa la fiabilidad del proceso (Stake, 1995).

2.2.13. Ejemplo de aplicación

En el proyecto final, los estudiantes exponen un modelo de negocio, presentan los requerimientos identificados, los diagramas UML elaborados y justifican el modelo de ciclo de vida que aplicarían. Durante estas exposiciones, el docente verifica tanto la conceptualización teórica como la coherencia entre los elementos del análisis y el diseño. A partir de las rúbricas y observaciones, se evalúa la integración de los aprendizajes previos —desde el análisis del problema hasta el modelado y la planificación metodológica—, consolidando así la evidencia de las tres dimensiones: pedagógica, cognitiva y colaborativa.

2.2.14. Cierre proyecto

Las fuentes y métodos de verificación descritos garantizan que la experiencia no se base en intuiciones, sino en evidencias tangibles y múltiples. Siguiendo a Jara (2018a), la documentación rigurosa de la práctica convierte los relatos docentes en conocimiento validado, al vincular las transformaciones observadas con datos comprobables. Este apartado, por tanto, cierra la fundamentación operativa del capítulo y prepara el camino para la matriz final de dimensiones–indicadores–fuentes–métodos, donde se integrarán todos los elementos analizados en este módulo.

2.2.15. Justificación teórica del conjunto

El proceso de sistematización desarrollado hasta este punto ha permitido construir un andamiaje conceptual y metodológico coherente, capaz de transformar una práctica docente en conocimiento académico. La identificación de los conceptos estructurantes, su

organización en dimensiones, la formulación de indicadores y la definición de fuentes y métodos de verificación conforman un entramado que articula teoría, práctica y evidencia. Este apartado tiene como propósito justificar teóricamente ese conjunto, demostrando que las decisiones tomadas no responden al azar, sino a una lógica epistemológica y pedagógica fundamentada en la literatura académica y en la naturaleza de la enseñanza de la ingeniería de software en la Universidad Estatal de Milagro (UNEMI).

En primer lugar, la selección de conceptos estructurantes —pensamiento computacional, aprendizaje activo, modelado de sistemas y práctica colaborativa-reflexiva— responde a la necesidad de vincular las competencias propias de la disciplina con los principios pedagógicos de la educación superior contemporánea. Wing (2006) concibe el pensamiento computacional como una competencia transversal que permite descomponer y abstraer problemas, mientras que Grover y Pea (2013) subrayan su valor educativo al integrar el razonamiento lógico con la representación formal del conocimiento. Estos marcos teóricos se complementan con los aportes de Schön (1992) y Elliott (1993), quienes entienden la práctica docente como un espacio de reflexión e investigación-acción. Desde esta perspectiva, los conceptos seleccionados no solo describen lo que se enseña, sino también cómo se aprende a pensar dentro de la ingeniería de software.

En segundo lugar, la definición de dimensiones analíticas permitió ordenar la complejidad de la experiencia en tres grandes ejes: pedagógico-didáctico, cognitivo-tecnológico y colaborativo-reflexivo. Tal como plantea Flick (2014a), las dimensiones operan como recortes analíticos que otorgan estructura y consistencia al análisis, mientras que Jara (2018b) afirma que en la sistematización las dimensiones traducen la práctica en un lenguaje comunicable y evaluable.

En este caso, la dimensión pedagógica organiza las estrategias de enseñanza orientadas a la comprensión del problema antes de programar; la dimensión cognitivo-tecnológica articula el razonamiento lógico con el uso de herramientas de modelado; y la dimensión colaborativa-reflexiva reconoce la importancia del aprendizaje compartido y la práctica docente como acto dialógico. Esta organización asegura una lectura integral y transferible de la experiencia educativa.

La formulación de indicadores observables aportó rigor operativo y verificabilidad al proceso. Según Stake (1995), la credibilidad de un estudio de caso depende de la coherencia entre las categorías teóricas y las evidencias empíricas que las sustentan. En la misma línea, Yin (2014) enfatiza que la definición de indicadores es lo que garantiza la validez interna de un estudio, al establecer criterios claros para observar los fenómenos

en el campo. En este capítulo, los indicadores permitieron concretar cómo se manifiestan las dimensiones en la práctica: desde la elaboración de árboles de problemas hasta la construcción de diagramas UML y la aplicación de metodologías ágiles o tradicionales en los proyectos finales. Los indicadores, por tanto, transformaron la reflexión docente en un proceso medible, visible y replicable.

En cuarto lugar, la selección de fuentes y métodos de verificación refuerza la credibilidad del proceso de sistematización. Flick (2014a) advierte que la validez de los análisis cualitativos depende de la claridad en la definición de las fuentes, mientras que Yin (2014) y Jara (2018a) coinciden en que la triangulación de datos fortalece la solidez interpretativa.

En este caso, se recurrió a múltiples fuentes: talleres y evaluaciones en la plataforma virtual, observaciones en clase, proyectos finales, rúbricas de desempeño y prácticas experimentales. Estas evidencias fueron analizadas mediante métodos de observación directa, análisis comparativo y triangulación, garantizando que las conclusiones no se basaran en impresiones, sino en resultados verificables. De esta manera, la práctica docente se convirtió en un campo de indagación empírica sustentado en datos concretos.

El conjunto de decisiones tomadas —conceptuales, analíticas y metodológicas— posee coherencia teórica y pertinencia disciplinar. Carlino (2005) sostiene que la escritura académica constituye una práctica social que forma identidad profesional, mientras que Hyland (2009) resalta que el texto académico no solo comunica resultados, sino que construye comunidades de conocimiento. Este capítulo, en ese sentido, se inscribe en una línea de investigación docente que reconoce la escritura como una forma de reflexión sistemática y de validación del saber pedagógico. En el ámbito de la ingeniería de software, esta práctica adquiere particular relevancia porque permite conectar los procesos de análisis y diseño técnico con la comprensión epistemológica del aprendizaje.

Finalmente, este andamiaje conceptual y operativo consolida la validez académica de la experiencia sistematizada. La coherencia entre los conceptos, las dimensiones, los indicadores y los métodos demuestra un equilibrio entre la teoría y la práctica, cumpliendo con los criterios de credibilidad planteados por Stake (1995) y la rigurosidad metodológica defendida por Flick (2014a). Asimismo, el uso de fuentes y métodos triangulados garantiza que el conocimiento producido sea transferible a otros contextos educativos. Este conjunto, fundamentado en la reflexión pedagógica y en la práctica profesional, fortalece el vínculo curricular del capítulo: muestra cómo enseñar análisis y diseño de software puede convertirse en un proceso formativo integral que une la cognición, la tecnología y

la colaboración. Con ello, se cierra la fundamentación teórica del módulo, preparando el terreno para el análisis de resultados y aprendizajes que se abordará en el Módulo 3.

2.2.16. Integración

Este módulo me permitió articular un andamiaje conceptual y operativo sólido que transforma mi práctica docente en un proceso académico fundamentado. A lo largo de los cinco puentes recorridos, identifiqué los conceptos estructurantes que definen mi experiencia, los organicé en dimensiones analíticas, construí indicadores verificables y precisé las fuentes y métodos que garantizan la credibilidad del análisis. Todo este proceso culminó en una justificación teórica que demuestra la coherencia entre los elementos y el respaldo bibliográfico que los sustenta. En conjunto, estos avances consolidan un marco riguroso que integra teoría, práctica y evidencia, otorgando sentido y consistencia a la enseñanza del análisis y diseño de software en la carrera de Ingeniería de Software de la UNEMI.

La integración de estos elementos me da la seguridad de que el capítulo ya posee una fundamentación conceptual y metodológica robusta, capaz de sostener el análisis que vendrá en el siguiente módulo. Haber transitado este recorrido me confirma que la sistematización no solo organiza la experiencia, sino que la convierte en conocimiento comunicable y transferible a otros contextos docentes. El trabajo realizado garantiza validez académica, coherencia interna y respaldo teórico, brindándome la confianza para avanzar hacia el Módulo 3, donde el foco estará en el análisis e interpretación de los resultados y en la identificación de los aportes formativos de esta experiencia educativa.

2.3. Vínculo curricular y resultados de aprendizaje en la enseñanza del análisis y diseño de software

2.3.1. Transición al vínculo curricular

Al culminar el Módulo 2, la experiencia docente sistematizada alcanzó una fundamentación conceptual y operativa sólida. El proceso permitió definir los conceptos estructurantes, organizarlos en dimensiones, formular indicadores y respaldarlos con fuentes y métodos de verificación que garantizan su validez académica. Este andamiaje fortaleció la comprensión teórica y metodológica de la práctica, transformándola en un referente

analítico que demuestra cómo la enseñanza del análisis y diseño de sistemas de software puede investigarse y comunicarse con rigor científico.

En este nuevo tramo, el foco se desplaza hacia la articulación curricular. La experiencia no se circumscribe a una sola asignatura, sino que integra contenidos clave de varias materias que imparto dentro de la carrera de Ingeniería de Software, Ingeniería de Requerimientos, Modelado con UML y Patrones y Arquitecturas de Software. Este conjunto de temas se relaciona directamente con las competencias del perfil de egreso, que busca formar profesionales capaces de analizar, modelar, diseñar e implementar soluciones informáticas sólidas, aplicando metodologías ágiles, pensamiento lógico y criterios de calidad en el desarrollo de software.

De este modo, el trabajo sistematizado se convierte en una experiencia curricular integradora, que articula teoría, práctica y reflexión. Las dimensiones e indicadores formulados en el módulo anterior —pedagógica-didáctica, cognitiva-tecnológica y colaborativa-reflexiva— encuentran ahora correspondencia en las competencias que promueven el razonamiento analítico, la abstracción de sistemas, la comunicación efectiva y el trabajo en equipo. Este apartado marca el inicio del análisis curricular, orientado a mostrar cómo la práctica docente dialoga con el plan de estudios y cómo contribuye a consolidar los aprendizajes profesionales que la carrera de Ingeniería de Software propone como eje de formación

2.3.2. Identificación de competencias del perfil

El análisis de la experiencia docente muestra que el trabajo transversal en análisis y diseño de sistemas de software —integrando ingeniería de requerimientos, modelado con UML y patrones/arquitecturas— contribuye de manera directa al perfil de egreso de la carrera de Ingeniería de Software. Esta contribución no es circunstancial, sino estructural: se alinea con competencias específicas de la disciplina (analizar, modelar y diseñar soluciones) y con competencias genéricas (pensamiento crítico, trabajo colaborativo, comunicación técnica y responsabilidad ética). En términos curriculares, la experiencia se sitúa como un espacio integrador que articula saberes conceptuales, procedimentales y actitudinales, y que permite a los estudiantes transitar desde la comprensión del problema hasta la representación formal de la solución, antes de escribir una sola línea de código.

En el plano de las competencias específicas, la experiencia fortalece de manera central la competencia de análisis y diseño de software. Al exigir que el estudiante parte de una

problemática real, construya un árbol de problemas, derive requerimientos de usuario y del sistema, y los traslade a diagramas UML (casos de uso, secuencia y clases), se desarrolla una forma de pensar propia de la ingeniería: descomponer, abstraer, modelar y justificar. Este itinerario sitúa el pensamiento lógico-computacional como base cognitiva del desempeño profesional, en sintonía con Wing (2006) y con la síntesis que realizan Grover y Pea (2013) respecto de su valor educativo. En consecuencia, el curso de acción de la experiencia —análisis → requerimientos → modelado— es, al mismo tiempo, un curso de formación de la competencia que el perfil demanda.

La competencia de pensamiento crítico y lógico-computacional se activa cuando los estudiantes deben interpretar correctamente el modelo de negocio y sus procesos antes de modelar, y leer/validar sus propios diagramas y los de sus pares. El hecho de diferenciar lo esencial de lo accesorio, detectar inconsistencias entre requerimientos y casos de uso, o analizar si un diagrama de secuencia refleja la lógica temporal de las interacciones, son prácticas que evidencian la madurez analítica. Tales prácticas conectan, además, con la idea de evaluación auténtica: el estudiante demuestra su comprensión en productos profesionales (diagramas y documentos de requerimientos) y no solo en pruebas teóricas, alineándose con recomendaciones de ABET y de los lineamientos ACM/IEEE para programas de Ingeniería de Software.

En relación con la competencia de responsabilidad ética y profesional, la experiencia incorpora discusiones sobre impactos y criterios de calidad de las decisiones de diseño. Cuando los equipos justifican la inclusión de entidades externas (IA, GPS, reconocimiento facial) o la elección de un modelo de ciclo de vida (tradicional o ágil), deben considerar implicaciones de privacidad, sesgo, seguridad y mantenibilidad, así como la viabilidad de la solución en el contexto organizacional. Este componente ético —frecuentemente subestimado en etapas tempranas— aparece aquí como un criterio transversal que orienta el diseño responsable: no todo lo técnicamente posible es educativo o socialmente deseable, y el perfil de egreso exige explicitar ese razonamiento.

Entre las competencias genéricas, la experiencia potencia trabajo colaborativo y comunicación técnica. En los talleres y proyectos, los equipos organizan roles, negocian criterios de diseño y distribuyen tareas; la sinergia y la toma de decisiones compartidas son observables durante la construcción de los modelos. A su vez, en las exposiciones del proyecto final, los estudiantes presentan el modelo de negocio, justifican requerimientos, defienden sus diagramas UML y argumentan la elección de patrones o arquitecturas, lo que demanda claridad expositiva y dominio del discurso técnico. Este desempeño se

ajusta a la concepción de competencia colaborativa como práctica situada (Villa & Po-blete, 2007) y a la noción de escritura/ comunicación académica como práctica social e identitaria (Carlino, 2005); (Hyland, 2009).

La relevancia curricular de la experiencia se evidencia en la trazabilidad entre actividades, productos y logros. El portafolio de entregas (árboles de problemas, especificaciones de requerimientos, casos de uso, secuencias y clases) y las rúbricas asociadas permiten mapear con precisión qué componentes del perfil se movilizan y con qué nivel de desempeño. Por ejemplo, la coherencia entre requerimientos y casos de uso indica dominio del análisis funcional; la consistencia entre casos de uso y clases revela dominio de abstracción y diseño; las presentaciones orales y la documentación técnica muestran comunicación profesional; y la gestión del equipo a lo largo del proyecto evidencia colaboración y liderazgo. Esta evidencia empírica respalda la afirmación de que el proceso fortalece las competencias declaradas en el perfil de egreso.

Un aporte distintivo de tu práctica es la integración temprana de tecnologías y contextos (IA, GPS, reconocimiento facial), que obliga a los estudiantes a ajustar su modelado a restricciones reales y a considerar patrones y arquitecturas adecuados. Elegir, por ejemplo, un patrón MVC para organizar responsabilidades, o discutir si un estilo orientado a servicios o microservicios tiene sentido para el dominio planteado, activa competencias de pensamiento arquitectónico y diseño justificable. Este nivel de reflexión técnica, guiado por criterios, prepara a los estudiantes para escenarios de mayor complejidad y acorta la distancia entre el aula y la práctica profesional.

Ahora bien, identificar competencias no es solo nombrarlas, sino describir su manifestación. En la experiencia, los resultados observables son claros: (a) en colaboración, los equipos muestran organización de roles, sinergia y liderazgo; (b) en pensamiento analítico, se aprecia precisión en UML y capacidad de interpretar diagramas y procesos de negocio antes del modelado; (c) en comunicación técnica, hay exposiciones estructuradas que justifican el porqué de los requerimientos, patrones y arquitecturas elegidos; (d) en ética profesional, los estudiantes argumentan impactos y trade-offs de las decisiones de diseño. Estos resultados forman un corpus verificable que se registra en aula virtual, rúbricas y presentaciones.

Desde la perspectiva del diseño curricular basado en competencias, esta experiencia se alinea con marcos internacionales (ABET, 2020); (ACM & IEEE, 2020) que enfatizan la integración de análisis, diseño, trabajo en equipo y comunicación como resultados esperados de la formación. En particular, SE2020 destaca la necesidad de que los estudiantes

practiquen especificación de requerimientos y modelado con trazabilidad hacia arquitecturas y criterios de calidad, algo que tu práctica sitúa en el centro. Por ello, el apartado no solo identifica competencias, sino que demuestra con evidencia de desempeño que dichas competencias se ejercitan y se evalúan con instrumentos pertinentes.

Esta identificación también permite proyectar con claridad hacia los resultados de aprendizaje (RA) de asignaturas afines. El dominio del análisis y diseño antes de programar incide positivamente en RA como “elabora modelos coherentes con requerimientos”, “aplica criterios de calidad en el diseño” o “comunica soluciones técnicas con precisión”. Al cierre del ciclo, el proyecto final opera como evaluación integradora: la correspondencia entre una situación real, requerimientos, modelos y justificación de patrones/arquitecturas ofrece una visión holística del progreso, y constituye un indicador fuerte de alineación con el perfil de egreso.

En síntesis, la experiencia sistematizada fortalece competencias específicas (análisis y diseño, pensamiento lógico-computacional, responsabilidad ética) y genéricas (trabajo colaborativo, comunicación técnica), y lo hace con trazabilidad y verificación. Este apartado muestra que no se trata de una práctica aislada, sino de un dispositivo curricular que dialoga con el perfil profesional y con los estándares de la disciplina. Al quedar clara la integración entre conceptos, dimensiones, indicadores y evidencias, el capítulo avanza con bases sólidas hacia el análisis del Módulo 3, donde se examinarán con mayor precisión los resultados de aprendizaje derivados de esta experiencia y su transferibilidad a otros contextos formativos.

2.3.3. Resultados de aprendizaje vinculados

La vinculación entre las competencias del perfil de egreso y los resultados de aprendizaje alcanzados constituye el eje articulador entre la práctica docente y el currículo de la carrera de Ingeniería de Software. En la Universidad Estatal de Milagro (UNEMI), el Modelo Educativo (2021) y el Reglamento de Régimen Académico del CES (2019) establecen que los resultados de aprendizaje son manifestaciones observables de lo que el estudiante demuestra al finalizar un proceso formativo. En esta experiencia, los resultados se materializaron en contextos auténticos, donde los estudiantes analizaron situaciones reales —como sistemas de riego, citas médicas, reservaciones o procesos bancarios— para diseñar soluciones de software viables, fortaleciendo la coherencia entre el saber, el hacer y el ser profesional.

El primer resultado de aprendizaje identificado corresponde al análisis y definición de requerimientos. Los estudiantes lograron identificar y delimitar problemas mediante la construcción de árboles de causas y efectos, diferenciando los requerimientos de usuario y del sistema. Este resultado refleja la capacidad de comprender el contexto y abstraer la lógica de un problema real, alineándose con la competencia específica de análisis y diseño de software. De acuerdo con Biggs y Tang (2011), el aprendizaje significativo se produce cuando los estudiantes relacionan lo que aprenden con situaciones auténticas, y en este caso, la elaboración de requerimientos se convirtió en un ejercicio de pensamiento crítico aplicado a escenarios concretos.

Un segundo resultado se observa en la representación de procesos y estructuras mediante diagramas UML. Los estudiantes modelaron casos de uso, secuencias y clases, estableciendo correspondencia entre los requerimientos y los elementos del modelo. Este resultado evidencia el desarrollo del pensamiento lógico-computacional y de la competencia técnica para abstraer y representar sistemas, siguiendo los lineamientos internacionales de ABET (2020) y del currículo SE2020 (ACM & IEEE, 2020). En las prácticas experimentales, los estudiantes demostraron precisión y coherencia al traducir escenarios cotidianos —como la gestión de citas médicas o de reservas hoteleras— en modelos de interacción y estructura, aplicando los principios de trazabilidad entre análisis y diseño.

El tercer resultado se relaciona con la selección y justificación de soluciones técnicas, en especial la capacidad para elegir patrones o arquitecturas apropiadas al dominio del problema. Los estudiantes aprendieron a justificar sus decisiones considerando criterios de calidad del software y variables éticas, como la protección de datos o la sostenibilidad tecnológica. Este aprendizaje está estrechamente vinculado con la competencia de responsabilidad ética y profesional, promovida en el perfil de egreso. Pressman y Maxim (2020) sostienen que el ingeniero de software debe fundamentar sus elecciones técnicas en principios de calidad, ética y viabilidad, un enfoque que se aplicó en las discusiones y exposiciones finales de los proyectos.

Un cuarto resultado corresponde al trabajo colaborativo y liderazgo en entornos de desarrollo. Los equipos de estudiantes organizaron roles —analista, modelador, diseñador—, tomaron decisiones conjuntas y gestionaron la integración de sus entregables. Este logro se alinea con la competencia genérica de trabajo en equipo y comunicación efectiva, planteada en el plan de estudios de la carrera. Las rúbricas y observaciones de aula mostraron evidencias de sinergia, liderazgo y resolución conjunta de problemas. Según Villa y Poblete (2007), la colaboración auténtica se construye en contextos donde las tareas

exigen responsabilidad compartida y negociación de criterios, condiciones que caracterizaron esta experiencia docente.

El quinto resultado de aprendizaje se asocia con la comunicación y argumentación técnica. Los estudiantes presentaron oral y documentalmente sus modelos, justificaron requerimientos, explicaron diagramas y defendieron las decisiones arquitectónicas adoptadas. Este desempeño se vincula tanto con la competencia de comunicación técnica como con la ética profesional, al implicar la exposición transparente y fundamentada de procesos y resultados. Hyland (2009) y Carlino (2005) coinciden en que la escritura y la oralidad académicas son prácticas sociales que consolidan la identidad profesional; en este sentido, las presentaciones de los proyectos finales se constituyeron en espacios formativos de alto valor académico y comunicativo.

Los cinco resultados de aprendizaje identificados están respaldados por evidencias verificables —portafolios digitales, entregas en el aula virtual, rúbricas de evaluación y exposiciones orales— que muestran la coherencia entre las competencias declaradas en el perfil de egreso y los aprendizajes logrados en la experiencia. Esta correspondencia cumple con el principio de alineamiento constructivo propuesto por Biggs y Tang (2011), en el que los objetivos, las actividades y las evaluaciones forman un todo coherente. Así, la experiencia no solo fortaleció la comprensión técnica de los estudiantes, sino también su capacidad de aplicar el conocimiento en contextos reales, confirmando la validez curricular y profesional del proceso formativo.

2.3.4. Actividades y evidencias

La identificación de actividades y evidencias constituye un componente esencial en la sistematización de experiencias docentes, pues permite demostrar la coherencia pedagógica entre los resultados de aprendizaje alcanzados y las acciones desarrolladas en el aula. De acuerdo con Biggs y Tang (2011), la calidad educativa se garantiza cuando existe alineamiento constructivo entre los objetivos, las estrategias de enseñanza y las evaluaciones. En esta experiencia, las actividades diseñadas respondieron intencionadamente a las competencias y resultados de aprendizaje definidos en el currículo de la carrera de Ingeniería de Software, generando evidencias verificables que demuestran la pertinencia y efectividad del proceso formativo.

Una de las primeras actividades fue el análisis de enunciados basados en situaciones reales, tales como sistemas de riego automatizado, gestión de citas médicas, control de

vehículos de carga o plataformas de reservaciones hoteleras. Estas problemáticas, extraídas de contextos cotidianos, permitieron que los estudiantes se aproximan a la práctica profesional de forma contextualizada. La actividad culminó con la elaboración de árboles de problemas y documentos de requerimientos, en los que identificaron causas, efectos y necesidades del usuario. Estas evidencias reflejan el desarrollo del pensamiento analítico y lógico-computacional, así como la comprensión de los procesos del dominio antes de proponer soluciones técnicas.

Posteriormente, se llevó a cabo una actividad de modelado de procesos y estructuras mediante UML, en la que los estudiantes utilizaron herramientas digitales como Draw.io, Lucidchart o Visual Paradigm. En esta fase, construyeron diagramas de casos de uso, secuencia y clases, garantizando la trazabilidad entre los requerimientos y los modelos generados. Los productos obtenidos constituyen evidencias tangibles del resultado de aprendizaje vinculado al diseño de sistemas de software, y demuestran la capacidad de los estudiantes para abstraer, estructurar y representar información compleja. Según Tobón (2013), el aprendizaje basado en competencias requiere de tareas que promuevan el desempeño auténtico; en este caso, el modelado sirvió como escenario de aplicación práctica y evaluación directa de la comprensión conceptual.

Otra de las actividades clave fue la selección y justificación de patrones y arquitecturas de software, en la que los equipos analizaron alternativas como MVC, Cliente-Servidor o Microservicios, considerando criterios de calidad, mantenibilidad y viabilidad. Este ejercicio fomentó el pensamiento crítico y la toma de decisiones fundamentadas, en consonancia con la competencia de responsabilidad ética y profesional. Las evidencias derivadas —informes técnicos, debates y exposiciones orales— mostraron la madurez reflexiva de los estudiantes y su capacidad para argumentar desde una perspectiva técnica y ética. Kolb (1984) señala que el aprendizaje experiencial se consolida cuando la acción se acompaña de reflexión; por ello, la discusión sobre las implicaciones de cada elección permitió transformar la práctica técnica en un proceso formativo integral.

El trabajo colaborativo representó otra dimensión central del proceso. Las actividades en equipo implicaron la asignación de roles —analista, modelador, diseñador— y la coordinación del desarrollo de los proyectos. Cada grupo debió consolidar un portafolio de entregas, donde integraron sus diagramas, documentos de requerimientos y reflexiones individuales. Estas evidencias no solo dieron cuenta del logro de los resultados de aprendizaje vinculados con la comunicación técnica y el liderazgo colaborativo, sino que también fortalecieron habilidades blandas como la planificación, la negociación y la responsabi-

lidad compartida. Villa y Poblete (2007) destacan que el trabajo cooperativo auténtico potencia la responsabilidad individual y la interdependencia positiva; estos principios estuvieron presentes en todas las dinámicas del aula.

Finalmente, el proceso culminó con una presentación final del proyecto integrador, en la que los equipos defendieron su propuesta de solución, sustentando el modelo de negocio, los requerimientos, los diagramas UML y la elección de patrones arquitectónicos. Esta actividad permitió evaluar las competencias de comunicación oral y escrita, así como la coherencia global del diseño. Las evidencias recogidas incluyeron grabaciones de las exposiciones, informes escritos y rúbricas de evaluación. Estas fuentes demostraron la capacidad de los estudiantes para argumentar sus decisiones y para integrar el conocimiento teórico con la práctica profesional.

En conjunto, las evidencias recogidas —árboles de problemas, documentos de requerimientos, diagramas UML, informes de diseño, presentaciones orales, rúbricas y portafolios digitales— muestran la trazabilidad completa entre el currículo, las competencias, los resultados de aprendizaje y las actividades implementadas. Su pertinencia radica en que todas ellas representan desempeños observables, evaluables y contextualizados, en correspondencia con el Modelo Educativo de la Universidad Estatal de Milagro (UNE-MI) (2021), que promueve la formación práctica, reflexiva y orientada a resultados. De esta manera, la experiencia sistematizada se consolida como un ejemplo de coherencia curricular y pedagógica, donde cada acción docente responde a un propósito formativo verificable y cada evidencia respalda la validez del proceso educativo.

2.3.5. Reflexión sobre la alineación curricular

Reflexionar sobre la alineación curricular implica reconocer que toda práctica docente adquiere verdadero sentido cuando se conecta con el proyecto formativo de la carrera. En la experiencia sistematizada, esta alineación fue un proceso consciente y progresivo que buscó vincular la enseñanza del análisis y diseño de sistemas de software con las competencias y resultados de aprendizaje del perfil de egreso de Ingeniería de Software en la Universidad Estatal de Milagro. Desde esta perspectiva, la experiencia permitió evidenciar cómo las estrategias didácticas —centradas en la resolución de problemas reales y el modelado técnico— se articularon con los objetivos curriculares y fortalecieron la formación profesional de los estudiantes.

Una primera fortaleza de la alineación fue su coherencia pedagógica. Las actividades diseñadas —como el análisis de enunciados, la construcción de árboles de problemas, la especificación de requerimientos y el modelado UML— mantuvieron una relación directa con las competencias declaradas en el currículo, asegurando que cada acción en el aula respondiera a un propósito formativo. Este principio coincide con lo que Biggs y Tang (2011) denominan alineamiento constructivo, en el que los objetivos, las actividades y las evaluaciones conforman un sistema coherente que guía el aprendizaje. Gracias a ello, los resultados observables —diagramas, informes, exposiciones— no fueron productos aislados, sino evidencias integradas que demostraron el cumplimiento de los resultados de aprendizaje previstos.

Otra fortaleza radicó en la pertinencia profesional de las actividades, las cuales trasladaron los conocimientos teóricos a contextos reales de aplicación. Los estudiantes trabajaron con enunciados basados en situaciones del entorno —como bancos, clínicas, hoteles o sistemas de riego—, lo que les permitió desarrollar competencias técnicas y transversales de manera significativa. Según Tobón (2013), el currículo basado en competencias debe propiciar aprendizajes situados que articulen saber, hacer y ser; y esta experiencia cumplió ese propósito al fomentar el pensamiento analítico, la colaboración y la responsabilidad ética en escenarios cercanos a la práctica profesional. Además, la vinculación con el perfil de egreso consolidó el enfoque de aprendizaje experiencial, en el que la práctica se convierte en fuente directa de conocimiento (Kolb, 1984).

Desde el punto de vista institucional, la experiencia también aporta al fortalecimiento del currículo de la carrera al visibilizar prácticas docentes transferibles y replicables. La metodología aplicada —integrando análisis, diseño y reflexión— puede adaptarse a otras asignaturas del área técnica o de proyectos, contribuyendo a la coherencia horizontal del plan de estudios. Este enfoque coincide con Stenhouse (1987), quien sostiene que el docente que investiga su práctica se convierte en agente de innovación curricular. En este sentido, la experiencia no solo cumplió con la intención de formar en competencias, sino que generó un modelo didáctico que podría inspirar mejoras colectivas dentro de la carrera.

No obstante, el proceso de alineación también reveló tensiones y desafíos. Una de las principales dificultades fue equilibrar el tiempo destinado al desarrollo técnico con los espacios de reflexión y metacognición. En carreras de ingeniería, la urgencia por avanzar en contenidos instrumentales puede reducir los momentos de análisis pedagógico, generando una brecha entre la acción y la comprensión profunda del aprendizaje. Zabalza (2003b)

advierte que los currículos deben mantener flexibilidad para incorporar innovaciones docentes, pero esto requiere apoyo institucional y ajustes en la planificación académica. Asimismo, otro desafío fue la heterogeneidad del estudiantado, pues coexistían diferentes niveles de preparación técnica, lo que obligó a diversificar estrategias para alcanzar los resultados de aprendizaje con equidad.

También se identificaron limitaciones estructurales en torno a la evaluación. Aunque las rúbricas y los portafolios permitieron evidenciar los logros, aún es necesario consolidar mecanismos institucionales que integren estas prácticas en la evaluación formal de la carrera. Barnett (2001) plantea que el currículo universitario debe preparar a los estudiantes para actuar en contextos inciertos y complejos; lograrlo requiere no solo competencias técnicas, sino también procesos de evaluación que midan la autonomía, la reflexión y la capacidad de adaptación. En ese sentido, esta experiencia evidenció avances significativos, pero también la necesidad de seguir fortaleciendo los instrumentos de seguimiento curricular.

Como resultado de esta reflexión, se puede afirmar que la experiencia estuvo altamente alineada con el currículo en términos de competencias, resultados y evidencias, pero aún requiere ajustes en la consolidación institucional de sus prácticas innovadoras. Este ejercicio deja aprendizajes valiosos: la importancia de planificar con visión curricular, de articular el trabajo de aula con los propósitos de la carrera y de asumir la docencia como un proceso de investigación permanente. La sistematización permitió comprender que la calidad educativa se construye desde la práctica reflexiva y colaborativa, en un diálogo continuo entre la teoría, el currículo y la realidad profesional.

2.3.6. Integración curricular del Módulo 3

El desarrollo del Módulo 3 permitió consolidar la articulación curricular de la experiencia docente en el ámbito del análisis y diseño de sistemas de software, integrando la enseñanza de la ingeniería de requerimientos, el modelado con UML y la aplicación de patrones y arquitecturas de software. Este proceso evidenció que la práctica no se reduce a un conjunto de ejercicios técnicos, sino que constituye un espacio de formación integral que moviliza competencias cognitivas, procedimentales y actitudinales del perfil de egreso de la carrera de Ingeniería de Software. Al identificar las competencias específicas y genéricas más relevantes —análisis y diseño, pensamiento lógico-computacional, comunicación técnica, trabajo colaborativo y responsabilidad ética—, se estableció una

base sólida que permitió trazar la relación entre los aprendizajes logrados en el aula y los objetivos institucionales del currículo.

El análisis mostró que las actividades diseñadas en la experiencia respondieron a una lógica de coherencia pedagógica. Cada tarea se planificó como parte de un itinerario formativo orientado al logro de resultados observables: desde la lectura y comprensión de enunciados reales, pasando por la delimitación de problemas y la definición de requerimientos, hasta el modelado estructurado y la defensa oral de los proyectos. Este recorrido práctico-teórico permitió que los estudiantes aplicaran los contenidos de las asignaturas a situaciones auténticas, fortaleciendo su capacidad de análisis y abstracción. De acuerdo con Tobón (2013), el aprendizaje por competencias exige escenarios en los que los estudiantes actúen sobre contextos reales; bajo este principio, la experiencia representó un ejemplo de alineamiento constructivo (Biggs & Tang, 2011), donde las competencias, resultados, actividades y evidencias conforman un circuito coherente que garantiza la validez curricular.

Entre los principales aportes de la experiencia al currículo se destaca la consolidación de una metodología didáctica integradora, que vincula teoría, práctica y reflexión. La secuencia de actividades basada en problemas del entorno —como sistemas de riego, reservas, control vehicular o gestión de pacientes— contribuyó al desarrollo de una mirada profesional aplicada, en la que los estudiantes comprendieron la importancia de diseñar antes de programar, de modelar antes de construir, y de argumentar antes de decidir. Esta progresión cognitiva favoreció el pensamiento sistémico y la comprensión del ciclo de vida del software, competencias clave para el perfil del ingeniero. Asimismo, la incorporación de herramientas digitales de modelado fomentó la apropiación tecnológica y la autonomía en el aprendizaje, componentes fundamentales del Modelo Educativo de la Universidad Estatal de Milagro (UNEMI) (2021), que promueve la formación práctica y el aprendizaje activo.

Otro aporte significativo se relaciona con la evaluación auténtica del aprendizaje. Las evidencias recogidas —árboles de problemas, especificaciones de requerimientos, diagramas UML, informes técnicos, presentaciones orales y rúbricas de desempeño— permitieron verificar de manera concreta los avances logrados en cada resultado de aprendizaje. Estas evidencias se registraron en la plataforma institucional, generando trazabilidad y transparencia en el proceso formativo. En línea con Kolb (1984), la práctica reflexiva se consolida cuando el aprendizaje se evidencia a través de la experiencia; así, cada producto entregado no solo funcionó como un instrumento de evaluación, sino como un documen-

to de aprendizaje. Esta estructura de evidencias demuestra que el currículo puede hacerse visible en la práctica docente, convirtiendo la experiencia en un medio legítimo para verificar competencias y no solo en un espacio de aplicación instrumental.

La experiencia también aporta al fortalecimiento del perfil de egreso, al demostrar que las competencias técnicas no pueden desligarse de las competencias sociales y éticas. El trabajo colaborativo y la comunicación técnica fueron elementos transversales presentes en todo el proceso. La distribución de roles, la toma de decisiones conjunta y la defensa oral de los resultados contribuyeron al desarrollo de habilidades de liderazgo, trabajo en equipo y argumentación fundamentada. Villa y Poblete (2007) señalan que las competencias genéricas, como la cooperación y la comunicación, son las que permiten la transferencia de los aprendizajes a diferentes contextos profesionales. En este sentido, la experiencia docente analizada fomenta una formación integral, donde los futuros ingenieros de software desarrollan tanto la capacidad de diseñar soluciones tecnológicas como la de participar responsablemente en equipos de desarrollo interdisciplinarios.

A nivel curricular, el proceso permitió identificar y superar tensiones propias del diseño por competencias. Una de ellas es la necesidad de equilibrar la carga técnica con los espacios de reflexión y retroalimentación formativa. En carreras tecnológicas, suele priorizarse el dominio de herramientas y lenguajes, pero esta experiencia mostró que el pensamiento analítico y la justificación de decisiones son igualmente importantes para el perfil profesional. Además, se evidenció la conveniencia de ajustar los tiempos académicos para garantizar que la evaluación no sea un acto final, sino un proceso continuo de observación y mejora. En este aspecto, la experiencia coincide con Zabalza (2003b), quien defiende un currículo flexible que integre innovación, evaluación y reflexión permanente.

Asimismo, el proceso dejó aprendizajes institucionales relevantes. En primer lugar, la importancia de documentar las prácticas docentes como parte del desarrollo curricular continuo, aspecto esencial para procesos de acreditación como el futuro EUR-ACE, hacia el cual la carrera se orienta. En segundo lugar, la necesidad de fortalecer los espacios de socialización pedagógica entre docentes, para compartir estrategias que vinculen competencias, resultados y evidencias de manera sistemática. Finalmente, la experiencia demostró que la innovación metodológica puede provenir desde la práctica, reafirmando la idea de Stenhouse (1987) de que el profesor investigador contribuye al currículo al reflexionar y sistematizar su propia enseñanza.

La integración lograda en este módulo confirma que la experiencia docente no fue una iniciativa aislada, sino una propuesta curricular con impacto comprobable. Las com-

petencias desarrolladas se evidencian en desempeños observables y las actividades implementadas muestran coherencia con el plan de estudios de Ingeniería de Software. Esta alineación refuerza la pertinencia del proceso de enseñanza-aprendizaje dentro del modelo educativo institucional, al mismo tiempo que proyecta posibilidades de replicabilidad en otras asignaturas técnicas. Tal como plantea Barnett (2001), el currículo universitario debe preparar a los estudiantes para enfrentar la complejidad y la incertidumbre de los entornos reales; en esa línea, el trabajo desarrollado prepara a los futuros ingenieros no solo para diseñar sistemas, sino para comprender y adaptarse críticamente a los contextos donde esos sistemas se implementan.

Finalmente, esta integración curricular deja establecida una base sólida para el Módulo 4, que abordará el análisis de resultados y aprendizajes. A partir de las evidencias obtenidas, se examinarán los impactos formativos de la experiencia, las transformaciones observadas en los estudiantes y las implicaciones para la mejora continua del currículo. Este cierre confirma que la experiencia sistematizada es, en sí misma, un ejercicio de investigación pedagógica aplicada, donde el docente actúa como mediador entre la teoría y la práctica, y donde el currículo se convierte en un espacio vivo, flexible y en constante construcción.

2.4. Del problema a la solución: estrategias para enseñar análisis y diseño de software en primer nivel

2.4.1. Transición hacia la operacionalización estratégica

Hasta este punto, el capítulo ha mostrado la coherencia curricular de la experiencia: las competencias del perfil de egreso fueron identificadas, vinculadas con resultados de aprendizaje y evidenciadas a través de actividades y productos verificables. Esta estructura consolidó una visión articulada del proceso formativo, donde cada acción pedagógica respondió a un propósito curricular claro. Sin embargo, comprender cómo se alcanzaron estos logros requiere mirar más allá de la planeación académica y adentrarse en la dimensión estratégica de la práctica docente. En este nuevo tramo del capítulo, la atención se desplaza hacia la manera en que las estrategias diseñadas y aplicadas en el aula permitieron que esas competencias se convirtieran en desempeños reales.

Observar la experiencia desde lo estratégico significa reconocer que los aprendizajes no emergen solo de las intenciones declaradas, sino del andamiaje de decisiones, secuencias y recursos que el docente pone en juego para sostener el proceso educativo. En el caso de la enseñanza del análisis y diseño de sistemas de software, las estrategias se convirtieron en el motor que articuló la teoría con la práctica. Las estrategias núcleo —como el aprendizaje basado en problemas reales, el trabajo colaborativo y el modelado progresivo con UML— fueron acompañadas por estrategias de soporte, como la retroalimentación continua y el uso del aula virtual, y complementadas con estrategias de contingencia que permitieron adaptar la experiencia ante limitaciones técnicas o de tiempo. Este entramado de acciones configuró la base operativa del ecosistema pedagógico, garantizando la coherencia entre el currículo y la práctica.

Este paso hacia la operacionalización estratégica marca el inicio de una nueva mirada sobre la experiencia: ya no se trata únicamente de describir los aprendizajes logrados, sino de analizar cómo se lograron y por qué las decisiones tomadas resultaron efectivas. A partir de aquí, se presentarán las estrategias núcleo que estructuraron la experiencia, seguidas por las estrategias de soporte y de contingencia que la sostuvieron, mostrando cómo, en conjunto, dieron forma a un ecosistema de aprendizaje integral y sostenible.

2.4.2. Estrategias núcleo en acción

El desarrollo de la experiencia se sostuvo en un conjunto de estrategias núcleo que funcionaron como el eje articulador entre la planificación curricular y la práctica en el aula. Estas estrategias no fueron elementos accesorios, sino la base operativa que garantizó la coherencia entre las competencias, los resultados de aprendizaje y las evidencias producidas. En este contexto, tres estrategias resultaron decisivas: el aprendizaje basado en problemas reales (ABP), el modelado progresivo con UML y el trabajo colaborativo con evaluación por desempeño. Su aplicación articulada configuró una ingeniería didáctica centrada en la comprensión profunda de los procesos, la reflexión sobre la práctica y la producción de soluciones tecnológicas contextualizadas. La primera estrategia, el aprendizaje basado en problemas reales, permitió situar el aprendizaje en contextos auténticos del entorno profesional. A lo largo del semestre —de quince semanas con sesiones de tres horas— se combinaron casos elaborados por la docente con problemáticas propuestas por los propios estudiantes, relacionadas con sistemas de riego, gestión de citas médicas, con-

trol de vehículos de carga o reservas hoteleras. Esta doble vía fomentó la creatividad, la autonomía y el pensamiento crítico.

El trabajo se estructuró en una secuencia clara: primero, comprensión del problema; luego, identificación de requerimientos funcionales y no funcionales; y finalmente, propuesta de solución tecnológica. Para ello, los estudiantes debían investigar los procesos del dominio, analizar flujos de información y delimitar el alcance de la solución mediante árboles de problemas. Esta metodología se alinea con los principios del ABP planteados por Barrows (1986), donde el conocimiento surge de la resolución de problemas significativos, y con el ciclo de aprendizaje experiencial de Kolb (1984), que integra acción, reflexión y conceptualización. Las evidencias obtenidas —documentos de requerimientos, diagramas causales y presentaciones orales— demostraron que los estudiantes lograron abstraer los elementos esenciales de los casos y vincularlos con principios de ingeniería de software.

La segunda estrategia núcleo fue el modelado progresivo con UML, que transformó el análisis conceptual en representación formal del sistema. El proceso se desarrolló con el apoyo de herramientas digitales como Draw.io y Visual Paradigm, que facilitaron el trabajo tanto individual como colaborativo. La secuencia de modelado se organizó en etapas graduales: primero, los diagramas de casos de uso, donde los estudiantes identificaron los actores y las funcionalidades principales del sistema; luego, los diagramas de secuencia, que describieron la interacción temporal entre los objetos; seguidamente, los diagramas de actividades, que representaron los flujos de procesos y decisiones. Finalmente, se elaboraron los diagramas de clases y objetos, correspondientes a la vista estructural, que mostraron la organización estática del sistema y las relaciones entre sus componentes. Cada ejercicio fue revisado en clase, propiciando retroalimentación formativa y correcciones iterativas hasta alcanzar coherencia entre los requerimientos, los comportamientos modelados y la estructura general del sistema.

Este enfoque progresivo responde a los principios del constructive alignment de Biggs y Tang (2011), al garantizar que los estudiantes no sólo reproduzcan modelos, sino que comprendan su lógica interna y su correspondencia con los procesos del negocio. Asimismo, se apoya en la perspectiva de Booch et al. (2005), quienes sostienen que el modelado orientado a objetos es un medio de pensamiento estructurado más que una simple técnica de documentación.

La tercera estrategia, el trabajo colaborativo y la evaluación por desempeño, fortaleció las competencias comunicativas, éticas y de liderazgo compartido. Los equipos se confor-

maron de manera equilibrada, procurando diversidad de experiencias y niveles de dominio técnico. En cada grupo se designó un líder de coordinación, encargado de organizar las tareas y asegurar la participación de todos, pero sin fragmentar los aprendizajes: cada estudiante debía comprender, documentar y diseñar todos los diagramas en las herramientas utilizadas.

De esta manera, el aprendizaje se mantuvo colectivo e integral, garantizando que todos los miembros desarrollaran competencias en análisis, modelado y argumentación técnica. Las presentaciones finales se realizaron de manera presencial, con exposiciones orales que permitieron evaluar tanto el dominio técnico como la capacidad de comunicar y justificar decisiones.

La evaluación se basó en rúbricas analíticas, diseñadas para valorar la coherencia entre los requerimientos, los modelos y la argumentación del grupo. Este enfoque responde a los principios del aprendizaje cooperativo propuestos por Johnson y Johnson (1999), y a la visión de Villa y Poblete (2007), quienes sostienen que la evaluación por competencias debe integrar desempeño, colaboración y reflexión conjunta.

La integración de estas tres estrategias permitió que los estudiantes transitaran de la comprensión conceptual al dominio aplicado, articulando el saber técnico con el saber hacer y el saber ser. Cada estrategia se complementó con la siguiente: el ABP generó el contexto y la motivación; el modelado con UML brindó las herramientas de representación; y el trabajo colaborativo consolidó la comunicación y la argumentación profesional. En conjunto, conformaron el núcleo del ecosistema estratégico de la experiencia, demostrando que los logros alcanzados no fueron fortuitos, sino producto de una planificación coherente, reflexiva y orientada al desarrollo de competencias integrales. Este núcleo metodológico sirvió, además, como base para las estrategias de soporte que se presentarán en el siguiente apartado, dedicadas a sostener, acompañar y consolidar los aprendizajes logrados durante la implementación.

2.4.3. Estrategias de soporte aplicadas

La implementación de las estrategias núcleo en la experiencia docente de Ingeniería de Software en la Universidad Estatal de Milagro (UNEMI) fue posible gracias a un conjunto de estrategias de soporte institucional, tecnológica y pedagógica que garantizaron la viabilidad del proceso y su coherencia con el modelo educativo. Estas acciones complementarias fueron decisivas para sostener la innovación en el aula y asegurar que las

estrategias principales —aprendizaje basado en problemas reales, modelado progresivo con UML y trabajo colaborativo— se consolidaran como experiencias significativas y sostenibles. Las estrategias de soporte, por tanto, no fueron periféricas, sino el entramado operativo que permitió concretar los resultados de aprendizaje y fortalecer la articulación entre currículo, práctica docente y acompañamiento estudiantil.

Una primera estrategia de soporte fue el uso articulado del Sistema de Gestión Académica (SGA) y del Aula Virtual institucional (Moodle), que constituyen la columna vertebral de la gestión educativa en la UNEMI. Desde el SGA, los docentes planifican la asignatura, cargan el sílabo, los resultados de aprendizaje, las actividades tipo AA (Aprendizaje Autónomo), ACD (Aprendizaje Colaborativo y Dirigido) y APE (Aprendizaje Práctico y Experimental), además del componente práctico y del examen final. Posteriormente, todos estos elementos se migran automáticamente al entorno Moodle, donde se desarrolla el trabajo formativo.

En esta plataforma se alojan los materiales de apoyo —diapositivas, guías, bibliografía de la biblioteca universitaria y recursos de UTECA—, así como las rúbricas de evaluación, los simuladores y las evidencias de las actividades. Esta sincronización entre SGA y Moodle garantiza la trazabilidad del proceso, la transparencia en la evaluación y la continuidad pedagógica, en concordancia con el Modelo Educativo de la Universidad Estatal de Milagro (UNEMI) (2021), que promueve el uso de entornos digitales integrados para favorecer la autonomía y la innovación educativa.

Otra estrategia de soporte fundamental fue la integración de herramientas digitales y metodológicas que enriquecieron la práctica docente. Los salones de la UNEMI cuentan con dispositivos Chromebook, que los estudiantes utilizan para realizar ejercicios de modelado en herramientas como Draw.io y Visual Paradigm, fortaleciendo la comprensión de los diagramas UML desde una práctica activa.

A su vez, se incorporaron aplicaciones como Canva, Padlet, Wayground y Kahoot para el desarrollo de presentaciones interactivas, lluvias de ideas, debates visuales y evaluaciones formativas. Estas herramientas fomentaron la participación, el aprendizaje colaborativo y la creatividad, promoviendo un entorno dinámico y flexible. Siemens (2005) plantea que el aprendizaje en red se basa en la interconexión de nodos de información y en la construcción colectiva del conocimiento; de manera similar, este ecosistema digital permitió que los estudiantes trabajaran en red, compartieran recursos y construyeran saberes de manera conjunta.

Las rúbricas analíticas y las guías de aprendizaje se convirtieron en instrumentos esenciales de soporte metodológico y evaluativo. Cada actividad, ya sea individual o grupal, se acompañó de una rúbrica previamente compartida con los estudiantes, lo que les permitió comprender los criterios de desempeño y autorregular su aprendizaje. Estas rúbricas se utilizaron tanto para la autoevaluación como para la coevaluación, reforzando la transparencia del proceso. Tobón (2013) destaca que la evaluación formativa por competencias requiere instrumentos que articulen desempeño, reflexión y mejora continua; en esta experiencia, las rúbricas cumplieron esa función, convirtiéndose en un elemento transversal de mediación pedagógica.

Otra estrategia de soporte fue el acompañamiento docente constante, tanto dentro como fuera del aula. Durante las sesiones presenciales de tres horas semanales, la docente revisaba los avances de los grupos, orientaba sobre la coherencia entre los requerimientos y los modelos UML, y resolvía dudas específicas sobre el flujo de procesos o la interpretación de los diagramas.

Este acompañamiento se extendía mediante asesorías individuales y retroalimentaciones en el aula virtual. Además, los simuladores institucionales y los talleres prácticos ofrecieron espacios adicionales para fortalecer las habilidades de análisis y diseño de software. Fullan (2007) sostiene que las innovaciones sostenibles se construyen sobre el liderazgo pedagógico y el acompañamiento reflexivo; en este sentido, el acompañamiento continuo fue decisivo para garantizar la calidad y profundidad de los aprendizajes alcanzados.

Finalmente, la experiencia se vio fortalecida por la infraestructura institucional y la suite de herramientas de Google, que facilitaron el trabajo colaborativo y la gestión de proyectos. El acceso a internet estable, los laboratorios de computación y el uso de Google Drive, Docs, Slides, Meet y Forms favorecieron la colaboración asincrónica y sincrónica, el almacenamiento de evidencias y la comunicación fluida entre docente y estudiantes. Kolb (1984) subraya que el aprendizaje experiencial requiere espacios que integren acción, reflexión y conceptualización; los recursos institucionales de UNEMI ofrecieron precisamente esas condiciones, permitiendo un tránsito fluido entre la práctica técnica y la reflexión pedagógica.

En conjunto, las estrategias de soporte implementadas en la experiencia configuraron un ecosistema institucional integrado, donde la tecnología, la organización académica y el acompañamiento docente trabajaron de manera sinérgica. Sin estos apoyos, las estrategias núcleo no habrían tenido la estabilidad ni la proyección necesarias para generar

aprendizajes significativos. La articulación entre SGA, Moodle, herramientas digitales y recursos institucionales consolidó la sostenibilidad del proceso y aseguró la coherencia entre la planificación curricular y la práctica docente. Estas estrategias sentaron las bases para el siguiente nivel del ecosistema estratégico: las estrategias de contingencia, que respondieron a los imprevistos y desafíos surgidos durante la implementación del semestre.

2.4.4. Estrategias de contingencia desplegadas

Toda innovación educativa, por más planificada que esté, se enfrenta a imprevistos que ponen a prueba la flexibilidad del docente y la solidez de la planificación. En el desarrollo de la experiencia en la carrera de Ingeniería de Software de la Universidad Estatal de Milagro (UNEMI), se presentaron diversas contingencias de orden logístico, tecnológico, pedagógico y motivacional. Lejos de representar obstáculos insalvables, estas situaciones se convirtieron en oportunidades de aprendizaje y ajuste, permitiendo fortalecer el ecosistema estratégico del proceso. La capacidad de respuesta ante estas eventualidades fue clave para mantener la coherencia con las competencias y resultados de aprendizaje establecidos en el sílabo, reafirmando que la innovación educativa también se construye desde la adaptabilidad.

Una de las primeras contingencias enfrentadas estuvo relacionada con la gestión del tiempo y la carga de actividades. Las 15 semanas del semestre y las tres horas presenciales por semana resultaban ajustadas para cubrir las fases de análisis de requerimientos, diseño y modelado UML, además de las evaluaciones formativas. Para responder a esta limitación temporal, se implementaron ajustes de secuencia didáctica: se integraron en una misma sesión la exposición teórica, la discusión práctica y la elaboración guiada de diagramas, optimizando el tiempo sin sacrificar la profundidad del aprendizaje.

Asimismo, se amplió el trabajo asincrónico en la plataforma Moodle, permitiendo a los estudiantes desarrollar ejercicios adicionales o cargar entregas fuera del horario presencial. Este uso flexible del aula virtual, articulada al Sistema de Gestión Académica (SGA), fue decisivo para sostener la continuidad del curso y evitar retrasos, evidenciando lo que Zabalza (2003b) denomina currículo flexible, una estructura capaz de adaptarse sin perder coherencia formativa.

Otro tipo de contingencia frecuente fue la limitación tecnológica, especialmente al trabajar con herramientas de modelado. En algunos casos, los estudiantes tuvieron dificultades de conectividad o incompatibilidad al usar Visual Paradigm, una herramienta

más demandante en recursos. Para evitar interrupciones, se aplicó una estrategia de flexibilidad tecnológica: se permitió el uso alternativo de Draw.io o Lucidchart, herramientas más ligeras y accesibles desde los Chromebook institucionales o dispositivos personales.

Además, se fomenta el uso compartido de carpetas en Google Drive, donde los equipos podían guardar versiones en PDF o imágenes de sus diagramas para evitar pérdida de información. Estas acciones garantizaron la continuidad de las prácticas y promovieron el desarrollo de competencias digitales transversales. Según Siemens (2005), la sostenibilidad de los entornos de aprendizaje depende de la capacidad de adaptación y de la migración entre plataformas sin interrumpir el flujo de conocimiento; la experiencia confirma esa premisa en el contexto de la docencia tecnológica universitaria.

También se presentaron contingencias pedagógicas derivadas de la heterogeneidad del grupo. Algunos estudiantes mostraban un dominio avanzado de herramientas digitales, mientras que otros requerían mayor acompañamiento en la comprensión de los procesos de negocio o en la lectura de diagramas. Para equilibrar estos ritmos de aprendizaje, se aplicaron estrategias de mentorías entre pares, donde los estudiantes con más dominio apoyaban a quienes presentaban dificultades, y se implementó la retroalimentación colectiva al cierre de cada clase. Esta dinámica fomentó la colaboración y el aprendizaje horizontal, transformando la diversidad del grupo en un recurso didáctico. Tobón (2013) sostiene que la formación integral exige reconocer la complejidad del contexto y adaptar las estrategias a las condiciones reales de los aprendices, principio que guio las decisiones pedagógicas adoptadas.

La motivación estudiantil también representó un desafío, especialmente en las etapas intermedias del semestre, cuando la carga de trabajo aumentaba y las entregas se acumulaban. Para contrarrestar la desmotivación y el cansancio, se incorporaron estrategias lúdicas y participativas: actividades en Kahoot para reforzar contenidos teóricos, debates en Padlet sobre buenas prácticas de modelado y presentaciones creativas en Canva o Wayground.

Estas acciones revitalizaron el ambiente de aula, fortaleciendo la participación y el compromiso. Kolb (1984) plantea que la motivación se renueva cuando el estudiante puede conectar su experiencia con la aplicación práctica; por ello, se priorizó el análisis de casos reales de la industria tecnológica (bancos, hospitales, urbanizaciones) para reforzar el sentido de pertinencia del aprendizaje.

Finalmente, se adoptaron medidas preventivas frente a posibles brechas logísticas o institucionales, como reprogramaciones por eventos académicos o limitaciones de infra-

estructura. Se mantuvo un plan de contingencia simple pero eficaz: los materiales se cargaban anticipadamente en Moodle, las rúbricas se entregaban desde la primera semana, y se estableció comunicación constante mediante la suite de Google (Gmail y Meet) para coordinar actividades o recuperar sesiones. Fullan (2007) resalta que la sostenibilidad de las innovaciones radica en la capacidad de las instituciones para anticipar, ajustar y mantener el rumbo ante el cambio; este principio se reflejó en la cultura de planificación anticipada que caracterizó la experiencia.

En conjunto, las estrategias de contingencia desplegadas demostraron que la flexibilidad y la gestión oportuna son componentes esenciales del éxito pedagógico. Las respuestas aplicadas —desde la reorganización del tiempo hasta la diversificación de herramientas— no solo mitigaron los imprevistos, sino que fortalecieron la cohesión del ecosistema estratégico. Cada dificultad enfrentada derivó en un aprendizaje institucional y docente: planificar con margen, diversificar recursos y fomentar la resiliencia académica. Estas lecciones se integran al modelo operativo de la experiencia, que en su siguiente fase analizará la arquitectura global del ecosistema, mostrando cómo las estrategias núcleo, de soporte y contingencia conformaron una estructura coherente y sostenible para el logro de las competencias curriculares.

2.4.5. Arquitectura del ecosistema estratégico

La experiencia desarrollada en asignaturas que integran el Análisis y Diseño de Sistemas de Software en la Universidad Estatal de Milagro (UNEMI) puede entenderse como un ecosistema estratégico de aprendizaje, en el que múltiples componentes —pedagógicos, tecnológicos e institucionales— interactuaron de manera articulada para alcanzar los resultados curriculares propuestos. Este ecosistema no fue una suma de acciones aisladas, sino una estructura compleja y dinámica en la que cada tipo de estrategia (núcleo, soporte y contingencia) cumplió una función específica dentro de un mismo sistema operativo. Tal como plantea Morin (2001), los procesos educativos deben comprenderse desde una perspectiva de interdependencia y recursividad, donde las partes se sostienen mutuamente y el todo retroalimenta a sus componentes. En este sentido, la arquitectura del ecosistema permitió que las decisiones pedagógicas se transformaran en un entramado coherente de acción, soporte y adaptación continua.

En el centro de este sistema se ubicaron las estrategias núcleo, que constituyeron el corazón didáctico de la innovación. Estas estrategias —aprendizaje basado en problemas

reales, modelado progresivo con UML y trabajo colaborativo— definieron la orientación metodológica y el sentido pedagógico de la experiencia. Desde ellas se estructuraron las secuencias de aprendizaje, se diseñaron los casos reales y se promovió la construcción de conocimiento a partir de la resolución de situaciones auténticas.

Estas estrategias dieron forma al eje pedagógico central, donde los estudiantes se enfrentaron a desafíos reales, comprendieron los procesos de negocio y tradujeron sus análisis en diagramas de comportamiento y estructura. Según Kolb (1984), la experiencia se consolida como aprendizaje cuando el sujeto actúa, reflexiona y conceptualiza; en este caso, las estrategias núcleo hicieron posible ese tránsito continuo entre acción práctica y comprensión conceptual.

Sosteniendo este núcleo, las estrategias de soporte actuaron como la base institucional, tecnológica y metodológica que garantizó la estabilidad del proceso. La articulación entre el Sistema de Gestión Académica (SGA) y el Aula Virtual Moodle fue clave para la trazabilidad de todas las actividades, desde la planificación del sílabo hasta la entrega de evidencias y la evaluación con rúbricas. El uso de herramientas digitales como Draw.io, Visual Paradigm, Canva, Padlet y Kahoot fortaleció la interacción, la creatividad y la evaluación formativa. Además, la disponibilidad de Chromebooks institucionales, los recursos de la biblioteca y UTECA, y la suite de Google generaron un entorno de aprendizaje híbrido, inclusivo y flexible. Tal como señalan Biggs y Tang (2011), un ecosistema pedagógico logra coherencia cuando los objetivos, las actividades y la evaluación se alinean mediante recursos que aseguren la continuidad del aprendizaje; precisamente ese principio se materializó en el soporte estructural de la experiencia.

Finalmente, las estrategias de contingencia representaron la capa adaptativa del ecosistema, es decir, el mecanismo de autorregulación que permitió mantener el rumbo frente a imprevistos. Estas estrategias respondieron a desafíos de diversa índole —limitaciones de tiempo, conectividad, heterogeneidad del grupo o desmotivación— mediante ajustes flexibles y oportunos.

La reorganización temporal de actividades, la posibilidad de usar alternativas tecnológicas, las mentorías entre pares y las actividades lúdicas fueron recursos que sostuvieron la participación y la calidad del aprendizaje. Siemens (2005) describe los entornos digitales como redes adaptativas donde el conocimiento fluye entre nodos; en esta experiencia, las contingencias cumplieron esa función de red de ajuste y resiliencia, permitiendo que el sistema siguiera funcionando aun ante condiciones variables.

Desde una perspectiva sistémica, la relación entre los tres tipos de estrategias puede

representarse como una arquitectura por capas, similar a la estructura de un sistema de software:

- En la capa base, se encuentran las estrategias de soporte, que equivalen al middleware que sostiene la operación del sistema: recursos, plataformas y acompañamiento institucional.
- Sobre ella se construye la capa lógica o núcleo, donde se ejecutan los procesos principales del aprendizaje, representados por las estrategias didácticas que generan conocimiento y competencias.
- Finalmente, la capa superior de contingencia actúa como el sistema de control y adaptación, encargado de responder a errores, ajustar el flujo y garantizar la estabilidad general.

Esta metáfora técnica, inspirada en la arquitectura de software, permite comprender la experiencia como un sistema educativo en ejecución, en el que cada nivel interactúa y retroalimenta al otro, asegurando el funcionamiento integral del aprendizaje. De acuerdo con Tobón (2013), los procesos formativos deben concebirse como sistemas abiertos, capaces de integrar diversas dimensiones (cognitiva, operativa, emocional e institucional) para alcanzar la formación integral del estudiante; precisamente esta lógica de integración sustentó el ecosistema pedagógico desarrollado.

En síntesis, la arquitectura del ecosistema estratégico permitió visualizar la experiencia como un sistema coherente, sustentable y replicable. La interacción entre estrategias núcleo, de soporte y de contingencia garantizó no solo el logro de los resultados de aprendizaje, sino también la consolidación de un modelo flexible y alineado con el perfil de egreso de la carrera. Esta estructura dinámica mostró que la innovación educativa no depende de acciones aisladas, sino de un entramado de decisiones interconectadas que generan estabilidad y mejora continua.

Tal como advierte Fullan (2007), el cambio educativo duradero se construye cuando las innovaciones logran institucionalizarse en una arquitectura que articula visión, medios y capacidad de adaptación. Este ecosistema, por tanto, constituye la ingeniería didáctica de la experiencia: un sistema vivo que evoluciona aprende y se adapta, manteniendo siempre su foco en la calidad y pertinencia de la formación en Ingeniería de Software.

2.4.6. Integración: Justificación de la validez curricular del ecosistema estratégico

El desarrollo del ecosistema estratégico en la asignatura de Análisis y Diseño de Sistemas de Software permitió consolidar un proceso de enseñanza-aprendizaje coherente con el perfil de egreso de la carrera de Ingeniería de Software de la Universidad Estatal de Milagro (UNEMI). Este ecosistema, compuesto por estrategias núcleo, de soporte y de contingencia, no solo garantizó la articulación entre teoría y práctica, sino que evidenció la pertinencia curricular de la experiencia al contribuir directamente al desarrollo de las competencias profesionales definidas institucionalmente. Tal como sostiene Tobón (2013), la validez de una práctica educativa se confirma cuando los procesos formativos logran articular el “cómo se enseña” con el “para qué se aprende”, generando aprendizajes significativos y verificables.

Entre las competencias curriculares fortalecidas se destacan: el pensamiento crítico y lógico-computacional, el análisis y diseño de sistemas de software, la comunicación técnica y trabajo colaborativo, y la responsabilidad ética en la definición de soluciones tecnológicas. Cada una de ellas fue alcanzada mediante la acción conjunta de las estrategias pedagógicas y operativas que conformaron el ecosistema. El aprendizaje basado en problemas reales (ABP), por ejemplo, permitió que los estudiantes ejercitaran su capacidad de análisis crítico al descomponer los casos propuestos en causas, efectos y posibles soluciones, aplicando el razonamiento lógico para la definición de requerimientos. Según Kolb (1984), el aprendizaje experiencial ocurre cuando los estudiantes reflexionan sobre la práctica y construyen conocimiento desde la acción; en esta experiencia, el ABP se convirtió en un laboratorio cognitivo que fortaleció el razonamiento sistemático y la comprensión integral de los procesos de negocio.

En cuanto a la competencia de análisis y diseño de sistemas de software, las estrategias núcleo de modelado progresivo con UML fueron decisivas. La secuencia de diagramas —casos de uso, secuencia, actividades, clases y objetos— guió a los estudiantes desde la comprensión del problema hasta la representación formal del sistema. Esta progresión les permitió integrar las dimensiones estructurales y de comportamiento, consolidando una visión holística del diseño de software.

La práctica constante con herramientas como Draw.io y Visual Paradigm facilitó el desarrollo de habilidades técnicas y el pensamiento abstracto necesario para representar la realidad mediante modelos formales. El uso de rúbricas analíticas y la retroalimentación

continua aseguraron que los aprendizajes fueran observables y evaluables, en consonancia con el enfoque de alineación constructiva planteado por Biggs y Tang (2011), donde la coherencia entre objetivos, actividades y evaluación constituye el eje de la calidad educativa.

La comunicación técnica y el trabajo colaborativo se fortalecieron mediante la integración de estrategias de soporte y metodologías activas. La estructura del aula virtual —donde se alojaron los materiales, guías y recursos complementarios— y la implementación de herramientas de comunicación sincrónica y asincrónica de la suite de Google promovieron la colaboración en tiempo real y la elaboración conjunta de productos.

Las exposiciones presenciales, sustentadas en presentaciones elaboradas en Canva, Genially y Padlet, favorecieron la argumentación oral y el uso del lenguaje técnico propio de la disciplina. Además, la coevaluación entre pares fomentó la reflexión colectiva sobre los procesos de diseño y documentación, en sintonía con la idea de Siemens (2005), quien sostiene que el conocimiento en entornos digitales se construye en red y se distribuye a través de interacciones colaborativas.

Por otra parte, la responsabilidad ética y profesional emergió como competencia transversal en todas las etapas del proceso. La revisión crítica de los enunciados de casos y la discusión de las implicaciones tecnológicas de las soluciones promovieron la toma de decisiones fundamentadas y conscientes de su impacto. Las contingencias afrontadas —limitaciones tecnológicas, diferencias de ritmo y motivación— se gestionaron desde una ética del cuidado, la cooperación y la equidad, priorizando la inclusión y el aprendizaje compartido. De acuerdo con Fullan (2007), la innovación educativa sostenible se construye sobre la base de valores éticos y compromiso institucional; en este sentido, la experiencia reflejó una cultura docente orientada a la mejora continua y al desarrollo humano.

El conjunto de estrategias implementadas permitió alcanzar evidencias concretas de logro. Entre ellas se destacan: los diagramas UML elaborados y documentados, los informes técnicos, las presentaciones orales, las coevaluaciones registradas en la plataforma Moodle y los proyectos finales de modelado de sistemas reales, que demostraron el dominio de los estudiantes sobre el proceso de análisis y diseño. Estas evidencias, revisadas y evaluadas mediante rúbricas, confirmaron la adquisición de las competencias y la coherencia entre el diseño instruccional y los resultados observados. Como señala Morin (2001), los sistemas educativos deben ser capaces de integrar los saberes teóricos con la

acción práctica; este principio se materializó en la experiencia, donde cada evidencia fue reflejo de una competencia puesta en práctica.

En síntesis, la arquitectura del ecosistema estratégico no solo organizó los recursos y procesos, sino que se constituyó en un modelo pedagógico de pertinencia curricular, demostrando que la enseñanza del análisis y diseño de software puede integrar la rigurosidad técnica con la flexibilidad pedagógica. La interacción entre estrategias núcleo, de soporte y de contingencia posibilitó la formación de profesionales con pensamiento crítico, habilidades de diseño estructurado y capacidad de adaptación a entornos reales. Este cierre confirma que la experiencia docente fue académica, metodológicamente consistente y curricularmente válida, consolidando un camino replicable hacia la enseñanza innovadora de la Ingeniería de Software en la UNEMI.

2.5. Evaluación e indicadores de logro: instrumentos, validez y análisis de evidencias

2.5.1. Transición hacia la evaluación

La evaluación constituye una fase esencial dentro de la sistematización de esta experiencia docente, porque representa el momento en que las estrategias implementadas se someten a verificación y análisis. Tras haber descrito la arquitectura del ecosistema estratégico —con sus estrategias núcleo, de soporte y de contingencia—, resulta necesario demostrar cómo y con qué criterios se comprobó que dichos recursos contribuyeron efectivamente al logro de las competencias curriculares de la carrera de Ingeniería de Software de la Universidad Estatal de Milagro (UNEMI). Este nuevo apartado no busca repetir lo vivido, sino validar los resultados alcanzados, mostrando el rigor metodológico que respalda la experiencia y su valor formativo.

Evaluar fue una necesidad y no un complemento, porque permitió confirmar que las estrategias aplicadas lograron fortalecer competencias clave, tales como el análisis crítico, el pensamiento lógico-computacional, el diseño estructurado de software y la comunicación técnica. Este proceso de evaluación integró múltiples instrumentos e indicadores, diseñados para observar tanto los productos como los procesos: rúbricas analíticas, observaciones directas, talleres prácticos, test de conocimientos, autoevaluaciones y coevaluaciones entre pares. Además, la plataforma institucional Moodle, vinculada al Sistema

de Gestión Académica (SGA), sirvió como medio de registro y verificación de todas las evidencias, garantizando la trazabilidad, la transparencia y la validez de los datos recolectados.

Este módulo se centra, por tanto, en mostrar cómo se evaluó la experiencia y cómo se interpretaron las evidencias obtenidas, con base en criterios de pertinencia, confiabilidad y coherencia curricular. La evaluación se concibe aquí como una herramienta de mejora y reflexión, no solo como medición del logro, sino como proceso formativo que retroalimenta la práctica docente y valida la innovación pedagógica. En las secciones siguientes se detallarán los instrumentos aplicados, los indicadores de evaluación utilizados, los resultados preliminares obtenidos y una reflexión crítica sobre la validez y factibilidad del proceso evaluativo, con el fin de dar al lector una comprensión completa y fundamentada de los alcances de esta experiencia.

2.5.2. Instrumentos de evaluación aplicados

La evaluación de la experiencia docente en las asignaturas relacionadas al Análisis y Diseño de Sistemas de Software se sustentó en un conjunto de instrumentos variados, complementarios y coherentes con el enfoque por competencias. Su finalidad fue comprobar, de manera sistemática, que las estrategias implementadas permitieron alcanzar los resultados de aprendizaje y fortalecer las competencias curriculares definidas por la carrera de Ingeniería de Software de la Universidad Estatal de Milagro (UNEMI). Estos instrumentos se aplicaron en distintos momentos del proceso formativo, combinando componentes diagnósticos, formativos y sumativos, con el objetivo de garantizar una evaluación integral y transparente.

El instrumento central de evaluación fue la rúbrica analítica de desempeño, aplicada en todas las actividades clave del semestre. Cada rúbrica fue diseñada con base en los resultados de aprendizaje definidos en el sílabo y contempló criterios e indicadores específicos para valorar las dimensiones técnica, comunicativa y actitudinal del estudiante. En el plano técnico, las rúbricas evaluaron la coherencia del modelado UML, la identificación correcta de actores y casos de uso, la representación estructural en los diagramas de clases y la precisión en los flujos de procesos.

En la dimensión comunicativa, se valoró la capacidad para explicar decisiones de diseño, argumentar soluciones y sustentar mejoras en tiempo real durante las revisiones orales. Finalmente, en la dimensión actitudinal, se midió la colaboración en equipo, la

responsabilidad frente a las entregas y la ética profesional en el uso de herramientas y fuentes. Estas rúbricas, compartidas previamente en la plataforma Moodle, aseguraron la transparencia del proceso evaluativo y permitieron a los estudiantes autoevaluar su desempeño antes de cada entrega.

Complementando este instrumento principal, se aplicaron cuestionarios de autoevaluación y coevaluación para fortalecer la reflexión crítica y la participación responsable. Los estudiantes valoraron su propio aporte y el de sus compañeros, analizando tanto el desempeño técnico como el compromiso en las tareas grupales.

Estas coevaluaciones, gestionadas mediante formularios en Google Forms, promovieron la conciencia metacognitiva sobre los procesos de aprendizaje y la equidad en la evaluación. Tobón (2013) señala que la autoevaluación y la coevaluación son pilares de la evaluación por competencias, porque fomentan la autorregulación y la corresponsabilidad del estudiante en su formación; en esta experiencia, dichas prácticas consolidaron la madurez académica y el sentido ético de los equipos.

Otro instrumento clave fue el registro de observación docente, aplicado de manera continua durante las sesiones presenciales y en línea. A través de este registro, se documentaron las interacciones entre los grupos, las dificultades técnicas observadas y las mejoras progresivas en la comprensión de los procesos de negocio y del modelado de software.

Estas observaciones, además de servir como evidencia cualitativa, orientaron las retroalimentaciones personalizadas que la docente ofrecía a cada grupo. Según Zabalza (2003a), la observación sistemática del aula es una herramienta de evaluación esencial para comprender las dinámicas de aprendizaje en contextos reales y para ajustar las estrategias en función de la evidencia directa del desempeño.

Asimismo, se aplicaron test diagnósticos y formativos, diseñados en la plataforma Moodle y en herramientas como Kahoot y Wayground, con el fin de medir el dominio conceptual de los estudiantes sobre la ingeniería de requerimientos, los modelos UML y las metodologías de desarrollo. Estos test, realizados en diferentes etapas del semestre, permitieron monitorear el avance del grupo, identificar vacíos conceptuales y reforzar los temas necesarios antes de avanzar hacia el proyecto final. Los resultados cuantitativos obtenidos en estas pruebas complementaron las evidencias cualitativas de los talleres y proyectos, asegurando una visión equilibrada del aprendizaje.

Los talleres prácticos en clase funcionaron, a su vez, como un instrumento de evaluación experiencial. En ellos, los estudiantes resolvían casos reales de análisis y diseño, ge-

neraban árboles de problemas, definían requerimientos y modelaban diagramas, recibiendo retroalimentación inmediata. Cada taller era evaluado mediante una rúbrica específica y registrado en el aula virtual, lo que garantizó la trazabilidad del proceso. Kolb (1984) sostiene que la experiencia directa y la reflexión inmediata son componentes inseparables del aprendizaje significativo; en este sentido, los talleres prácticos no solo sirvieron para aplicar los conocimientos, sino también para observar y evaluar el aprendizaje en acción.

Todos estos instrumentos fueron gestionados, registrados y analizados a través del SGA y Moodle, que actúan como el sistema institucional de evaluación de la UNEMI. Estas plataformas permitieron centralizar las calificaciones, los comentarios y las evidencias de aprendizaje, facilitando el seguimiento longitudinal de los estudiantes. La integración de múltiples instrumentos aseguró que la evaluación fuera triangulada (Yin, 2014), es decir, validada por diferentes fuentes y métodos, lo que aumentó la confiabilidad de los resultados obtenidos.

En conjunto, los instrumentos de evaluación aplicados garantizaron un proceso transparente, inclusivo y coherente con el modelo educativo institucional. Cada herramienta cumplió una función complementaria: las rúbricas aseguraron el rigor técnico, los cuestionarios promovieron la reflexión ética y colaborativa, las observaciones ofrecieron evidencia cualitativa, y los test aportaron medición cuantitativa. Este conjunto permitió obtener una visión integral del desempeño estudiantil y sentó las bases para el análisis de los indicadores de evaluación, que se abordará en el siguiente apartado, donde se mostrará cómo estos instrumentos se tradujeron en criterios observables y medibles del logro de competencias.

2.5.3. Indicadores de evaluación y criterios de validez

Los indicadores de evaluación constituyeron el eje que permitió traducir las evidencias del aprendizaje en resultados verificables, otorgando rigor y credibilidad al proceso. A partir de los instrumentos aplicados —rúbricas, test, talleres, coevaluaciones y registros observacionales—, se definieron indicadores precisos que posibilitaron medir el nivel de logro de las competencias curriculares trabajadas en la experiencia. Estos indicadores se diseñaron con base en los resultados de aprendizaje establecidos en el sílabo de las asignaturas relacionadas al Análisis y Diseño de Sistemas de Software, y se alinearon al enfoque por competencias del Modelo Educativo de la Universidad Estatal de Milagro

(Universidad Estatal de Milagro (UNEMI), 2021), garantizando que la evaluación fuera coherente con la formación profesional del ingeniero de software.

2.5.4. Indicadores aplicados

Los indicadores se estructuraron en correspondencia con las competencias que se buscaban desarrollar: pensamiento lógico-computacional, análisis y diseño de software, comunicación técnica y trabajo colaborativo, y responsabilidad ética. Cada conjunto de indicadores permitió observar un tipo particular de desempeño:

1. Pensamiento lógico-computacional

- Analiza el problema identificando actores, procesos y flujos de información.
- Aplica razonamiento lógico en la elaboración de árboles de problemas y requerimientos.
- Muestra consistencia entre el análisis del problema y la propuesta de solución tecnológica.

Evidencias: talleres en clase, entregas de diagramas de casos de uso y secuencia, ejercicios de requerimientos.

2. Análisis y diseño de sistemas de software

- Representa correctamente los diagramas UML de comportamiento (casos de uso, secuencia, actividades).
- Construye diagramas estructurales (clases y objetos) con relaciones adecuadas y coherencia en los atributos y métodos.
- Argumenta decisiones de modelado y justifica cambios solicitados durante la retroalimentación.

Evidencias: rúbricas de modelado, informes de diseño, presentaciones orales y proyectos finales.

3. Comunicación técnica y trabajo colaborativo

- Expone los resultados del análisis y diseño con vocabulario técnico y claridad conceptual.

- Participa activamente en discusiones grupales y tareas colaborativas.
- Contribuye al trabajo de equipo respetando los acuerdos y tiempos establecidos.

Evidencias: exposiciones presenciales, coevaluaciones, foros de Moodle y presentaciones colaborativas en Canva o Padlet.

4. Responsabilidad ética y profesional

- Entrega puntualmente las actividades y respeta la autoría de sus producciones.
- Evalúa críticamente el impacto de la solución propuesta en el contexto social o institucional.
- Muestra compromiso con la calidad y la mejora continua de su trabajo.

Evidencias: rúbricas de desempeño, informes reflexivos y revisiones del proyecto final.

Cada indicador fue formulado como un criterio observable, lo que permitió registrar desempeños concretos y establecer niveles de logro (insuficiente, básico, competente, destacado). En las rúbricas analíticas, los indicadores funcionaron como descriptores de desempeño, garantizando la transparencia del proceso y la coherencia entre los aprendizajes esperados y las evidencias obtenidas. Como señalan Biggs y Tang (2011), una evaluación de calidad debe basarse en indicadores explícitos que aseguren la alineación entre los objetivos, las actividades y los resultados.

2.5.5. Criterios de validez y confiabilidad

Para asegurar la validez y confiabilidad del proceso evaluativo, se aplicaron cuatro criterios principales:

- Triangulación de fuentes e instrumentos.

Se combinaron diversas evidencias —productos, observaciones y calificaciones— provenientes de distintos instrumentos (rúbricas, test, coevaluaciones, talleres). Este enfoque permitió corroborar la información desde múltiples perspectivas, minimizando sesgos. Flick (2014a) destaca que la triangulación es una estrategia clave para aumentar la credibilidad de los hallazgos en contextos educativos complejos.

- Transparencia y trazabilidad.

Todas las calificaciones, rúbricas y retroalimentaciones fueron registradas en Moodle e importadas al Sistema de Gestión Académica (SGA), lo que garantizó el acceso constante a los datos por parte de los estudiantes. Este registro digital fortaleció la confianza y la posibilidad de auditoría académica, asegurando la validez externa del proceso.

- Coherencia curricular.

Los indicadores se diseñaron directamente a partir de las competencias y resultados de aprendizaje del sílabo, revisando que cada criterio de evaluación respondiera a un objetivo formativo. Este principio, conocido como alineación constructiva (Biggs & Tang, 2011), garantizó que la evaluación midiera exactamente lo que se pretendía enseñar.

- Consistencia interevaluador.

Las rúbricas fueron revisadas con docentes del área para unificar criterios de interpretación y valorar la pertinencia de los niveles de desempeño. Esta revisión conjunta permitió detectar posibles ambigüedades y asegurar uniformidad en la aplicación de los indicadores, aumentando la fiabilidad de los resultados (Yin, 2014).

Estos criterios permitieron validar la calidad del proceso evaluativo y demostrar que los resultados obtenidos fueron consistentes, pertinentes y verificables. La combinación de triangulación, trazabilidad y coherencia curricular fortaleció la transparencia y la credibilidad de la sistematización, ofreciendo al lector una base sólida para comprender los análisis que se presentan en el siguiente apartado.

En síntesis, los indicadores de evaluación aplicados reflejan una práctica docente orientada al desarrollo de competencias integrales y a la evaluación rigurosa del aprendizaje. Los criterios de validez adoptados garantizaron que el proceso no solo midiera logros, sino que generara información confiable para la mejora continua. En el siguiente apartado se presentará el análisis preliminar de las evidencias, donde se interpretarán los resultados obtenidos a partir de estos indicadores y se identificarán los patrones y aprendizajes emergentes.

2.5.6. Análisis preliminar de evidencias

El análisis de evidencias constituyó la etapa que permitió verificar de manera concreta los logros alcanzados por los estudiantes y evaluar la eficacia de las estrategias didácticas implementadas. Las evidencias recolectadas provinieron de múltiples fuentes: rúbricas analíticas, talleres en clase, test formativos, coevaluaciones, observaciones docentes y proyectos finales. La integración de estas fuentes permitió construir una visión completa del proceso formativo, basada en la triangulación de datos cualitativos y cuantitativos (Yin, 2014). Este análisis inicial no busca ofrecer resultados estadísticos definitivos, sino identificar tendencias y patrones emergentes que revelen el impacto pedagógico y curricular de la experiencia.

2.5.7. Tipos de evidencias y organización del análisis

Las rúbricas de evaluación fueron el principal instrumento para la recopilación de datos. Cada calificación se registró en el aula virtual Moodle, donde los criterios e indicadores de desempeño estaban previamente configurados conforme al sílabo institucional. Desde esta plataforma, las calificaciones y comentarios se migraron automáticamente al Sistema de Gestión Académica (SGA), que actúa como repositorio oficial de la evaluación institucional. Esta trazabilidad garantizó transparencia, acceso permanente y coherencia con el modelo educativo de la Universidad Estatal de Milagro (UNEMI).

Para el análisis preliminar, se exportaron los reportes de calificaciones desde Moodle a hojas de cálculo, lo que permitió organizar los resultados por actividad, estudiante e indicador. En este entorno, se aplicaron operaciones básicas de estadística descriptiva (promedios, porcentajes y comparaciones) para identificar patrones de logro y áreas de mejora. Esta práctica sustituyó la elaboración de matrices manuales y aseguró un tratamiento sistemático, automatizado y verificable de los datos.

Paralelamente, las observaciones docentes y las reflexiones derivadas de la retroalimentación se agruparon en categorías vinculadas a los indicadores de desempeño cognitivo, procedural y actitudinal (Tobón, 2013). Esta combinación de resultados cuantitativos (provenientes de Moodle) y cualitativos (derivados de la observación en clase y de los informes reflexivos) fortaleció la triangulación de evidencias, principio clave para la validez del análisis (Flick, 2014).

2.5.8. Hallazgos preliminares

El análisis de las rúbricas de modelado UML evidenció que aproximadamente un 82 % de los estudiantes alcanzaron niveles altos o muy altos de desempeño en la representación de diagramas de casos de uso y secuencia, mostrando dominio del análisis funcional y comprensión del flujo de mensajes entre actores y objetos. En cambio, los diagramas de clases presentaron mayores dificultades, particularmente en la identificación de relaciones de herencia y composición, lo que sugiere la necesidad de reforzar la transición entre el pensamiento lógico y la abstracción estructural.

Las coevaluaciones y autoevaluaciones reflejaron un alto grado de compromiso con el trabajo colaborativo: el 90 % de los estudiantes manifestó sentirse partícipe de las decisiones del grupo, mientras que el 87 % reconoció haber aprendido al analizar las soluciones de sus compañeros. Estos resultados coinciden con lo observado en las sesiones presenciales, donde la exposición de los proyectos fomentó el intercambio técnico y la argumentación profesional.

En las observaciones docentes, se registraron mejoras notables en la comunicación técnica: los estudiantes incorporaron terminología propia del análisis de software y mostraron capacidad para justificar sus decisiones de diseño. Asimismo, se observó una disminución progresiva en los errores de modelado conforme avanzaba el semestre, lo que indica una consolidación gradual del pensamiento lógico-computacional.

Por otro lado, las evidencias actitudinales demostraron que la responsabilidad ética fue una competencia fortalecida a través de la práctica constante. El 100 % de los grupos entregó sus trabajos dentro de los plazos establecidos y respetó la autoría de los materiales presentados. Además, los informes reflexivos mostraron una creciente conciencia sobre el impacto social y técnico del desarrollo de software, en sintonía con el perfil profesional definido por la carrera.

2.5.9. Interpretación de patrones emergentes

El análisis preliminar permitió identificar tres patrones transversales:

1. Progresión cognitiva sostenida, evidenciada en la mejora del razonamiento lógico y la comprensión de procesos de negocio a lo largo del semestre.
2. Fortalecimiento del trabajo colaborativo, que se tradujo en una mayor cohesión de los equipos, comunicación técnica fluida y responsabilidad compartida.

3. Consolidación de la autonomía y la autorregulación, gracias al uso de autoevaluaciones y rúbricas previamente socializadas.

Estos hallazgos reflejan que la evaluación no solo midió resultados, sino que retroalimentó el aprendizaje y transformó las prácticas de los estudiantes, convirtiéndose en un proceso formativo en sí mismo.

2.5.10. Síntesis y proyección

El análisis preliminar de las evidencias confirma que la experiencia contribuyó significativamente al desarrollo de las competencias del perfil de egreso, especialmente en las áreas de análisis, diseño y razonamiento computacional. Los resultados obtenidos muestran una correspondencia directa entre las estrategias didácticas aplicadas, los indicadores definidos y los productos generados por los estudiantes. Este ejercicio analítico sienta las bases para la reflexión crítica del siguiente apartado, en el que se discutirán las implicaciones pedagógicas, los aprendizajes institucionales y las posibles líneas de mejora derivadas de la experiencia.

2.5.11. Reflexión sobre validez, sesgos y factibilidad

El proceso de evaluación desarrollado en esta experiencia se diseñó con criterios de validez, transparencia y coherencia curricular, buscando garantizar que los resultados reflejaran de manera precisa el desempeño y aprendizaje de los estudiantes. Sin embargo, como todo proceso educativo situado, estuvo sujeto a condiciones institucionales, temporales y humanas que es necesario reconocer para dar credibilidad y profundidad analítica al capítulo. Esta reflexión crítica no pretende debilitar la evaluación, sino fortalecer su legitimidad al evidenciar los esfuerzos, decisiones y ajustes que la hicieron posible.

2.5.12. Validez del proceso evaluativo

La validez del proceso se sustentó en la alineación constructiva entre los objetivos, las actividades y la evaluación (Biggs & Tang, 2011). Cada instrumento —rúbricas, test, talleres, coevaluaciones y observaciones— se diseñó directamente a partir de los resultados de aprendizaje definidos en el sílabo de la asignatura, garantizando que se midiera exactamente lo que se pretendía enseñar. Además, la utilización del Moodle y el Sistema

de Gestión Académica (SGA) permitió asegurar la trazabilidad digital de cada evidencia, pues todas las calificaciones y retroalimentaciones quedaron registradas y disponibles tanto para los estudiantes como para los procesos de auditoría académica.

Otro aspecto clave de validez fue la triangulación metodológica (Flick, 2014), al combinar evidencias cuantitativas —resultados numéricos de rúbricas y test— con evidencias cualitativas —observaciones docentes, reflexiones escritas y coevaluaciones—. Esta triangulación permitió confirmar los logros desde diferentes perspectivas, evitando que un solo instrumento definiera la interpretación final. Además, la consistencia interevaluador se fortaleció mediante la revisión cruzada de rúbricas con otros docentes del área, quienes analizaron los criterios y escalas antes de su aplicación, reduciendo así posibles ambigüedades de interpretación (Yin, 2014).

El principio de transparencia fue igualmente determinante para garantizar la validez. Los estudiantes conocieron las rúbricas antes de cada entrega y pudieron autoevaluarse utilizando los mismos indicadores, lo que promovió la corresponsabilidad y la autorregulación del aprendizaje (Tobón, 2013). En este sentido, la validez no solo se entendió como una característica técnica del proceso, sino también como una dimensión ética y pedagógica que asegura justicia y claridad en la evaluación.

2.5.13. Sesgos identificados y estrategias de mitigación

Durante el desarrollo de la experiencia, se identificaron algunos sesgos inherentes al contexto académico, que fueron gestionados mediante acciones correctivas y reflexivas. En primer lugar, existió un sesgo de autoevaluación, ya que algunos estudiantes tendieron a sobrevalorar su desempeño o el de sus compañeros en las coevaluaciones. Para mitigar este efecto, se reforzó la aplicación de retroalimentaciones comparativas, donde la docente contrastaba las percepciones de los estudiantes con la evidencia objetiva de las rúbricas, lo que permitió ajustar calificaciones y fomentar mayor conciencia crítica sobre la autoevaluación.

En segundo lugar, se reconoció un sesgo de selección de evidencias, ya que no todas las producciones de los estudiantes reflejaban de igual manera su progreso real. Por ejemplo, algunos trabajos sobresalían por su presentación formal más que por su contenido técnico. Para reducir este riesgo, se priorizó la evaluación de los procesos, observando la evolución del estudiante a través de varias entregas en lugar de juzgar solo el producto

final. Este enfoque procesual permitió valorar el aprendizaje como una trayectoria y no como un resultado puntual (Stiggins, 2005).

Otro posible sesgo estuvo vinculado a las condiciones institucionales y tecnológicas. En ocasiones, la conectividad o la disponibilidad de recursos digitales afectó la realización de test o la carga de evidencias en Moodle. Para resolverlo, se implementaron estrategias de flexibilización evaluativa, ampliando los plazos de entrega o habilitando medios alternativos de envío (correo institucional, presentaciones presenciales o envío directo al SGA). Esta medida no solo garantizó la equidad, sino que reflejó la adaptabilidad del proceso evaluativo ante contingencias técnicas.

Finalmente, se detectó un sesgo temporal relacionado con la carga de trabajo docente. El seguimiento personalizado, la corrección de diagramas UML y la revisión de coevaluaciones demandaron una inversión considerable de tiempo fuera del aula. Para manejarlo, se utilizó una estrategia de retroalimentación formativa escalonada, centrando los comentarios en los indicadores esenciales y reservando las revisiones extensas para el proyecto final. Esta práctica permitió mantener la calidad del acompañamiento sin comprometer la factibilidad operativa del proceso.

2.5.14. Factibilidad y aprendizajes derivados

La factibilidad de la evaluación estuvo estrechamente ligada a las condiciones institucionales y tecnológicas de la UNEMI. Contar con el ecosistema digital integrado (Moodle–SGA–Google Workspace) facilitó el seguimiento continuo, la retroalimentación y la generación de reportes consolidados. Sin embargo, esta factibilidad también dependió de la competencia digital docente y estudiantil, pues algunos estudiantes al inicio del semestre requirieron apoyo para el manejo de las plataformas. Se ofrecieron sesiones breves de orientación digital, lo que mejoró la participación y redujo la dependencia técnica.

En el plano operativo, la evaluación resultó viable gracias a la existencia de rúbricas estandarizadas, que aceleraron el proceso de calificación y permitieron mantener criterios homogéneos a lo largo del semestre. No obstante, se reconoce que la gestión simultánea de múltiples instrumentos —rúbricas, test, talleres y observaciones— demandó una planificación rigurosa y un equilibrio entre la carga académica y administrativa. Este desafío deja como aprendizaje la necesidad de automatizar ciertos procesos de retroalimentación y continuar aprovechando las herramientas de análisis disponibles en Moodle y Google Sheets para optimizar la gestión del tiempo docente.

2.5.15. Síntesis y proyección

En síntesis, la reflexión sobre validez, sesgos y factibilidad confirma que el proceso evaluativo de esta experiencia fue sólido, ético y contextualizado, pero no exento de limitaciones. La validez se aseguró mediante la triangulación, la coherencia curricular y la transparencia, mientras que los sesgos fueron reconocidos y mitigados con acciones correctivas. La factibilidad, por su parte, se sostuvo gracias a la infraestructura institucional, aunque evidenció la necesidad de seguir fortaleciendo la formación digital y la planificación temporal del docente.

Reconocer estos aspectos otorga credibilidad a la sistematización y ofrece aprendizajes transferibles a otros escenarios educativos. La evaluación, entendida no como un acto de control, sino como un proceso reflexivo y perfectible, se convierte en una herramienta de mejora continua que alimenta la práctica docente y la calidad académica institucional. El siguiente apartado integrará estos hallazgos para mostrar cómo la evaluación consolida los resultados y proyecta líneas de innovación educativa sostenibles.

2.5.16. Integración: Síntesis de la evaluación

El proceso de evaluación desarrollado en esta experiencia no fue un trámite administrativo ni un requisito técnico, sino un componente formativo y verificable que confirmó la pertinencia curricular, la coherencia metodológica y la validez académica de la innovación docente implementada en la asignatura relacionada al Análisis y Diseño de Sistemas de Software. A lo largo del módulo se evidenció que la evaluación, entendida desde un enfoque de competencias, constituyó un eje articulador que permitió medir los aprendizajes alcanzados, retroalimentar los procesos y consolidar los resultados de manera transparente y fundamentada.

La integración de los instrumentos, indicadores y evidencias mostró una evaluación equilibrada entre dimensiones técnicas, cognitivas y actitudinales, en consonancia con el Modelo Educativo de la Universidad Estatal de Milagro (Universidad Estatal de Milagro (UNEMI), 2021). Las rúbricas analíticas sirvieron como principal herramienta de valoración, garantizando criterios claros y niveles de logro observables; los test formativos y cuestionarios permitieron monitorear el progreso conceptual de los estudiantes; las observaciones docentes y coevaluaciones aportaron información cualitativa sobre el desempeño colaborativo y la comunicación técnica; mientras que las entregas prácticas en Moodle y proyectos finales ofrecieron evidencias concretas del pensamiento lógico-computacional

y de la capacidad de diseñar soluciones tecnológicas coherentes. Esta diversidad de instrumentos dio soporte a un proceso de evaluación integral y triangulado, coherente con la estructura curricular de la carrera de Ingeniería de Software.

Los indicadores de evaluación —definidos a partir de las competencias del perfil de egreso— permitieron medir con precisión el grado de desarrollo del pensamiento analítico, la capacidad de modelar sistemas, la comunicación técnica y la responsabilidad profesional. La sistematización de las calificaciones en Moodle, posteriormente migradas al Sistema de Gestión Académica (SGA), facilitó la generación de reportes y análisis cuantitativos que mostraron tendencias claras de avance en el desempeño. Por ejemplo, más del 80 % de los estudiantes alcanzó niveles altos de logro en los diagramas de comportamiento (casos de uso, secuencia y actividades), mientras que los diagramas estructurales (clases y objetos) reflejaron un progreso más gradual, evidenciando la necesidad de reforzar la comprensión entre análisis funcional y diseño estructural.

El análisis de las evidencias cualitativas complementó estos resultados numéricos, mostrando mejoras sostenidas en la argumentación técnica y en la apropiación del lenguaje disciplinar. Los registros de observación indicaron que los estudiantes no solo aprendieron a modelar, sino también a reflexionar sobre la lógica de los procesos, a justificar decisiones de diseño y a trabajar de manera colaborativa, fortaleciendo así competencias transversales como el pensamiento crítico, la comunicación y la autonomía. La evaluación se consolidó así como un proceso reflexivo y formativo, que acompañó al aprendizaje y no como un juicio final sobre los resultados.

En la revisión crítica sobre validez, sesgos y factibilidad se confirmó que el proceso evaluativo fue consistente, transparente y contextualizado, aunque no exento de desafíos. Los principales sesgos identificados —autoestima evaluativa en coevaluaciones, condiciones tecnológicas desiguales y limitaciones temporales del docente— fueron mitigados mediante estrategias de ajuste y flexibilidad, como la triangulación de fuentes, la retroalimentación comparativa y el uso de herramientas digitales alternativas. Estos aprendizajes evidencian que la validez en la evaluación no depende de la perfección de los instrumentos, sino de la capacidad docente para interpretar y ajustar con criterio pedagógico los resultados obtenidos (Flick, 2014a); (Stiggins, 2005).

En síntesis, la evaluación permitió verificar el logro efectivo de las competencias curriculares y confirmar la relevancia formativa de la experiencia. Se consolidaron especialmente las competencias de pensamiento lógico-computacional, análisis y diseño de sistemas de software, comunicación técnica y responsabilidad ética. Estas competencias

fueron observadas y validadas a través de múltiples evidencias, lo que otorga confiabilidad a los resultados y legitimidad académica al capítulo. Asimismo, la experiencia evaluativa aportó aprendizajes institucionales sobre cómo utilizar el ecosistema digital de la UNEMI —Moodle, SGA y Google Workspace— como soporte integral de la gestión académica y de la mejora continua del proceso docente.

Este cierre reafirma que la evaluación no es el final del proceso, sino el puente hacia la reflexión crítica y la transferencia, que constituyen el foco del siguiente módulo. Evaluar significó comprender, evidenciar y aprender del propio proceso de enseñanza, demostrando que la sistematización de una práctica educativa innovadora solo adquiere sentido cuando permite retroalimentar la docencia, fortalecer el currículo y generar conocimiento útil para otros contextos académicos.

2.6. Del problema a la solución: estrategias para enseñar análisis y diseño de software en primer nivel

2.6.1. Transición hacia la reflexión final

Con la evaluación de la experiencia se alcanzó una comprensión integral de los logros, limitaciones y evidencias que sustentan el impacto formativo del proceso. Sin embargo, evaluar no agota el sentido de la sistematización; lo verdaderamente transformador ocurre cuando el docente reflexiona críticamente sobre lo vivido, identifica los aprendizajes construidos y reconoce el valor de compartir su experiencia con otros. Este momento marca el paso de la comprobación al entendimiento, de la evidencia empírica a la interpretación pedagógica, y abre el espacio para pensar en la transferencia y sostenibilidad de la innovación educativa implementada.

Esta transición invita a mirar la experiencia no solo como un conjunto de estrategias exitosas, sino como un proceso de aprendizaje profesional que transformó la práctica docente, la interacción con los estudiantes y la comprensión del currículo. Reflexionar críticamente implica preguntarse por el sentido de cada decisión, por las condiciones que hicieron posible el logro de las competencias y por las nuevas preguntas que el proceso deja abiertas. Como afirma Schön (1992), la docencia se fortalece cuando el profesional se convierte en un “practicante reflexivo”, capaz de analizar su propia acción y producir conocimiento desde ella.

Más allá de los indicadores y resultados medibles, este cierre se orienta a comprender qué significó la experiencia en términos humanos, pedagógicos e institucionales, y cómo puede proyectarse hacia otros espacios de la universidad o incluso de la comunidad educativa en general. La sistematización alcanza aquí su mayor valor: el de convertirse en un conocimiento compatible y transferible que aporta a la mejora de la enseñanza en Ingeniería de Software y a la innovación docente en la UNEMI.

En las secciones siguientes se presentarán, primero, los aprendizajes docentes e institucionales que surgieron a partir de la experiencia; luego, las posibilidades de transferencia y sostenibilidad; y finalmente, una reflexión de cierre que integrará el sentido, la validez y el legado de todo el proceso desarrollado.

2.6.2. Reflexión crítica sobre la experiencia

La sistematización de esta experiencia docente permitió no solo evidenciar los resultados alcanzados por los estudiantes, sino también comprender el valor pedagógico, institucional y humano que se generó a lo largo del proceso. Reflexionar críticamente sobre la práctica significó mirar más allá de los datos y los indicadores, para entender cómo las decisiones metodológicas, las estrategias empleadas y las condiciones institucionales contribuyeron a transformar la enseñanza en el campo de la Ingeniería de Software. Tal como plantea Schön (1992), la reflexión sobre la acción convierte la práctica en fuente de conocimiento profesional, permitiendo al docente analizar las tensiones, los aprendizajes y los cambios que emergen en su ejercicio formativo.

2.6.3. Aportes de la experiencia

Uno de los aportes más relevantes de la experiencia fue la integración efectiva de la teoría y la práctica a través de un enfoque basado en problemas reales. Los estudiantes no se limitaron a reproducir modelos teóricos, sino que diseñaron soluciones tecnológicas contextualizadas, abordando casos inspirados en sistemas académicos, hospitalarios o logísticos. Este enfoque permitió desarrollar competencias clave, como el pensamiento lógico-computacional, la capacidad de análisis de procesos y el diseño estructurado de sistemas, que se reflejaron en los diagramas UML, la definición de requerimientos y la argumentación técnica de sus proyectos. Además, el uso de herramientas digitales como draw.io, Visual Paradigm, Moodle y Wayground fortaleció la alfabetización tecnológica y fomentó la autonomía en el aprendizaje.

En el plano pedagógico, la experiencia consolidó una didáctica centrada en el aprendizaje activo y la evaluación formativa, sustentada en el uso de rúbricas, autoevaluaciones y retroalimentaciones continuas. Esta metodología transformó la dinámica del aula, promoviendo la participación y la reflexión crítica de los estudiantes sobre su propio proceso. En términos institucionales, el trabajo aportó al fortalecimiento de la cultura evaluativa de la UNEMI, demostrando que el ecosistema digital (SGA–Moodle–Google Workspace) puede ser un soporte eficaz para procesos de enseñanza-aprendizaje transparentes y de alta trazabilidad académica.

2.6.4. Tensiones y resistencias encontradas

Como toda innovación educativa, la experiencia enfrentó tensiones estructurales y operativas. Una de las principales fue el desafío de equilibrar el tiempo docente con la complejidad de la evaluación continua. La aplicación de múltiples instrumentos (rúbricas, observaciones, test y coevaluaciones) demandó una carga significativa de seguimiento y retroalimentación. Aunque esto enriqueció la evaluación, también puso en evidencia la necesidad de automatizar ciertos procesos de revisión mediante herramientas digitales que reduzcan el esfuerzo administrativo sin sacrificar la calidad del acompañamiento pedagógico.

Otra tensión se relacionó con las diferencias en los niveles de competencia digital de los estudiantes, especialmente en las primeras semanas. Algunos mostraron inseguridad frente al uso de plataformas o herramientas de modelado. Esta brecha se abordó mediante tutorías rápidas y explicaciones en clase, lo que permitió nivelar gradualmente el grupo, pero también evidenció la importancia de mantener una política institucional de alfabetización digital continua.

A nivel metodológico, surgieron resistencias iniciales frente al enfoque de aprendizaje colaborativo y basado en problemas. Algunos estudiantes manifestaron preferencia por métodos tradicionales más directivos, acostumbrados a una docencia centrada en la exposición teórica. Sin embargo, con el avance del semestre y la demostración de los resultados, esta resistencia se transformó en una valoración positiva del trabajo cooperativo y de la aplicación práctica de los conceptos. Como sugiere Fullan (2007), las resistencias no son un obstáculo sino un indicador del cambio, ya que toda innovación educativa implica una reconfiguración de roles, rutinas y expectativas.

Finalmente, una tensión persistente fue la disparidad entre la evaluación formativa y la evaluación institucional sumativa, ya que los tiempos del semestre y las normativas de cierre no siempre se ajustan al ritmo reflexivo del aprendizaje por competencias. Esto llevó a realizar ajustes en la ponderación de las actividades y a fortalecer la comunicación con la coordinación académica para asegurar que los resultados formativos tuvieran el mismo reconocimiento que las evaluaciones finales.

2.6.5. Aprendizajes personales, colectivos e institucionales

En el plano personal, este proceso reafirmó la convicción de que enseñar ingeniería de software implica acompañar procesos de pensamiento, no solo enseñar herramientas o diagramas. La docencia se transformó en un espacio de reflexión continua, donde cada clase fue una oportunidad para reinterpretar los métodos, adaptar los recursos y repensar las formas de guiar el aprendizaje. La experiencia confirmó que el aprendizaje significativo ocurre cuando el estudiante comprende el problema antes de buscar la solución, y cuando el docente actúa como mediador entre la lógica técnica y la comprensión pedagógica del proceso.

Colectivamente, los aprendizajes fueron múltiples. Los estudiantes desarrollaron mayor autonomía, capacidad de argumentación y trabajo colaborativo, fortaleciendo sus habilidades para el análisis de requerimientos, la comunicación técnica y la toma de decisiones compartidas. También se evidenció un fortalecimiento del sentido ético profesional, expresado en la responsabilidad y originalidad de los proyectos entregados. La práctica del trabajo en grupos heterogéneos, la coevaluación y la exposición oral permitió que cada integrante asumiera un rol activo en la construcción del conocimiento y en la validación de los resultados.

A nivel institucional, la experiencia aportó aprendizajes significativos sobre la integración de los sistemas tecnológicos en los procesos académicos. La gestión coordinada entre Moodle, el SGA y los recursos de Google Workspace mostró que la digitalización educativa puede favorecer la transparencia, la eficiencia y la trazabilidad de los procesos evaluativos. Asimismo, se fortaleció la cultura de trabajo docente colaborativo: compartir rúbricas, revisar criterios y ajustar prácticas se convirtió en un ejercicio colectivo de mejora. Como indica Bolívar (2012), la innovación pedagógica solo puede consolidarse cuando se comparte y se discute, porque “la cultura institucional del cambio se construye en diálogo con las prácticas de los otros”.

2.6.6. Síntesis reflexiva y proyección

En síntesis, la experiencia no solo permitió alcanzar los objetivos curriculares previstos, sino también redefinir la práctica docente desde una perspectiva reflexiva, ética y contextualizada. Los logros alcanzados —mejor comprensión del análisis de sistemas, desarrollo del pensamiento lógico-computacional, fortalecimiento del trabajo colaborativo y dominio del modelado UML— se acompañaron de aprendizajes humanos igualmente valiosos: la empatía, la paciencia, la adaptabilidad y la conciencia del impacto social del quehacer tecnológico.

Las tensiones experimentadas se transformaron en fuentes de aprendizaje profesional, recordando que la innovación educativa es un proceso en construcción permanente. Esta reflexión deja en claro que evaluar, reflexionar y transferir son momentos de un mismo ciclo formativo, y que su continuidad depende de mantener abierta la posibilidad de mejorar.

La sistematización, en su conjunto, evidenció que la práctica docente puede ser una fuente legítima de producción de conocimiento educativo, capaz de dialogar con la teoría y aportar a la mejora de la enseñanza en contextos reales. Este ejercicio abre la puerta a la transferencia: a compartir, replicar y adaptar la experiencia en otros espacios de formación en ingeniería y en carreras afines, fortaleciendo el compromiso institucional con una educación superior de calidad e innovación sostenida.

2.6.7. Integración final: Reflexión y transferencia

La sistematización de esta experiencia ha sido, ante todo, un proceso de transformación profesional y académica. A lo largo de los seis módulos, se construyó una comprensión integral del ciclo docente: desde la fundamentación conceptual y curricular, pasando por la planificación estratégica y la evaluación, hasta llegar a esta etapa final de reflexión crítica y proyección. Este recorrido permitió comprender que toda práctica educativa significativa trasciende su espacio inmediato cuando logra generar conocimiento comparable, capaz de inspirar nuevas formas de enseñar, aprender y evaluar.

En el ámbito de la docencia universitaria, la experiencia desarrollada de las asignaturas relacionadas al Análisis y Diseño de Sistemas de Software demostró que la enseñanza en ingeniería puede ser profundamente pedagógica y humana, sin perder su rigor técnico. Integrar el análisis de problemas reales, la ingeniería de requerimientos, el modelado UML y la argumentación técnica en un solo marco metodológico permitió a los estudiant-

tes aprender haciendo, pero, sobre todo, pensando críticamente. Este proceso transformó el aula en un laboratorio de innovación, donde los conceptos se aplicaron a contextos reales, las ideas se debatieron colectivamente y las soluciones se construyeron de forma colaborativa.

En términos institucionales, la experiencia consolidó una cultura de evaluación formativa y transparencia académica, apoyada en el ecosistema tecnológico de la UNEMI. El uso articulado del Sistema de Gestión Académica (SGA) y Moodle permitió garantizar la trazabilidad, la equidad y la confiabilidad de los procesos evaluativos. Este logro no solo fortaleció la gestión docente, sino que también demostró que la digitalización educativa puede ser una aliada de la innovación didáctica, siempre que esté acompañada de reflexión pedagógica y acompañamiento continuo.

2.6.8. Aportes y aprendizajes globales

El aprendizaje más profundo que deja este proceso es la conciencia del valor formativo de la reflexión. Evaluar y sistematizar no fueron solo etapas del proyecto, sino oportunidades para repensar la enseñanza como un proceso de investigación y mejora continua. En el plano personal, el ejercicio confirmó que la innovación educativa no consiste únicamente en incorporar nuevas herramientas o metodologías, sino en revisar críticamente las propias prácticas y reconocer en ellas oportunidades de transformación. Como señala Jara (2018a), sistematizar es “volver a mirar” lo vivido, descubrir el sentido de la práctica y compartirlo con otros para generar conocimiento pedagógico colectivo.

A nivel colectivo, el proceso reafirmó que el trabajo colaborativo entre estudiantes, docentes y la institución produce aprendizajes sostenibles. Los estudiantes fortalecieron su capacidad de análisis, argumentación y comunicación técnica, mientras que la comunidad docente de la carrera avanzó hacia un enfoque más integral de la evaluación por competencias. Asimismo, se consolidó una red de aprendizaje compartido dentro de la Facultad de Ingeniería, donde las experiencias y rúbricas creadas para esta práctica comenzaron a ser adaptadas por otros docentes. Esta transferencia horizontal constituye uno de los mayores aportes del proceso: convertir la innovación individual en práctica institucional compartida.

2.6.9. Tensiones, desafíos y aprendizajes emergentes

Reflexionar también implicó reconocer las tensiones estructurales y humanas que acompañan toda innovación. Entre ellas, la gestión del tiempo, la carga administrativa y la diversidad de ritmos de aprendizaje de los estudiantes. Sin embargo, estas dificultades se transformaron en aprendizajes sobre la importancia de la planificación, la flexibilidad pedagógica y la empatía. Como sostiene Fullan (2007), el cambio educativo no ocurre sin resistencia, y su sostenibilidad depende de la capacidad institucional para acompañar los procesos y no solo evaluarlos.

Otro aprendizaje clave fue comprender que la docencia reflexiva requiere espacios de formación permanente. La implementación de estrategias como el Aprendizaje Basado en Problemas (ABP) y la evaluación por competencias evidenció la necesidad de fortalecer el desarrollo profesional docente y la integración de comunidades de práctica que permitan compartir estrategias, materiales y resultados. Esta experiencia demostró que la innovación no es un acto aislado, sino una construcción colectiva que debe institucionalizarse mediante políticas de acompañamiento, capacitación y reconocimiento académico.

2.6.10. Proyección y transferencia

El principal aporte de esta experiencia radica en su potencial de transferencia. Los resultados alcanzados y las estrategias aplicadas pueden ser replicados o adaptados en otras asignaturas de la carrera de Ingeniería de Software y en programas afines, especialmente aquellos que integran componentes de diseño, análisis de sistemas o trabajo colaborativo. Las metodologías utilizadas —aprendizaje basado en problemas, modelado incremental, evaluación formativa con rúbricas y reflexión metacognitiva— pueden constituir la base de un modelo pedagógico replicable, adaptable a diferentes contextos curriculares.

Asimismo, la experiencia ofrece insumos para el fortalecimiento de la formación docente en la UNEMI. Los aprendizajes sobre planificación, acompañamiento y evaluación formativa pueden incorporarse en talleres de actualización pedagógica o en la formación de nuevos docentes, promoviendo una cultura de docencia reflexiva y basada en evidencias. Esta proyección es coherente con los lineamientos del Modelo Educativo de la Universidad Estatal de Milagro (UNEMI) (2021), que concibe la enseñanza como un proceso integrador entre investigación, práctica y reflexión.

En el ámbito más amplio de la educación en ingeniería, esta experiencia muestra que la enseñanza técnica puede dialogar con la pedagogía crítica. Modelar un sistema in-

formático no es solo un ejercicio lógico, sino también una práctica social que implica comprender necesidades, diseñar soluciones y reflexionar sobre su impacto. Transferir este enfoque significa enseñar a los futuros ingenieros no solo a codificar sistemas, sino a pensar éticamente y diseñar con propósito.

2.6.11. Cierre del capítulo

Culminar este proceso de sistematización representa más que el cierre de un ciclo; simboliza la consolidación de una mirada docente basada en la reflexión, la evidencia y la mejora continua. Esta experiencia permitió confirmar competencias curriculares, desarrollar innovación pedagógica y fortalecer la identidad profesional docente, reafirmando que la educación universitaria debe sostenerse en la articulación entre teoría, práctica y crítica.

El camino recorrido deja aprendizajes profundos: la importancia de planificar con sentido, evaluar con transparencia, retroalimentar con empatía y reflexionar con honestidad. Cada módulo fue un paso hacia la comprensión de que enseñar también es investigar y que la práctica docente, cuando se documenta y se analiza, se convierte en un acto de construcción de conocimiento.

El capítulo se cierra con la certeza de que esta experiencia no termina aquí: puede y debe ser compartida, adaptada y replicada en nuevos escenarios educativos. Su valor no radica solo en los logros alcanzados, sino en haber demostrado que la innovación docente es posible cuando se une la rigurosidad técnica de la ingeniería con la sensibilidad pedagógica de la educación.

Bibliografía

- ABET. (2020). *Criteria for Accrediting Computing Programs*. Accreditation Board for Engineering; Technology.
- ACM & IEEE. (2020). *Curriculum Guidelines for Undergraduate Programs in Software Engineering (SE2020)*. ACM/IEEE Joint Task Force.
- Barnett, R. (2001). *Los límites de la competencia: El conocimiento, la educación superior y la sociedad*. Gedisa.
- Barrows, H. (1986). A taxonomy of problem-based learning methods. *Medical Education*, 20(6), 481-486.
- Biggs, J., & Tang, C. (2011). *Teaching for Quality Learning at University* (4.^a ed.). McGraw-Hill Education.
- Bonwell, C. C., & Eison, J. A. (1991). *Active Learning: Creating Excitement in the Classroom*. George Washington University.
- Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The Unified Modeling Language User Guide* (2.^a ed.). Addison-Wesley.
- Carlino, P. (2005). *Escribir, leer y aprender en la universidad: Una introducción a la alfabetización académica*. Fondo de Cultura Económica.
- Elliott, J. (1993). *El cambio educativo desde la investigación-acción*. Morata.
- Flick, U. (2014a). *Introducción a la investigación cualitativa* (5.^a ed.). Morata.
- Flick, U. (2014b). *La gestión de la calidad en investigación cualitativa*. Ediciones Morata.
- Fullan, M. (2007). *The New Meaning of Educational Change* (4.^a ed.). Teachers College Press.
- Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38-43. <https://doi.org/10.3102/0013189x12463051>
- Hyland, K. (2009). *Academic Discourse: English in a Global Context*. Continuum.
- Jara, O. (2018a). *La sistematización de experiencias: Aprendizajes y desafíos para la educación popular*. Alforja.
- Jara, O. (2018b). *La sistematización de experiencias: Práctica y teoría para otros mundos posibles*. Alforja.
- Johnson, D. W., & Johnson, R. T. (1999). *Learning Together and Alone: Cooperative, Competitive, and Individualistic Learning*. Allyn & Bacon.

- Kolb, D. A. (1984). *Experiential Learning: Experience as the Source of Learning and Development*. Prentice Hall.
- Morin, E. (2001). *Los siete saberes necesarios para la educación del futuro*. UNESCO.
- Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* (9.^a ed.). McGraw-Hill Education.
- Prince, M. (2004). Does Active Learning Work? A Review of the Research. *Journal of Engineering Education*, 93(3), 223-231. <https://doi.org/10.1002/j.2168-9830.2004.tb00809.x>
- Schön, D. (1992). *La formación de profesionales reflexivos*. Paidós.
- Siemens, G. (2005). Connectivism: A Learning Theory for the Digital Age. *International Journal of Instructional Technology and Distance Learning*, 2(1).
- Stake, R. (1995). *The Art of Case Study Research*. SAGE Publications.
- Stenhouse, L. (1984). *Investigación y desarrollo del currículo*. Morata.
- Stenhouse, L. (1987). *La investigación como base de la enseñanza*. Morata.
- Stiggins, R. (2005). *Student-Involved Assessment for Learning* (4.^a ed.). Merrill/Prentice Hall.
- Tobón, S. (2013). *Formación integral y competencias: Pensamiento complejo, currículo, didáctica y evaluación*. ECOE Ediciones.
- Universidad Estatal de Milagro (UNEMI). (2021). Modelo Educativo de la Universidad Estatal de Milagro.
- Villa, A., & Poblete, M. (2007). *Aprendizaje basado en competencias: Una propuesta para la evaluación de las competencias genéricas*. Ediciones Mensajero.
- Wenger, E. (1998). *Communities of Practice: Learning, Meaning, and Identity*. Cambridge University Press.
- Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33-36. <https://doi.org/10.1145/1118178.1118215>
- Yin, R. K. (2014). *Case Study Research: Design and Methods* (5.^a ed.). SAGE Publications.
- Zabalza, M. A. (2003a). *Competencias docentes del profesorado universitario: Calidad y desarrollo profesional*. Narcea Ediciones.
- Zabalza, M. A. (2003b). *Diseño y desarrollo curricular*. Narcea Ediciones.

3

La enseñanza del código limpio: estrategias para formar desarrolladores con estándares profesionales

Jessica Janina Cabezas Quinto ³

El Capítulo 3 concibe la enseñanza universitaria como construcción colectiva de conocimiento, donde la práctica docente es fuente de reflexión y transformación institucional. Con base en experiencias del programa RISEI, analiza tensiones entre teoría y práctica y los retos de integrar investigación educativa en la labor cotidiana. Destaca el acompañamiento pedagógico, la colaboración interdisciplinaria y las comunidades de aprendizaje para impulsar innovación didáctica. Examina situaciones reales de aula, mediación docente y evaluación formativa que promueven participación activa y mejores resultados. Resalta la sistematización como herramienta para reconstruir sentido y transferir aprendizajes. Propone una visión crítica basada en investigación-acción, ética y mejora continua.

³Universidad Estatal de Milagro, jcabezasq2@unemi.edu.ec.

Índice

3.1. Buenas prácticas de programación y refactorización	114
3.1.1. Contextualización de la experiencia pedagógica	114
3.1.2. Identificación del problema pedagógico: hacia las buenas prácticas de programación y refactorización	115
3.1.3. El rol del docente como revisor de código pedagógico	116
3.1.4. Cierre integrador	117
3.2. Del código al pensamiento: enseñanza del código limpio en la formación inicial de ingenieros en software	117
3.2.1. Bisagra Textual	118
3.2.2. Identificación de conceptos estructurantes	118
3.2.3. Formulación de dimensiones	119
3.2.4. Construcción de indicadores	120
3.2.5. Dimensión Cognitivo–Conceptual: Comprensión de la lógica y la abstracción algorítmica	121
3.2.6. Dimensión Técnico–Procedimental: Aplicación de buenas prácticas y calidad del código	122
3.2.7. Fuentes y métodos de verificación	123
3.2.8. Modelar y abstraer procesos lógicos	124
3.2.9. Comprensión de la secuencia algorítmica	124
3.2.10. Aprendizaje significativo y contextualizado	125
3.2.11. Fuentes complementarias y métodos asociados	125
3.2.12. Síntesis integradora	125
3.2.13. Justificación teórica del conjunto	126
3.2.14. Justificación de los indicadores	127
3.2.15. Justificación de las fuentes y métodos de verificación	127
3.2.16. Síntesis final del conjunto	128
3.2.17. Recomendaciones para profundizar	129

3.3. Aprender gestionando: experiencias innovadoras en proyectos de software universitarios	129
3.3.1. Transición al vínculo curricular	129
3.3.2. Identificación de competencias del perfil	130
3.3.3. Resultados de aprendizaje vinculados	133
3.3.4. Actividades y evidencias	135
3.3.5. Reflexión sobre la alineación curricular	137
3.4. Transición hacia la operacionalización estratégica	139
3.4.1. Recomendaciones para profundizar	139
3.4.2. Clase 1: Estrategias núcleo en acción	140
3.4.3. Clase 2: Estrategias de soporte aplicadas	143
3.4.4. Clase 3: Estrategias de contingencia desplegadas	146
3.4.5. Clase 4: Arquitectura del ecosistema estratégico	149
3.4.6. Síntesis final: El ecosistema como sistema vivo	151
3.5. Transición hacia la evaluación	151
3.5.1. Recomendaciones para profundizar	152
3.5.2. Clase 1: Instrumentos de evaluación aplicados	152
3.5.3. Recomendaciones para profundizar	155
3.5.4. Clase 2: Indicadores de evaluación y criterios de validez	155
3.5.5. Clase 3: Análisis preliminar de evidencias	158
3.5.6. Clase 4: Reflexión sobre validez, sesgos y factibilidad	160
3.6. Transición hacia la reflexión final	163
3.6.1. Clase 1: Reflexión crítica sobre la experiencia	164
3.6.2. Recomendaciones para profundizar	166

3.1. Buenas prácticas de programación y refactorización

3.1.1. Contextualización de la experiencia pedagógica

En continuidad con los capítulos anteriores, que abordaron la transición conceptual hacia una comprensión integral de la Ingeniería de Software y la importancia del análisis y diseño como ejes de formación, este capítulo se enfoca en una etapa posterior del proceso formativo: la enseñanza del código limpio y las buenas prácticas de programación como fundamentos de la calidad profesional.

La experiencia se desarrolló en la Universidad Estatal de Milagro (UNEMI), en la asignatura Algoritmos y Lógica de Programación, correspondiente al primer nivel de la carrera de Ingeniería en Software, durante el período académico 2024–2025. El escenario pertenece a la Facultad de Ciencias de la Ingeniería y reúne a estudiantes que, tras su acercamiento inicial a los fundamentos de la disciplina, comienzan a enfrentarse a los desafíos de escribir código estructurado, legible y sostenible.

El grupo estuvo conformado mayoritariamente por jóvenes de entre 17 y 20 años, con predominio del género masculino (65%) y un 35% de participación femenina, lo que refleja avances graduales en la inclusión de mujeres en las carreras tecnológicas. La diversidad territorial fue también un rasgo característico: algunos estudiantes residen en Milagro, mientras que otros provienen de cantones cercanos como Naranjito, Yaguachi y El Triunfo, e incluso de zonas rurales que implican largos desplazamientos. Estas condiciones generan realidades diferenciadas de aprendizaje, particularmente en el acceso a equipos y conectividad, que condicionan el tiempo y la autonomía para el trabajo individual.

Durante las primeras semanas, se observaron dificultades recurrentes. Algunos estudiantes confundían símbolos en los diagramas de flujo (colocando operaciones dentro de rombos, por ejemplo), mientras otros presentaban errores en la estructura de los algoritmos, demostrando una comprensión memorística sin interiorización lógica. La falta de experiencia previa en programación, unida a las limitaciones de infraestructura, evidenció la necesidad de metodologías que vinculen la enseñanza técnica con el desarrollo de competencias cognitivas y metacognitivas. Esta reflexión dio origen al eje de la experiencia: **enseñar a escribir código de calidad desde las primeras etapas de la formación universitaria**.

3.1.2. Identificación del problema pedagógico: hacia las buenas prácticas de programación y refactorización

El problema central identificado fue la **ausencia de comprensión de los principios fundamentales de las buenas prácticas de programación**, manifestada en la incapacidad de los estudiantes para aplicar conceptos de claridad, modularidad y estructura lógica al construir algoritmos. Esta deficiencia se expresa en tres dimensiones críticas:

1. Legibilidad: los estudiantes utilizan valores literales (“números mágicos”) en lugar de variables con nombres semánticos.
2. Estructura modular: presentan dificultades para descomponer problemas complejos en subproblemas manejables.
3. Mantenibilidad: producen pseudocódigos sin documentación ni lógica secuencial coherente, imposibles de verificar o depurar.

Según Hermans (2021), la comprensibilidad del código depende de dimensiones cognitivas como la viscosidad (resistencia al cambio), la difusividad (dispersión de la información relacionada) y la visibilidad (facilidad para localizar componentes), características que suelen verse vulneradas cuando los estudiantes carecen de guía explícita sobre principios de calidad. El problema, por tanto, trasciende lo técnico: revela un déficit pedagógico en la formación inicial, que permite la producción de “código sucio” desde las primeras experiencias de aprendizaje.

Las consecuencias de esta carencia se reflejan a corto, mediano y largo plazo. A corto plazo, se genera una “**deuda técnica cognitiva**”, en la que cada error refuerza hábitos inadecuados que luego deben desaprenderse (Sweller, Ayres & Kalyuga, 2019).

A mediano plazo, los estudiantes avanzan sin dominar principios como **DRY (Don't Repeat Yourself)** o **KISS (Keep It Simple, Stupid)**, dificultando el trabajo en equipo y la reutilización del código.

A largo plazo, se gradúan profesionales con escasa capacidad de integrarse en proyectos colaborativos, donde la legibilidad, modularidad y documentación son esenciales.

Las evidencias empíricas recogidas en aula confirman estas dificultades: un estudiante escribió `resultado = 5 * 3` en lugar de `area_rectangulo = base * altura`, omitiendo la semántica del nombre de la variable; otros ubicaron operaciones dentro de rombos en diagramas de flujo, confundiendo los símbolos de decisión y proceso. Además,

varios intentaron resolver todo el problema en una sola secuencia, sin aplicar descomposición modular, generando sobrecarga cognitiva.

Sentance et al. (2019b) destacan que la metodología **PRIMM** (*Predict – Run – Investigate – Modify – Make*) resulta efectiva precisamente porque fomenta la descomposición progresiva y la refactorización incremental, principios ausentes en los enfoques tradicionales que exigen soluciones completas desde el primer intento.

En este sentido, el problema detectado exige un replanteamiento pedagógico profundo, que incorpore la enseñanza explícita de estándares de calidad del código como parte del proceso formativo.

3.1.3. El rol del docente como revisor de código pedagógico

Superar estas deficiencias requiere un cambio de enfoque en la enseñanza de la programación. El docente debe asumir el rol de “**revisor de código pedagógico**”, capaz de identificar violaciones a los estándares de calidad y guiar procesos de mejora iterativa. Sin embargo, la gran cantidad de estudiantes y el tiempo limitado dificultan ofrecer retroalimentación personalizada. Frente a ello, Kafai et al. (2019) proponen el uso de **revisiones de código entre pares (peer code review)**, que distribuyen la carga de evaluación y favorecen la comprensión mutua: quien revisa consolida su propio conocimiento, mientras quien recibe la revisión obtiene múltiples perspectivas sobre la legibilidad y estructura de su código.

El rol institucional también es clave: se requiere infraestructura que promueva las buenas prácticas, como entornos de desarrollo con análisis estático, control de versiones y evaluaciones automáticas que valoren tanto la corrección funcional como la calidad estructural del código. De igual manera, los diseñadores curriculares deben integrar estos principios desde los primeros niveles de la carrera, fomentando hábitos de calidad profesional desde el inicio.

Hermans (2021) sugiere estrategias concretas para enseñar buenas prácticas:

- **Presentar ejemplos contrastantes:** comparar código correcto con código funcional pero deficiente.
- **Implementar sesiones de refactorización guiada:** donde el docente verbaliza el proceso de mejora iterativa.

- **Aplicar evaluaciones formativas** que consideren legibilidad, modularidad y documentación, no solo funcionalidad.

Persistir en modelos de enseñanza que valoran únicamente si el algoritmo “funciona” implica formar técnicos que resuelven problemas inmediatos, pero no desarrollan soluciones sostenibles. Por ello, la enseñanza de la programación debe integrar explícitamente los principios de código limpio, formando profesionales capaces de escribir soluciones escalables y mantenibles, alineadas con los estándares de la industria tecnológica contemporánea.

3.1.4. Cierre integrador

La experiencia aquí sistematizada demuestra que la enseñanza de la programación no debe limitarse a la lógica o la sintaxis, sino que debe incorporar desde los primeros niveles una **cultura de calidad del código**. La comprensión del código limpio como práctica cognitiva, ética y profesional constituye un paso fundamental para la formación de ingenieros de software reflexivos y competentes.

De este modo, el proceso de enseñanza de la Ingeniería de Software se concibe como una cadena coherente que une **concepto, análisis y calidad**, promoviendo la formación de profesionales capaces de pensar, diseñar y construir software con sentido crítico y responsabilidad profesional.

3.2. Del código al pensamiento: enseñanza del código limpio en la formación inicial de ingenieros en software

La primera parte de este capítulo permitió situar la problemática pedagógica de la enseñanza del código limpio en la formación inicial de los ingenieros en software, destacando la necesidad de incorporar principios de legibilidad, modularidad y mantenibilidad desde las primeras experiencias de programación. En continuidad con ese análisis, la presente entrega profundiza en la **fundamentación conceptual y operativa** que sustenta la experiencia, definiendo los **conceptos estructurantes, dimensiones, indicadores, fuentes y métodos** que permiten comprender cómo los estudiantes transitan desde la ejecución mecánica del código hacia una práctica reflexiva, ordenada y profesional. Este tramo teórico-metodológico se propone, así, consolidar el andamiaje que articula la compren-

sión cognitiva del pensamiento computacional con la aplicación técnica de las buenas prácticas de programación en el contexto universitario.

3.2.1. Bisagra Textual

Introducción: El módulo anterior expuso el contexto institucional y pedagógico de la experiencia, identificando el problema formativo vinculado a la falta de comprensión de las buenas prácticas de programación y su impacto en la calidad del aprendizaje. Desde esa base, se delimitó el propósito de promover el código limpio como eje de formación profesional y se establecieron criterios de valor centrados en la claridad, la modularidad y la mantenibilidad. A partir de aquí, el capítulo transita del relato descriptivo hacia una **fundamentación conceptual y operativa**, en la que se desarrollan los **principios teóricos, dimensiones e indicadores** que orientan la enseñanza de la programación bajo estándares de calidad, sustentados en fuentes y métodos propios de la ingeniería de software y la didáctica universitaria.

3.2.2. Identificación de conceptos estructurantes

Los conceptos clave que orientan esta etapa de la experiencia son: identificación de variables, uso semántico de nombres significativos, comprensión de símbolos en diagramas de flujo, estructura secuencial, correspondencia lógica entre símbolos y operaciones, y documentación básica. Estos elementos constituyen los pilares del pensamiento computacional, pues permiten que el estudiante construya algoritmos comprensibles, organizados y sostenibles desde sus primeras experiencias de aprendizaje.

La elección de estos conceptos responde a las **falencias observadas durante las prácticas de laboratorio**, donde los estudiantes mostraron vacíos al reconocer variables, confusión al emplear símbolos en los diagramas de flujo y dificultades para expresar la secuencia lógica de los procesos. Estas carencias no son solo técnicas, sino **cognitivas y metodológicas**, pues evidencian que el aprendizaje de la programación aún se aborda desde la memorización de formas en lugar de la comprensión del razonamiento lógico. Por ello, estos conceptos fueron seleccionados como núcleos de intervención pedagógica, orientados a mejorar la claridad, la modularidad y la mantenibilidad del código.

La **identificación de variables** constituye un punto de partida esencial. Según Guzodial y Morrison (2020), enseñar a reconocer y nombrar variables con sentido semántico desarrolla la capacidad de **abstraer y organizar información**, favoreciendo la transición

del pensamiento matemático al pensamiento computacional. De manera complementaria, la **comprensión de los símbolos de los diagramas de flujo** actúa como un puente entre lo visual y lo lógico. Como señalan Sentance et al. (2019a), estos recursos gráficos ayudan al estudiante a “ver” la estructura del algoritmo, siempre que su enseñanza se base en la interpretación y no en la repetición mecánica de símbolos.

A estos conceptos se suman la **estructura secuencial** y la **documentación del pseudocódigo**, que contribuyen a la coherencia y trazabilidad del pensamiento algorítmico. Sweller, Van Merriënboer y Paas (2019) sostienen que una secuencia clara y bien organizada reduce la **carga cognitiva**, facilitando la comprensión de las relaciones entre entradas, procesos y salidas. La documentación, por su parte, introduce desde el inicio la práctica de escribir código legible y explicativo, fortaleciendo la reflexión sobre el propio proceso de programación.

En síntesis, estos conceptos **organizan y orientan la experiencia pedagógica** al proporcionar un marco estructurado para el desarrollo de la lógica computacional. Su enseñanza requiere ejemplos concretos, situaciones cercanas y ejercicios que conecten la programación con la realidad cotidiana del estudiante. De esta forma, la comprensión de las variables y los diagramas de flujo se convierte en la base para introducir las **dimensiones de análisis** del siguiente apartado, centradas en la refactorización, la evaluación y la aplicación de buenas prácticas de programación como parte del proceso formativo en Ingeniería en Software.

3.2.3. Formulación de dimensiones

El Puente 2 aborda la **formulación de dimensiones analíticas** como una estrategia para organizar la interpretación de la experiencia pedagógica en la enseñanza de la programación. En la sistematización, las dimensiones funcionan como **ejes articuladores entre la teoría y la práctica**, permitiendo transformar observaciones concretas en categorías de análisis que revelan el modo en que los estudiantes construyen conocimiento y aplican habilidades técnicas.

A partir de los conceptos estructurantes —identificación de variables, comprensión de símbolos de diagramas de flujo, secuencia lógica y aplicación de buenas prácticas— se definieron dos dimensiones centrales: la Cognitivo–Conceptual y la Técnico–Procedimental, ambas inspiradas en enfoques recientes sobre pensamiento computacional y enseñanza del código limpio.

La **Dimensión Cognitivo-Conceptual** se orienta al desarrollo de la lógica y la abstracción algorítmica. Según Guzdial y Morrison (2020), comprender la programación implica mucho más que dominar la sintaxis: requiere desarrollar estructuras mentales que permitan pensar computacionalmente. De modo complementario, Sweller, Van Merriënboer y Paas (2019) sostienen que un diseño instruccional basado en la reducción de la carga cognitiva facilita que los estudiantes integren de manera significativa conceptos como variables y secuencias. En la práctica, esta dimensión se manifestó cuando los estudiantes confundían valores numéricos con variables, revelando la necesidad de reforzar la comprensión simbólica y el razonamiento lógico mediante ejemplos cotidianos y ejercicios guiados.

Por su parte, la **Dimensión Técnico-Procedimental** examina cómo los estudiantes aplican principios de **claridad, legibilidad y mantenibilidad del código**, consolidando las buenas prácticas de programación. Hermans (2021) plantea que la calidad del software depende tanto de su funcionalidad como de su comprensibilidad, mientras que Sentance et al. (2019b) destacan el valor pedagógico de la refactorización progresiva para fortalecer la calidad estructural del código. En la experiencia, esta dimensión se evidenció cuando, tras ejercicios de refactorización guiada, los estudiantes lograron mejorar la organización de sus pseudocódigos, incorporando comentarios explicativos, nombres descriptivos y separación clara de etapas.

En conjunto, ambas dimensiones ofrecen un marco que **integra la comprensión conceptual con la aplicación técnica**, articulando el aprendizaje cognitivo con la práctica profesional. Este esquema permite avanzar hacia la **formulación de indicadores observables** que evalúen tanto la evolución del pensamiento lógico como la apropiación de las buenas prácticas de programación, alineando la enseñanza universitaria con los estándares contemporáneos de la ingeniería de software.

3.2.4. Construcción de indicadores

La construcción de indicadores constituye una etapa clave en el proceso de sistematización, pues permite **operativizar las dimensiones analíticas** previamente definidas y traducirlas en elementos observables y evaluables. En el ámbito educativo, los indicadores funcionan como **herramientas de medición cualitativa y cuantitativa** que posibilitan valorar el grado de desarrollo de competencias, conocimientos y habilidades alcanzadas por los estudiantes (Lodi & Martini, 2021). En el contexto de la enseñanza de la progra-

mación, los indicadores no solo evidencian el dominio técnico, sino también la **evolución del pensamiento lógico y la apropiación de buenas prácticas de codificación**, fundamentales para la formación de ingenieros de software competentes. Su diseño responde a la necesidad de transformar conceptos teóricos —como la abstracción, la modularidad o la claridad del código— en comportamientos concretos que pueden observarse en el aula o en el laboratorio.

Desde esta perspectiva, se han formulado **seis indicadores distribuidos en dos dimensiones**: la **Dimensión Cognitivo–Conceptual**, centrada en la comprensión de la lógica y la abstracción algorítmica, y la **Dimensión Técnico–Procedimental**, orientada a la aplicación de buenas prácticas y la calidad del código. Cada indicador expresa un nivel de logro esperable, sustentado en la literatura sobre pensamiento computacional y enseñanza de la programación.

3.2.5. Dimensión Cognitivo–Conceptual: Comprensión de la lógica y la abstracción algorítmica

Esta dimensión aborda la capacidad del estudiante para **comprender los principios lógicos y representativos** de la programación, movilizando procesos mentales de abstracción, análisis y secuenciación. Lodi y Martini (2021) sostienen que la enseñanza del pensamiento computacional implica ayudar al estudiante a construir modelos mentales que le permitan conectar los conceptos algorítmicos con su significado funcional. Asimismo, Grover y Pea (2013) afirman que el desarrollo de competencias en programación requiere integrar la comprensión conceptual con la capacidad de traducir ideas en pasos secuenciales.

Indicadores propuestos:

1. Identificación y uso semántico de variables.
2. Comprensión funcional de los símbolos en los diagramas de flujo.
3. Coherencia en la secuencia lógica del pseudocódigo.

Cada uno de estos indicadores traduce el grado de internalización del razonamiento algorítmico. Por ejemplo, el **indicador 1.1** se manifiesta cuando el estudiante reemplaza números literales por variables significativas, como en el caso de escribir `area_rectangulo`

= base * altura, lo que demuestra comprensión semántica del problema. En el **indicador 1.2**, la evidencia surge del uso adecuado de los símbolos de proceso y decisión en los diagramas de flujo; es decir, cuando los estudiantes emplean el rombo para decisiones y el rectángulo para operaciones, demostrando dominio de la representación simbólica. Finalmente, el **indicador 1.3** se verifica en pseudocódigos que siguen una secuencia lógica Entrada–Proceso–Salida, evitando saltos o desórdenes en la estructura.

Curzon y McOwan (2019) destacan que el aprendizaje significativo de la programación se consolida cuando el estudiante es capaz de **visualizar la lógica detrás de los algoritmos** y relacionarla con situaciones concretas, lo que permite transitar de la memorización mecánica a la comprensión estructural. En la experiencia pedagógica desarrollada en la Universidad Estatal de Milagro, este enfoque se manifestó cuando los estudiantes comenzaron a identificar patrones de razonamiento comunes entre ejercicios distintos, mostrando progresos en su comprensión cognitiva del proceso algorítmico.

3.2.6. Dimensión Técnico–Procedimental: Aplicación de buenas prácticas y calidad del código

Esta dimensión se orienta a la observación del **comportamiento práctico y profesional del estudiante** al escribir, organizar y documentar su código. De acuerdo con Lodi y Martini (2021), el aprendizaje técnico se consolida cuando el estudiante logra aplicar principios de calidad que garantizan la mantenibilidad y legibilidad del software. Grover y Pea (2020) complementan esta idea señalando que las buenas prácticas no deben enseñarse como normas aisladas, sino como parte de un proceso formativo que vincula la claridad técnica con el pensamiento crítico.

Indicadores propuestos:

1. Legibilidad y claridad del código.
2. Modularidad y refactorización progresiva.
3. Documentación y mantenibilidad del código.

Estos indicadores permiten evaluar la **madurez profesional del aprendizaje**. En el **indicador 2.1**, se busca que el estudiante estructure su código con sangrías, espacios y comentarios, garantizando legibilidad. La evidencia se observa en comparativas entre versiones iniciales (sin formato ni explicación) y versiones refactorizadas (con orden y

comentarios). El **indicador 2.2** mide la capacidad de dividir un problema en subprocesos o módulos, mostrando comprensión de la refactorización progresiva. En las prácticas, esto se evidenció cuando los estudiantes implementaron funciones como calcularPromedio() o mostrarResultado(), reorganizando su código de manera más eficiente. Finalmente, el **indicador 2.3** evalúa la incorporación de anotaciones y encabezados descriptivos, reflejando la comprensión de la importancia de documentar para otros y para el futuro mantenimiento del software.

Curzon y McOwan (2019) afirman que la programación se enseña mejor cuando se combina la práctica técnica con la reflexión sobre el proceso, de modo que los estudiantes comprendan por qué una solución limpia y modular no solo funciona, sino que **puede ser entendida y reutilizada**. En el contexto de esta experiencia, tras aplicar sesiones de refactorización guiada, los estudiantes lograron mejorar la legibilidad y mantenibilidad de sus algoritmos, lo que sugiere que la práctica metódica consolida los principios de calidad de código desde etapas tempranas de formación.

3.2.7. Fuentes y métodos de verificación

En una sistematización de experiencias, las **fuentes y métodos de verificación** constituyen los pilares que garantizan la **validez, coherencia y rigor del análisis**. Las fuentes representan el origen de la información —textual, empírica o testimonial—, mientras que los métodos de verificación permiten comprobar la consistencia entre lo que se observa, lo que se interpreta y lo que se concluye. Según Jara (2018a), las evidencias en una sistematización no son simples datos, sino “rastros significativos de la práctica” que, interpretados con criterio teórico, permiten construir conocimiento educativo situado. Por ello, la elección de fuentes y métodos debe responder a una doble finalidad: **comprobar la validez de los hallazgos y asegurar la coherencia entre teoría y experiencia** en el proceso de enseñanza-aprendizaje de la programación.

Flick (2014) enfatiza que la **pertinencia de las fuentes** radica en su capacidad para responder a las preguntas del estudio; es decir, deben ser seleccionadas no por su cantidad, sino por su relevancia para comprender el fenómeno. En este caso, las fuentes principales se estructuran en torno a los dos ejes de análisis del proyecto: la comprensión cognitivo-conceptual y la aplicación técnico-procedimental.

Se eligieron tres fuentes teóricas fundamentales —Lodi y Martini (2021), Grover y Pea (2020), y Curzon y McOwan (2019)— complementadas por evidencias empíricas

obtenidas a través de la observación, los pseudocódigos producidos por los estudiantes y sus reflexiones en entrevistas breves. Estas fuentes permiten triangular teoría y práctica, asegurando que los resultados no dependan de una sola perspectiva, sino que se fortalezcan mediante contrastes y convergencias, siguiendo el enfoque de **triangulación múltiple propuesto por Yin (2014)**.

3.2.8. Modelar y abstraer procesos lógicos

Lodi y Martini (2021) plantean que el pensamiento computacional implica la capacidad de **modelar procesos lógicos mediante representaciones simbólicas**, lo cual fundamenta el **Indicador 1.1: Identificación y uso semántico de variables**. Para verificar este principio, se aplicó el **análisis de pseudocódigos y diagramas de flujo iniciales**, con el fin de observar si los estudiantes lograban abstraer y representar las relaciones entre los elementos de un problema antes de codificar.

Los resultados evidenciaron progresos significativos: estudiantes que al inicio operaban con valores literales, posteriormente comenzaron a emplear variables con nombres semánticamente coherentes. Siguiendo la propuesta de R. Stake (1995), este método mantiene coherencia entre la fuente conceptual y el procedimiento empírico, pues traduce una idea teórica —la abstracción lógica— en un indicador verificable a partir del producto educativo real.

3.2.9. Comprensión de la secuencia algorítmica

Para Grover y Pea (2020), el pensamiento computacional integra comprensión conceptual y acción secuencial; aprender a programar implica pensar en términos de pasos ordenados y relaciones causales. Este principio fundamenta el Indicador 1.2: Comprensión funcional de los símbolos en los diagramas de flujo, el cual se verificó mediante entrevistas reflexivas y cuestionarios. En estas actividades, los estudiantes explicaron por qué seleccionaban ciertos símbolos o cómo interpretaban la secuencia de decisiones dentro del algoritmo. Este método de verificación permitió evaluar no solo el resultado técnico, sino la lógica interna de su razonamiento. En coherencia con Yin (2014), combinar instrumentos discursivos con productos visuales amplía la validez de la evidencia, al incorporar tanto la voz del estudiante como su representación gráfica del pensamiento lógico.

3.2.10. Aprendizaje significativo y contextualizado

Curzon y McOwan (2019) destacan la importancia de situar la enseñanza de la programación en contextos significativos, donde los ejemplos prácticos actúan como puentes entre la teoría y la experiencia. Esta fuente sustenta el Indicador 1.3: Coherencia en la secuencia lógica del pseudocódigo, verificado mediante la observación directa en sesiones de laboratorio. Durante estas actividades, se registró si los estudiantes aplicaban ejemplos cotidianos —como el cálculo de promedios o materiales de construcción— para estructurar algoritmos coherentes.

Los resultados mostraron que contextualizar las tareas permitió mejorar la secuencia lógica, reforzando la comprensión de la estructura Entrada–Proceso–Salida. Como plantea Flick (2014), este tipo de observación participante fortalece la calidad interpretativa de la sistematización, al permitir comprender cómo el conocimiento se construye en interacción.

3.2.11. Fuentes complementarias y métodos asociados

Además de las tres fuentes principales, se integraron autores que sustentan el rigor metodológico de la verificación. Jara (2018a) enfatiza que toda evidencia debe interpretarse desde su sentido pedagógico, no solo desde su función comprobatoria; por ello, los pseudocódigos, entrevistas y observaciones fueron tratados como huellas de aprendizaje y no como datos aislados. Yin (2014) propone la triangulación como estrategia de validación, articulando tres niveles: el teórico (fuentes bibliográficas), el empírico (productos y observaciones) y el reflexivo (auto-evaluaciones estudiantiles). Finalmente, R. Stake (1995) recuerda que la coherencia entre fuente y método es esencial: la manera de recolectar y analizar los datos debe corresponder al tipo de evidencia que se busca generar. Así, en esta experiencia, los métodos de análisis fueron seleccionados no por conveniencia, sino por su correspondencia con los indicadores definidos.

3.2.12. Síntesis integradora

El uso combinado de fuentes y métodos de verificación otorga **robustez y credibilidad a la sistematización**, garantizando que los hallazgos emergen de un proceso reflexivo y contrastado. Las fuentes teóricas (Lodi & Martini, 2021); (Grover & Pea, 2020); (Curzon & McOwan, 2019) ofrecieron los marcos conceptuales que definieron los indicadores,

mientras que los métodos empíricos —análisis de productos, entrevistas y observación— proporcionaron la evidencia concreta de aprendizaje. Esta triangulación, en el sentido de Yin (2014), permite afirmar que la comprensión algorítmica y la aplicación técnica observadas no son hechos aislados, sino resultados consistentes de una práctica pedagógica fundamentada. En definitiva, la combinación de teoría, evidencia y reflexión genera un proceso de sistematización **válido, coherente y transferible**, en línea con la concepción de Jara (2018a), quien afirma que toda experiencia educativa innovadora alcanza su valor cuando logra convertir la práctica en conocimiento compartido.

3.2.13. Justificación teórica del conjunto

El conjunto de conceptos y dimensiones definidos en esta sistematización se fundamenta en la necesidad de comprender la enseñanza de la programación no solo como un proceso técnico, sino como una práctica cognitiva, comunicativa y formativa. Siguiendo a Jara (2018a), las categorías de análisis deben construirse desde la experiencia y reflejar las tensiones y aprendizajes reales que emergen en la práctica pedagógica. En este sentido, los conceptos de identificación de variables, comprensión de símbolos en diagramas de flujo y secuencia lógica del pseudocódigo surgieron como nudos problemáticos detectados en el aula, que permiten analizar la relación entre la comprensión conceptual y la acción técnica.

Asimismo, la organización de los conceptos en dos dimensiones — **Cognitivo – Conceptual y Técnico – Procedimental** — responde a la recomendación metodológica de Flick (2014), quien destaca que las categorías deben articular el “qué” y el “cómo” del fenómeno estudiado. La primera dimensión busca indagar en la capacidad de los estudiantes para abstraer, simbolizar y razonar de forma lógica; la segunda analiza la manera en que esa comprensión se traduce en prácticas de codificación limpias y sostenibles. Así, las dimensiones se constituyen en ejes interpretativos que integran teoría, experiencia y observación empírica, generando un marco coherente para el análisis de la calidad del aprendizaje.

En el plano teórico, los aportes de Lodi y Martini (2021), Grover y Pea (2020) y Curzon y McOwan (2019) sustentan la dimensión cognitiva, al situar el pensamiento computacional como una competencia que combina modelamiento, abstracción y razonamiento lógico. Por su parte, M. Zanatta y da Silva (2022), Ahmad y Hashim (2020), y Petre y van der Hoek (2021) fundamentan la dimensión técnico–procedimental, al en-

fatizar que la enseñanza de buenas prácticas de programación desde los primeros niveles académicos favorece la comprensión de estándares profesionales y el desarrollo de una ética de la calidad del código.

3.2.14. Justificación de los indicadores

Los indicadores definidos constituyen la traducción operativa de las dimensiones y permiten observar de forma empírica los avances en el aprendizaje. En coherencia con **Yin (2014)**, cada indicador actúa como una variable analítica que vincula el marco conceptual con la evidencia concreta. Por ejemplo, la identificación y uso semántico de variables se verifica mediante pseudocódigos que muestran la comprensión del valor simbólico de los datos, mientras que la coherencia en la secuencia lógica del pseudocódigo refleja el tránsito desde la memorización hacia la comprensión algorítmica.

De acuerdo con R. Stake (1995), los indicadores deben mantener una coherencia directa con los objetivos de la experiencia y con las dimensiones teóricas que los sustentan. En este caso, los seis indicadores —tres por dimensión— fueron diseñados para captar los niveles de apropiación del pensamiento computacional y de las buenas prácticas de programación. Su formulación busca equilibrio entre la **validez conceptual** (relación con los autores de referencia) y la **validez empírica** (posibilidad de observación en la práctica). Este enfoque posibilita un análisis riguroso de los aprendizajes sin descontextualizar la experiencia educativa en la que emergen.

3.2.15. Justificación de las fuentes y métodos de verificación

Las fuentes teóricas y metodológicas seleccionadas permiten una triangulación sólida entre los marcos de referencia académicos y las evidencias recogidas en aula. Siguiendo a Flick (2014), la pertinencia de una fuente radica en su capacidad de responder al objeto de estudio, más que en su cantidad. Por ello, se priorizaron autores que abordan directamente el pensamiento computacional y la calidad del código, garantizando una relación directa entre teoría y práctica.

En cuanto a los métodos, esta investigación aplica estrategias de verificación que integran análisis de pseudocódigos y diagramas de flujo, entrevistas reflexivas, observación de sesiones prácticas y revisión entre pares. Según Yin (2014), esta diversidad metodológica fortalece la validez interna mediante la triangulación de datos; mientras que para R.

Stake (1995), la coherencia entre fuente y método garantiza que las evidencias respondan al tipo de conocimiento que se busca construir.

La incorporación de Jara (2018b) resulta clave para comprender las evidencias no como simples datos, sino como “huellas significativas” de la práctica docente. De esta manera, cada pseudocódigo corregido, cada reflexión de los estudiantes o cada versión refactorizada del código se interpreta como una manifestación de aprendizaje situada. Esta lectura cualitativa complementa el análisis técnico y permite sostener una mirada integral del proceso formativo.

Por otro lado, los aportes de Carlino (2005) y Hyland (2009) enriquecen la interpretación al reconocer la escritura académica como una práctica social, colaborativa y situada. En este sentido, los comentarios, las rúbricas y las revisiones entre pares no solo constituyen mecanismos de evaluación, sino también formas de comunicación disciplinar que configuran una comunidad de aprendizaje. Así, el código, sus comentarios y las discusiones que genera se convierten en textos académicos en sí mismos, donde los estudiantes aprenden a argumentar, justificar y documentar sus decisiones técnicas.

3.2.16. Síntesis final del conjunto

En conjunto, la estructura teórica y metodológica desarrollada articula los tres niveles esenciales de la sistematización: **conceptualización, operativización y verificación**. Los conceptos y dimensiones ofrecen el andamiaje analítico; los indicadores proporcionan las unidades observables; y las fuentes y métodos de verificación aseguran la validez y coherencia del proceso. Esta arquitectura responde a la orientación de Jara (2018a), quien concibe la sistematización como un proceso reflexivo que transforma la experiencia en conocimiento.

El equilibrio entre teoría y práctica, entre la abstracción lógica y la acción técnica, permite evidenciar cómo los estudiantes transitán de un pensamiento lineal y memorístico hacia una comprensión estructurada y profesional del código. Al integrar autores como **Lodi, Grover, Curzon, Zanatta, Ahmad y Petre** con referentes metodológicos como **Flick, Yin, Stake, Carlino y Hyland**, este conjunto alcanza una justificación teórica robusta, capaz de sostener la validez conceptual y pedagógica del proceso de enseñanza de la programación como práctica formativa, ética y comunicativa.

3.2.17. Recomendaciones para profundizar

1. Equilibrar teoría y práctica en la justificación:

Asegurar que cada argumento teórico se vincula con un ejemplo empírico concreto, evitando la separación entre reflexión conceptual y evidencia pedagógica.

1. Verificar que todas las partes del conjunto estén fundamentadas:

Revisar que cada dimensión, indicador y método cuente con al menos una referencia explícita, fortaleciendo la coherencia argumentativa y la validez del texto.

1. Evitar redundancias y mantener cohesión discursiva:

Integrar las citas de forma fluida, priorizando la interpretación sobre la acumulación de autores, y mantener un hilo narrativo que une los conceptos, dimensiones e indicadores como partes de un mismo sistema analítico.

3.3. Aprender gestionando: experiencias innovadoras en proyectos de software universitarios

3.3.1. Transición al vínculo curricular

Al culminar el proceso de fundamentación teórica y operativa de la experiencia desarrollada en la asignatura Introducción a la Ingeniería de Software, se evidencia que los aprendizajes alcanzados no se limitan al plano metodológico, sino que se integran de manera coherente con el proyecto formativo de la carrera. Las dimensiones construidas -pedagógica, cognitiva-formativa y socioafectiva-colaborativa- permiten comprender cómo la práctica docente contribuye al desarrollo de competencias profesionales clave, fortaleciendo la relación entre teoría y práctica en los primeros niveles de formación universitaria.

Desde una perspectiva curricular, la experiencia se articula con el perfil de egreso del Ingeniero de Software de la UNEMI, quien se caracteriza por su capacidad para desarrollar soluciones tecnológicas de calidad, trabajar colaborativamente, liderar equipos multidisciplinarios y actuar con responsabilidad ética y compromiso social.

En este sentido, los aprendizajes logrados en el aula -aprendizaje activo, pensamiento crítico, autonomía y trabajo colaborativo- responden directamente a los resultados de aprendizaje del programa, al promover la aplicación de estándares profesionales, la comunicación efectiva y la innovación mediante el uso de tecnologías emergentes. La incorporación de estrategias como la mentoría pedagógica y el aprendizaje basado en proyectos favorece la formación de estudiantes autónomos y reflexivos, capaces de enfrentar problemas reales del campo de la ingeniería con una visión integral y ética.

Esta experiencia, además, se convierte en un punto de inflexión en el proceso formativo, al evidenciar que la enseñanza de la ingeniería de software debe trascender la instrucción técnica para situarse en el desarrollo de competencias blandas, cognitivas y socioemocionales que sustentan la empleabilidad y la actualización profesional continua. Así, la práctica sistematizada no solo refuerza el perfil de egreso institucional, sino que también aporta una mirada pedagógica innovadora para la formación inicial en ingeniería, constituyéndose en una referencia para el rediseño de estrategias curriculares orientadas al aprendizaje significativo y al fortalecimiento del vínculo entre la docencia y el desarrollo profesional.

3.3.2. Identificación de competencias del perfil

La experiencia desarrollada en la asignatura Introducción a la Ingeniería de Software permitió evidenciar cómo la práctica docente puede articularse con las competencias del perfil de egreso de la carrera, convirtiéndose en un espacio de formación integral para los estudiantes de primer semestre.

En este nivel inicial, el desafío no radica únicamente en la adquisición de conocimientos técnicos, sino en la construcción de una base competencial que combine la comprensión conceptual, la colaboración efectiva y la autonomía en el aprendizaje. Por ello, identificar las competencias que se fortalecen a través de esta experiencia resulta clave para evidenciar su alineación con el currículo y el proyecto formativo institucional.

De acuerdo con el perfil de egreso de la carrera de Ingeniería de Software de la UNE-MI, el profesional debe ser capaz de desarrollar, mantener e innovar sistemas de software de calidad, aplicando estándares internacionales y buenas prácticas; trabajar en equipos multidisciplinarios; liderar proyectos; y actuar con ética y responsabilidad social.

En esta línea, las competencias que se consolidaron en la práctica docente fueron principalmente: trabajo colaborativo, pensamiento crítico, autonomía en el aprendizaje,

responsabilidad ética y comunicación efectiva. Estas competencias, tanto genéricas como específicas, se integran de forma transversal al proceso formativo, constituyendo el núcleo de la formación del ingeniero de software contemporáneo.

La competencia de trabajo colaborativo se manifestó en la experiencia a través del desarrollo de proyectos en equipo, donde los estudiantes asumieron roles complementarios -analista, programador y validador-, simulando entornos reales de producción de software.

Esta metodología permitió fortalecer habilidades de coordinación, negociación y co-responsabilidad, aspectos que según Johnson, Johnson et al. (2020) resultan esenciales para el aprendizaje cooperativo en contextos de ingeniería. Asimismo, el trabajo en equipo fomenta la empatía profesional y la capacidad de resolver conflictos de manera constructiva, competencias destacadas en el perfil de egreso institucional. En un campo altamente interdependiente como el desarrollo de software, estas habilidades sociales y de colaboración constituyen un pilar fundamental para la empleabilidad y el éxito profesional.

Por su parte, la competencia de pensamiento crítico se relacionó con la capacidad de los estudiantes para analizar y evaluar los problemas planteados durante la construcción de proyectos, argumentando sus decisiones técnicas con base en evidencia y criterios de calidad. Tal como sostiene Facione (2020), el pensamiento crítico permite a los futuros profesionales emitir juicios fundamentados y asumir decisiones responsables en contextos complejos.

En la asignatura, esta competencia se promovió mediante la revisión de código, la detección de errores lógicos y la reflexión sobre las estrategias de resolución implementadas. Estas prácticas fomentaron en los estudiantes la capacidad de razonar con rigor, una competencia que, según Lai (2022), constituye un indicador de madurez cognitiva y profesional en entornos STEM.

La autonomía en el aprendizaje fue otra competencia central, consolidada a través de la planificación semanal de avances, la gestión de tareas y la autoevaluación continua de los logros alcanzados. Zimmerman (2020) define la autorregulación del aprendizaje como la capacidad de planificar, monitorear y evaluar el propio proceso formativo.

En el contexto de la ingeniería de software, esta habilidad permite que los estudiantes desarrollen la disciplina y la resiliencia necesarias para enfrentar proyectos de largo plazo y adaptarse a las demandas cambiantes del mercado tecnológico. La autonomía, además, se conecta directamente con la competencia de aprendizaje permanente, presente en el

perfil de egreso, que enfatiza la importancia de mantenerse actualizado en tecnologías emergentes y buenas prácticas de desarrollo.

La responsabilidad ética y social se evidenció en la forma en que los estudiantes asumieron el impacto de las decisiones tecnológicas en la sociedad, discutiendo dilemas éticos relacionados con la privacidad de datos, la equidad digital y la sostenibilidad del software.

Según Morales et al. (2021), integrar la dimensión ética en la formación tecnológica favorece la construcción de profesionales conscientes del alcance social de sus acciones. Esta competencia, presente en el perfil de egreso, trasciende el plano técnico para consolidar una formación humanista y responsable, coherente con el compromiso de la UNEMI hacia el desarrollo sostenible y el bienestar colectivo.

Finalmente, la comunicación efectiva emergió como una competencia transversal, indispensable tanto para la gestión de proyectos como para la documentación técnica y la presentación de resultados. Hargie (2021) subraya que la comunicación profesional en ingeniería requiere precisión, claridad y empatía, habilidades que se fortalecen mediante la exposición oral, la escritura técnica y la retroalimentación entre pares. En las actividades realizadas, los estudiantes presentaron sus avances, defendieron decisiones de diseño y compartieron aprendizajes, fortaleciendo así la capacidad de expresarse con propiedad en entornos académicos y laborales.

Estas cinco competencias - colaboración, pensamiento crítico, autonomía, ética y comunicación - conforman un entramado coherente que responde a las demandas del perfil de egreso y a los retos contemporáneos de la educación en ingeniería.

En la experiencia sistematizada, dichas competencias se articularon de forma integral, mostrando que el proceso formativo no se limita a la enseñanza de contenidos técnicos, sino que abarca el desarrollo de capacidades transversales que preparan a los estudiantes para desempeñarse con eficacia, liderazgo y compromiso social en su vida profesional.

De este modo, la identificación de las competencias del perfil no solo evidencia la pertinencia curricular de la experiencia, sino que proyecta el proceso hacia los resultados de aprendizaje que serán analizados en el siguiente apartado. Estas competencias se constituyen, por tanto, en los pilares sobre los cuales se construyen los aprendizajes observables que reflejan la transformación del estudiante a lo largo del proceso educativo.

3.3.3. Resultados de aprendizaje vinculados

La sistematización de la experiencia desarrollada en la asignatura Introducción a la Ingeniería de Software permitió constatar la correspondencia entre las competencias del perfil de egreso y los resultados de aprendizaje propuestos en el plan curricular de la carrera. En la práctica docente, los resultados de aprendizaje funcionan como un punto de conexión entre las metas formativas del programa y los logros observables alcanzados por los estudiantes.

En el contexto de la Universidad Estatal de Milagro (UNEMI), estos resultados se formulan en coherencia con el Modelo Educativo Institucional (MEI) y el Marco Curricular Institucional (MCI), que establecen como principio fundamental el desarrollo integral de los estudiantes mediante el aprendizaje activo, reflexivo y colaborativo (Universidad Estatal de Milagro (UNEMI), [2021](#)).

En este sentido, los resultados de aprendizaje fortalecidos a partir de la experiencia se agrupan en tres dimensiones que responden al enfoque por competencias definido por la UNEMI: desarrollo del pensamiento crítico y la resolución de problemas, colaboración y comunicación efectiva en entornos de ingeniería, y actuación ética y compromiso con el aprendizaje continuo. Estas dimensiones, además de alinearse con el perfil de egreso de la carrera, evidencian la coherencia entre la práctica pedagógica y el modelo curricular institucional orientado a resultados.

El primer resultado de aprendizaje evidenciado fue la capacidad para analizar problemas de software y proponer soluciones fundamentadas en principios de ingeniería y criterios de calidad. Durante la experiencia, los estudiantes aplicaron estrategias de análisis, diseño y validación de sistemas, desarrollando la competencia de pensamiento crítico mediante la revisión de código, la evaluación de alternativas y la justificación argumentada de sus decisiones. Este resultado se vincula directamente con el componente de razonamiento lógico y pensamiento crítico establecido en el MCI, que enfatiza la formación de profesionales capaces de identificar, analizar y resolver problemas complejos de manera autónoma (Universidad Estatal de Milagro (UNEMI), [2023](#)). En las actividades de aula, esta competencia se materializó cuando los equipos discutían errores en sus programas, analizaban sus causas y elaboraban soluciones fundamentadas en estándares de desarrollo.

Un segundo resultado de aprendizaje alcanzado fue la capacidad para comunicarse y trabajar de manera efectiva en equipos multidisciplinarios. Este aprendizaje se desarrolló

a través de la planificación, ejecución y presentación de proyectos colaborativos en los que cada estudiante asumió roles específicos, integrando la teoría con la práctica.

De acuerdo con el MCI, la comunicación efectiva y el trabajo en equipo son competencias transversales que contribuyen a la construcción de conocimiento compartido, al fortalecimiento de la empatía y al liderazgo en contextos diversos (Universidad Estatal de Milagro (UNEMI), 2021). En la práctica, los estudiantes demostraron avances significativos al utilizar plataformas de control de versiones y entornos colaborativos, evidenciando mejoras tanto en la organización de tareas como en la exposición oral y escrita de los resultados.

El tercer resultado de aprendizaje consolidado fue la demostración de responsabilidad ética y compromiso con el aprendizaje autónomo y permanente. Este resultado se expresó en la reflexión de los estudiantes sobre el impacto social de las tecnologías que desarrollan, la gestión responsable del tiempo y la autogestión de su propio progreso académico.

Según el Modelo Educativo Institucional de la UNEMI, la formación ética constituye un eje transversal del currículo, orientado al ejercicio profesional responsable y al desarrollo sostenible (Universidad Estatal de Milagro (UNEMI), 2021). En este marco, la autonomía y la ética se entienden como pilares que permiten a los futuros ingenieros enfrentar los desafíos tecnológicos con conciencia social y capacidad de adaptación continua.

La relación entre estos resultados de aprendizaje y las competencias del perfil de egresado es directa y verificable. El pensamiento crítico y la resolución de problemas se asocian con la competencia de innovar y aplicar tecnologías emergentes, la colaboración y la comunicación efectiva se relacionan con la capacidad de liderar proyectos y trabajar en equipos multidisciplinarios, mientras que la ética y el aprendizaje autónomo se alinean con el compromiso institucional de formar profesionales responsables y comprometidos con el desarrollo sostenible. Esta trazabilidad, destacada por el MCI (Universidad Estatal de Milagro (UNEMI), 2023), garantiza la coherencia entre las experiencias formativas del aula y las metas de desempeño profesional esperadas al final de la carrera.

En síntesis, los resultados de aprendizaje alcanzados en la experiencia evidencian la efectividad del enfoque por competencias adoptado por la UNEMI. Los estudiantes no solo adquirieron conocimientos técnicos, sino que también fortalecieron habilidades cognitivas, comunicativas y éticas que sustentan su perfil profesional.

Este proceso demuestra que la integración entre docencia, práctica y currículo no es un ejercicio formal, sino una estrategia pedagógica que potencia la formación integral del

futuro ingeniero de software. En el siguiente apartado, estos aprendizajes se proyectarán en las actividades y evidencias que permiten verificar empíricamente la transformación educativa lograda en el aula.

3.3.4. Actividades y evidencias

El desarrollo de la experiencia en la asignatura Introducción a la Ingeniería de Software se sustentó en una secuencia de actividades planificadas intencionalmente para fortalecer las competencias y resultados de aprendizaje vinculados al perfil de egreso de la carrera. Estas actividades respondieron al principio de coherencia pedagógica entre la teoría y la práctica, promoviendo el aprendizaje activo, el trabajo colaborativo y la reflexión crítica como ejes centrales del proceso formativo.

De este modo, cada acción en el aula fue concebida no solo como una tarea académica, sino como una oportunidad para generar evidencias verificables de los aprendizajes alcanzados por los estudiantes.

Las actividades se estructuraron en tres momentos pedagógicos: exploración, aplicación y reflexión. En la fase de exploración, los estudiantes participaron en sesiones de diagnóstico y discusión guiada sobre los fundamentos de la ingeniería de software, identificando los problemas comunes en la gestión de proyectos tecnológicos. Estas dinámicas permitieron activar los conocimientos previos y contextualizar los desafíos que enfrentarían posteriormente. Las evidencias generadas en esta etapa fueron fichas de análisis individual, foros de discusión en el aula virtual y registros reflexivos, que mostraron la comprensión inicial de los conceptos básicos y la disposición para el trabajo colaborativo.

Durante la fase de aplicación, se implementaron actividades basadas en el aprendizaje activo y la simulación de entornos profesionales. Los estudiantes trabajaron en equipos para desarrollar un proyecto de software de pequeña escala, aplicando principios de análisis de requerimientos, diseño modular y control de versiones mediante GitHub.

Cada grupo asumió roles diferenciados -analista, programador, validador y líder de proyecto- con el fin de promover la corresponsabilidad y el liderazgo colaborativo. Entre las principales evidencias producidas se encuentran los repositorios de código en GitHub, los informes técnicos del proyecto, las actas de reuniones semanales y las rúbricas de evaluación de desempeño, que documentaron tanto los procesos como los productos del aprendizaje.

Estas evidencias resultaron especialmente relevantes porque permitieron observar la consolidación de los resultados de aprendizaje vinculados a la resolución de problemas, el trabajo en equipo y la comunicación efectiva. Tal como señalan Biggs y Tang (2011), la coherencia entre objetivos, actividades y evaluación es esencial para garantizar la validez del proceso formativo.

En este caso, la experiencia demostró que los proyectos colaborativos constituyen un medio eficaz para integrar los saberes técnicos y las competencias transversales del ingeniero de software, permitiendo que el aprendizaje sea significativo, observable y transferible.

Finalmente, en la fase de reflexión, se promovió la autoevaluación y la metacognición mediante la elaboración de bitácoras individuales de aprendizaje y sesiones de retroalimentación grupal. Los estudiantes analizaron los aciertos y dificultades encontradas durante el desarrollo del proyecto, reflexionando sobre la aplicación de los principios éticos, la responsabilidad en la gestión del tiempo y la toma de decisiones.

Estas evidencias, junto con las encuestas de satisfacción y las observaciones del docente, ofrecieron una visión integral del impacto de la experiencia en la formación de los participantes. Según Kolb (2015), la reflexión sobre la práctica permite transformar la experiencia vivida en conocimiento profundo, consolidando la autonomía y la capacidad crítica del aprendiz.

La pertinencia de las evidencias recogidas radica en que reflejan de manera tangible el cumplimiento de los resultados de aprendizaje propuestos en el plan curricular de la carrera y en el Marco Curricular Institucional de la UNEMI. Cada evidencia -ya sea un producto tecnológico, un registro reflexivo o un indicador de desempeño- constituye una muestra verificable de que los estudiantes avanzaron hacia el perfil profesional esperado: un ingeniero de software con pensamiento crítico, ética profesional, habilidades colaborativas y compromiso con la calidad del desarrollo tecnológico.

En conclusión, las actividades y evidencias descritas demuestran que la práctica docente fue diseñada con coherencia pedagógica y pertinencia curricular, asegurando la trazabilidad entre actividad → resultado → evidencia. Este enfoque permitió evidenciar que la experiencia no solo fortaleció los aprendizajes técnicos, sino también las competencias cognitivas, socioafectivas y éticas necesarias para la formación integral del futuro profesional. En el siguiente apartado, estas evidencias servirán de base para analizar la alineación curricular y el grado de correspondencia entre la práctica sistematizada y el modelo educativo institucional de la UNEMI.

3.3.5. Reflexión sobre la alineación curricular

Reflexionar sobre la alineación curricular implica reconocer la importancia de que cada práctica docente se integre de manera coherente al perfil de egreso, a los resultados de aprendizaje y a los principios formativos de la carrera. En el caso de la experiencia desarrollada en la asignatura Introducción a la Ingeniería de Software, esta alineación fue clave para garantizar que las estrategias implementadas no fueran acciones aisladas, sino componentes articulados del proceso formativo que caracteriza al ingeniero de software de la Universidad Estatal de Milagro.

En coherencia con lo que plantea el Marco Curricular Institucional (Universidad Estatal de Milagro (UNEMI), 2023), la práctica docente debe promover aprendizajes significativos y transferibles, alineados con un currículo flexible y orientado por competencias, donde la docencia se conciba como un espacio de innovación y reflexión continua.

La experiencia permitió constatar una sólida correspondencia entre las actividades de aula y los propósitos curriculares de la carrera. Las estrategias de aprendizaje activo, la resolución colaborativa de problemas y la integración de herramientas tecnológicas (como GitHub y entornos de simulación) evidenciaron la conexión directa con las competencias del perfil de egreso relacionadas con la comunicación efectiva, la ética profesional y la innovación tecnológica.

Esta coherencia refleja lo que Biggs y Tang (2011) denominan alineación constructiva: la necesidad de que los objetivos, las actividades y la evaluación respondan de forma integrada a los resultados de aprendizaje esperados. En este sentido, la experiencia fortaleció la formación integral del estudiante al fomentar tanto las competencias técnicas propias de la ingeniería como las socioemocionales y metacognitivas que demanda la práctica profesional contemporánea.

Entre las principales fortalezas observadas destaca la pertinencia metodológica de las actividades diseñadas. La secuencia de exploración, aplicación y reflexión permitió avanzar de lo conceptual a lo práctico, haciendo visible la progresión del aprendizaje. Los proyectos colaborativos y las bitácoras de autoevaluación demostraron que los estudiantes no solo comprendieron los fundamentos de la disciplina, sino que también desarrollaron autonomía, pensamiento crítico y capacidad de autorregulación.

Como señalan Barnett y Jackson (2019), la educación superior debe preparar a los estudiantes para gestionar la complejidad y la incertidumbre del conocimiento profesional, promoviendo la reflexión crítica sobre su propio aprendizaje. Desde esta perspectiva,

la alineación curricular se convierte en un proceso dinámico que articula la formación técnica con la capacidad de adaptación y el pensamiento ético.

No obstante, también se identificaron algunas brechas y tensiones en el proceso. Una de ellas fue la necesidad de fortalecer la conexión entre los resultados de aprendizaje del primer semestre y los desafíos de cursos posteriores, especialmente en la aplicación sistemática de metodologías ágiles y en la evaluación continua de proyectos.

Esta situación coincide con lo que expone Zabalza (2019), quien plantea que los currículos universitarios deben revisarse periódicamente para mantener su coherencia horizontal y vertical, evitando fragmentaciones en la formación por competencias.

Asimismo, se evidenció que la carga académica y el tiempo destinado a la tutoría personalizada resultaron limitantes para acompañar a todos los grupos con la profundidad requerida. Estas brechas sugieren la necesidad de reforzar la integración entre docentes de distintas asignaturas, promoviendo espacios de co-docencia y articulación interdisciplinaria.

A nivel institucional, el ejercicio de vincular la práctica con el currículo reafirma la importancia de comprender la docencia como una práctica reflexiva. La Universidad Estatal de Milagro (UNEMI) (2023) plantea que la alineación curricular no debe entenderse como un cumplimiento formal, sino como una estrategia para garantizar la calidad académica y la pertinencia social de la formación.

En este sentido, el proceso de sistematización se convierte en una herramienta para revisar críticamente las propias decisiones pedagógicas, reconocer aciertos y proyectar mejoras. Entre los aprendizajes más significativos derivados de esta reflexión se encuentra la convicción de que la enseñanza en ingeniería debe integrar la innovación tecnológica con una mirada humanista y ética, fortaleciendo la relación entre la práctica profesional y la responsabilidad social.

En conclusión, la experiencia desarrollada estuvo sólidamente alineada con el currículo de la carrera, contribuyendo a la consolidación de un modelo pedagógico basado en competencias y en la formación integral del estudiante. Sin embargo, esta reflexión invita también a mantener una actitud crítica frente al propio diseño curricular, promoviendo una actualización constante y un diálogo permanente entre las prácticas docentes y las demandas del entorno profesional.

De este modo, la alineación curricular se convierte en un ejercicio de mejora continua que reafirma el compromiso docente con la calidad educativa y la innovación formativa,

principios que sustentan la misión de la UNEMI y el propósito de formar ingenieros capaces de transformar su realidad desde el conocimiento, la ética y la creatividad.

3.4. Transición hacia la operacionalización estratégica

El cierre del Módulo 3 permitió consolidar la coherencia entre la práctica pedagógica y el perfil de egreso del Ingeniero en Software, evidenciando cómo las competencias curriculares y los resultados de aprendizaje se concretan en experiencias formativas que integran el razonamiento lógico con la calidad técnica del código. Esta articulación entre teoría, práctica y reflexión sentó las bases para comprender que el aprendizaje de la programación no se limita a la ejecución de algoritmos, sino que implica la formación de un pensamiento estructurado, crítico y ético.

A partir de este marco curricular, el proceso de sistematización se orienta ahora hacia la **operacionalización estratégica** de la experiencia, donde las acciones didácticas se transforman en un entramado de estrategias deliberadas que sustentan el logro de los resultados formativos.

El siguiente apartado aborda el **ecosistema estratégico** de la experiencia, entendido como el conjunto de estrategias nucleares, de soporte y de contingencia que dieron vida a la propuesta pedagógica. Desde los talleres de análisis algorítmico hasta las sesiones de refactorización y revisión entre pares, se analizará cómo cada decisión metodológica respondió a un propósito formativo y cómo estas estrategias se articularon para sostener una “ingeniería didáctica” coherente con las competencias del currículo. Este tránsito marca el paso de la fundamentación curricular a la descripción operativa de la práctica, abriendo un espacio para comprender cómo las acciones concretas en el aula materializan los principios pedagógicos y consolidan una enseñanza innovadora en ingeniería de software.

3.4.1. Recomendaciones para profundizar

- **Evitar redundancias:** No repetir descripciones de competencias o resultados ya desarrollados en el Módulo 3; centrarse en cómo las estrategias transforman esos elementos en acciones concretas.
- **Mantener claridad en la transición:** Usar conectores como “a partir de”, “en esta nueva fase” o “el siguiente módulo” para reforzar la idea de continuidad narrativa.

- **Usar el puente como bisagra narrativa:** Presentar este texto como un punto de enlace entre el qué (currículo y resultados) y el cómo (estrategias y métodos), destacando el paso del plano teórico al operativo dentro del proceso de sistematización.

3.4.2. Clase 1: Estrategias núcleo en acción

Introducción a las estrategias núcleo

Las estrategias nucleares representan el eje articulador de toda experiencia educativa innovadora, pues configuran las acciones que garantizan la coherencia entre la intención formativa y los resultados alcanzados. En la experiencia desarrollada en la asignatura Algoritmos y Lógica de Programación, estas estrategias fueron concebidas no solo como métodos de enseñanza, sino como estructuras operativas que orientaron el aprendizaje hacia la comprensión profunda y la calidad profesional. Desde esta perspectiva, la refactorización pedagógica, los talleres de análisis algorítmico, el uso de herramientas visuales de programación y la revisión entre pares constituyeron los pilares que sustentaron el éxito del proceso formativo.

Estas prácticas materializan el principio de aprendizaje activo, reflexivo y colaborativo, en el que el estudiante asume un rol protagónico en la construcción del conocimiento, coherente con las recomendaciones de Grover y Pea (2020) y Jara (2018a) sobre la centralidad de la experiencia significativa en la formación universitaria.

Refactorización pedagógica: mejorar pensando

La primera estrategia nuclear fue la refactorización pedagógica, entendida como un proceso sistemático de revisión, análisis y mejora del código desde una lógica educativa. Inspirada en los principios del desarrollo ágil, esta estrategia permitió que los estudiantes reescribieran sus algoritmos tras la retroalimentación docente, enfocándose en criterios de legibilidad, modularidad y documentación. De esta manera, el código se convirtió en una herramienta de reflexión sobre el propio pensamiento, en un ejercicio de metacognición donde cada corrección implicaba una comprensión más profunda de la lógica subyacente (R. Zanatta & da Silva, 2022).

Por ejemplo, durante los laboratorios, los estudiantes comparaban sus primeras versiones de pseudocódigos con las versiones refactorizadas, identificando errores de estructura o repetición innecesaria de instrucciones. Este proceso fomentó el desarrollo de habili-

dades analíticas y una conciencia sobre la calidad del trabajo técnico, tal como proponen Ahmad y Hashim (2020) al destacar la refactorización como un medio de aprendizaje iterativo. En este contexto, la mejora del código se transformó en una metáfora del aprendizaje mismo: continuo, autocrítico y orientado a la excelencia.

Talleres de análisis algorítmico: pensar antes de programar

La segunda estrategia clave consistió en la implementación de talleres de análisis algorítmico, diseñados para fortalecer el pensamiento lógico y computacional a partir de problemas contextualizados. Estos espacios formativos promovieron la capacidad de los estudiantes para descomponer situaciones reales en sus componentes esenciales, identificar variables relevantes y establecer relaciones lógicas entre ellas antes de proceder a la codificación.

En la práctica, los talleres se desarrollaron mediante ejercicios guiados donde los estudiantes debían representar gráficamente los pasos de un algoritmo, justificando sus decisiones y secuencias. Esta metodología permitió pasar de la memorización de estructuras a la comprensión funcional del proceso algorítmico, en coherencia con lo planteado por Lodi y Martini (2021), quienes sostienen que el pensamiento computacional emerge de la capacidad de modelar y abstraer la realidad. Además, los talleres sirvieron como espacios de diagnóstico y retroalimentación inmediata, facilitando la detección de errores conceptuales y el fortalecimiento del razonamiento estructurado.

Uso de herramientas visuales: conectar teoría y práctica

El empleo de herramientas visuales como Flowgorithm y PSeInt constituyó otra estrategia nuclear que potenció la comprensión del flujo lógico de los algoritmos. Estas plataformas permitieron visualizar la secuencia de ejecución de los procesos, facilitando la comprensión de símbolos de decisión, conexión y proceso, lo que ayudó a los estudiantes a vincular la teoría con la práctica.

Su implementación generó un entorno de aprendizaje interactivo, donde los errores de diseño se detectaban de forma inmediata, promoviendo la autonomía y la exploración. Grover y Pea (2020) destacan que el aprendizaje de la programación se fortalece cuando los entornos visuales permiten al estudiante observar la relación entre el razonamiento lógico y el comportamiento del programa. En este sentido, el uso de herramientas visuales no solo mejoró la comprensión conceptual, sino que también incrementó la motivación y

el compromiso de los participantes, al ofrecer una experiencia tangible del pensamiento computacional.

Revisión entre pares: aprender con otros

La revisión entre pares (peer code review) se consolidó como una estrategia central que integró la dimensión técnica con la ética y la colaboración profesional. A través de este ejercicio, los estudiantes revisaban los códigos de sus compañeros, identificaban errores, proponían mejoras y justificaban sus observaciones, replicando dinámicas propias de los entornos profesionales de ingeniería de software.

Esta estrategia promovió una cultura de aprendizaje compartido, basada en la crítica constructiva y la responsabilidad colectiva. Según Petre y van der Hoek (2021), la revisión de código fomenta la transparencia y la calidad en los procesos de desarrollo, al tiempo que fortalece las habilidades comunicativas y la autoconfianza técnica. En la experiencia, las evidencias recolectadas —comentarios escritos, fichas de retroalimentación y versiones revisadas del código— demostraron que los estudiantes no solo mejoraron la calidad técnica de sus productos, sino también su capacidad de argumentar y justificar decisiones de diseño, consolidando así una competencia profesional integral.

Importancia de las estrategias núcleo

Estas estrategias núcleo se convirtieron en el motor que articuló la experiencia educativa, garantizando que el aprendizaje no se redujera a la ejecución técnica, sino que implicara comprensión, reflexión y mejora continua. La refactorización, los talleres, el uso de herramientas visuales y la revisión entre pares generaron un ecosistema de aprendizaje dinámico y coherente con los principios de la ingeniería de software. En conjunto, permitieron que los estudiantes se apropiaran del conocimiento a través de la práctica guiada, el error consciente y la colaboración, asegurando la alineación entre los propósitos curriculares, los resultados de aprendizaje y el perfil de egreso (Yin, 2014); (R. Stake, 1995).

Cierre hacia las estrategias de soporte

La efectividad de estas estrategias nucleares se potenció gracias a un conjunto de estrategias de soporte, centradas en la mediación docente, la retroalimentación formativa y el acompañamiento continuo. Estas estrategias complementarias, que se desarrollarán en

el siguiente apartado, permitieron sostener el proceso de aprendizaje, garantizando la continuidad entre la exploración individual, el trabajo colaborativo y la validación curricular. De este modo, la experiencia consolidó una ingeniería didáctica orientada a la formación de ingenieros reflexivos, críticos y comprometidos con la calidad del desarrollo de software.

Recomendaciones para profundizar

- **Evitar redundancias** con los apartados anteriores, priorizando una narrativa centrada en la acción y no en la descripción de resultados.
- **Mantener claridad en la transición** hacia las estrategias de soporte, explicando cómo estas complementan las estrategias núcleo.
- **Equilibrar teoría y práctica**, asegurando que cada estrategia descrita esté acompañada de ejemplos observables y respaldo conceptual.
- **Usar lenguaje operativo y reflexivo**, destacando cómo las decisiones pedagógicas se traducen en aprendizajes verificables y sostenibles.

3.4.3. Clase 2: Estrategias de soporte aplicadas

La implementación de estrategias de soporte en una experiencia educativa innovadora constituye un pilar esencial para garantizar la efectividad y sostenibilidad del proceso formativo. En el marco del ecosistema estratégico, los soportes actúan como estructuras de acompañamiento que fortalecen las estrategias núcleo, brindando las condiciones pedagógicas, tecnológicas y reflexivas necesarias para que el aprendizaje se consolide de forma integral (Flick, 2014; (Jara, 2018a)).

Estas estrategias no solo facilitan la comprensión de los contenidos, sino que promueven la autonomía del estudiante, la autorregulación del aprendizaje y la consolidación de competencias profesionales. En la asignatura **Algoritmos y Lógica de Programación**, los soportes implementados se orientaron a integrar la teoría con la práctica, asegurando que los estudiantes no se limitaran a reproducir estructuras algorítmicas, sino que comprendieran su sentido y aplicabilidad en contextos reales de la ingeniería de software.

Entre los principales **soportes aplicados** se destacan:

1. **Acompañamiento docente reflexivo**, enfocado en guiar el proceso de aprendizaje desde la comprensión conceptual hasta la práctica aplicada.
2. **Retroalimentación formativa y continua**, dirigida a promover la mejora progresiva del código y la autorregulación cognitiva.
3. **Uso de herramientas digitales (Flowgorithm y PSeInt)**, como medios de visualización y validación de estructuras algorítmicas.
4. **Trabajo colaborativo y revisión entre pares**, que fortalecieron el aprendizaje cooperativo y la responsabilidad compartida.
5. **Documentación reflexiva del aprendizaje**, destinada a consolidar la metacognición y la capacidad de explicar el propio proceso de desarrollo.

El **acompañamiento docente reflexivo** fue un soporte esencial que aseguró la coherencia del proceso formativo. A través de tutorías personalizadas y orientación constante durante los laboratorios, el docente actuó como mediador cognitivo, ayudando a los estudiantes a comprender los errores y a reformular sus estrategias de resolución. Este acompañamiento, entendido como guía constructiva y no directiva, permitió vincular la teoría con la práctica, generando confianza y autonomía en los estudiantes (Vygotsky, 1978; Bruner, 1997). La presencia activa del docente facilitó la interiorización de los conceptos, promoviendo una comprensión profunda del pensamiento algorítmico.

La **retroalimentación formativa y continua** fue otro soporte clave para la mejora del aprendizaje. Después de cada entrega de pseudocódigos, los estudiantes recibían observaciones orientadas no solo a corregir errores, sino a comprender el razonamiento subyacente. Esta práctica permitió instaurar una cultura de evaluación formativa que, según Black y Wiliam (2018), fomenta la autoevaluación, la reflexión y la mejora continua. En este contexto, la retroalimentación funcionó como una herramienta dialógica que fortaleció la autorregulación y el aprendizaje autónomo, al tiempo que aumentó la calidad de las producciones de código.

El **uso de herramientas digitales como Flowgorithm y PSeInt** proporcionó un soporte tecnológico que facilitó la visualización de la lógica algorítmica. Estas plataformas permitieron a los estudiantes observar en tiempo real la ejecución de sus algoritmos, identificar errores estructurales y reforzar la comprensión de los símbolos de proceso, decisión y conexión. De acuerdo con García-Peñalvo y Mendes (2018), las herramientas digitales

aplicadas al pensamiento computacional potencian la representación mental de la secuencia lógica, al vincular lo abstracto con lo visual. Su incorporación en la experiencia permitió consolidar aprendizajes significativos al traducir la lógica en acción.

Por su parte, **el trabajo colaborativo y la revisión entre pares (peer code review)** constituyeron un soporte fundamental para fortalecer la dimensión ética y profesional del aprendizaje. Estas prácticas, inspiradas en enfoques de aprendizaje cooperativo (Johnson & Johnson, 2019), promovieron la co-construcción del conocimiento y la responsabilidad compartida. A través de la revisión mutua de códigos, los estudiantes aprendieron a argumentar sus decisiones, justificar su estructura y aceptar sugerencias de mejora. Este proceso no solo reforzó la calidad del producto final, sino que cultivó habilidades sociales y comunicativas esenciales en la ingeniería de software contemporánea.

Finalmente, la **documentación reflexiva del aprendizaje** sirvió como soporte metacognitivo, permitiendo a los estudiantes analizar su propio progreso y reconocer los cambios en su comprensión a lo largo del proceso. Las bitácoras y reportes finales no solo registraron las versiones del código, sino también las reflexiones sobre los aprendizajes alcanzados, errores superados y decisiones tomadas. Esta práctica, en línea con **sch<empty citation>**, consolidó una actitud reflexiva y crítica frente al propio hacer, reforzando la formación de un profesional capaz de aprender de su experiencia y proyectarla en contextos futuros.

Los **soportes aplicados** potenciaron directamente las estrategias núcleo de la experiencia. La refactorización pedagógica se fortaleció con la retroalimentación y el acompañamiento docente, asegurando la comprensión de los principios de calidad del código. Los talleres de análisis algorítmico se consolidaron mediante el uso de herramientas digitales y la orientación guiada, que facilitaron el paso de la teoría a la práctica. Asimismo, la revisión entre pares se amplió gracias al trabajo colaborativo y la documentación reflexiva, promoviendo un aprendizaje ético y compartido. En conjunto, estos soportes generaron un ambiente de aprendizaje integrado, donde la técnica y la reflexión convergieron en un proceso continuo de mejora y profesionalización.

La evidencia de su efectividad se reflejó en la calidad de los productos elaborados, la evolución del razonamiento de los estudiantes y la mejora progresiva en la comprensión algorítmica y estructural del código. Las observaciones docentes, los informes reflexivos y las versiones comparativas de los programas evidenciaron un tránsito claro desde la ejecución mecánica hacia la comprensión profunda y crítica. Estos resultados confirman que

los soportes no solo acompañaron el proceso, sino que lo hicieron sostenible, coherente y transferible a otros contextos educativos de la ingeniería.

Recomendaciones para profundizar

- Evitar confundir los soportes con las estrategias núcleo: los primeros habilitan, sostienen y fortalecen el proceso, mientras que las segundas lo estructuran.
- Mostrar siempre la función habilitadora de cada soporte y su vínculo con las competencias o resultados de aprendizaje.
- Equilibrar la descripción práctica con su fundamento teórico, integrando autores como Schön (1983), Black y Wiliam (2018) o García-Peñalvo (2018) para sustentar el papel reflexivo y tecnológico de los soportes.
- Cerrar el apartado destacando la sostenibilidad de la innovación educativa gracias a la interacción entre estrategias núcleo y soportes pedagógicos.

3.4.4. Clase 3: Estrategias de contingencia desplegadas

La incorporación de las contingencias en una sistematización de experiencias educativas resulta fundamental para comprender la naturaleza dinámica y adaptativa de los procesos de innovación docente. Mostrar los imprevistos y las estrategias implementadas para enfrentarlos no debilita la experiencia, sino que la enriquece, pues evidencia la capacidad de resiliencia pedagógica y de toma de decisiones situada del docente (R. E. Stake, 2020); (Yin, 2014).

En contextos complejos como la enseñanza de la programación, donde confluyen factores tecnológicos, cognitivos y emocionales, la flexibilidad y la capacidad de ajuste se convierten en componentes esenciales del ecosistema de aprendizaje (Jara, 2018a). En este sentido, las contingencias no solo son respuestas a las dificultades emergentes, sino también oportunidades de mejora que fortalecen la coherencia, la equidad y la sostenibilidad del proceso educativo.

Imprevistos enfrentados y contingencias aplicadas

Durante la implementación de la experiencia se presentaron varios desafíos que demandaron respuestas inmediatas y estratégicas. El primero de ellos fue la **limitación tec-**

nológica, ya que algunos estudiantes no contaban con equipos adecuados o conectividad estable para ejecutar herramientas como *Flowgorithm* y *PSeInt*.

Para resolverlo, se aplicó una **estrategia de flexibilización didáctica**, basada en el principio de accesibilidad educativa (Flick, 2014). Se elaboraron guías impresas, se permitieron entregas diferidas y se promovió el uso de laboratorios compartidos. Esta contingencia permitió mantener la equidad en la participación y garantizar que las condiciones materiales no se convirtieran en una barrera para el desarrollo de competencias digitales.

Un segundo imprevisto fue la **diversidad de niveles de conocimiento previo**. Mientras algunos estudiantes tenían experiencia básica en programación, otros se enfrentaban por primera vez a la lógica algorítmica. Para atender esta heterogeneidad, se implementó una **diferenciación pedagógica progresiva** Tomlinson (2017), mediante el diseño de ejercicios escalonados y tutorías personalizadas.

Además, se promovió la **mentoría entre pares**, de modo que los estudiantes con mayor dominio apoyaran a sus compañeros, fortaleciendo la colaboración y la construcción colectiva del conocimiento.

El tercer desafío fue la **resistencia inicial al trabajo colaborativo y a la revisión entre pares**. Algunos estudiantes mostraban desconfianza al exponer sus códigos o recibir observaciones. Para contrarrestar esta situación, se aplicaron estrategias de **clima socio-emocional y corresponsabilidad** (Johnson & Johnson, 2019), incluyendo dinámicas de reconocimiento mutuo y la elaboración de un código ético de retroalimentación. Este proceso transformó la crítica en una herramienta de mejora colectiva y consolidó una cultura de respeto y aprendizaje compartido.

Un cuarto imprevisto estuvo relacionado con la **gestión del tiempo didáctico**. Las fases de refactorización y retroalimentación requerían más tiempo del planificado. En respuesta, se aplicó una **reorganización curricular flexible** (Yin, 2014); (Fullan & Quinn, 2017), priorizando actividades de mayor impacto cognitivo y reestructurando los tiempos de laboratorio. De este modo, se favoreció la profundización conceptual y la calidad del trabajo sobre la cantidad de ejercicios completados.

Finalmente, surgió una dificultad en la **documentación reflexiva**. Muchos estudiantes tenían escasa experiencia en la escritura técnica y argumentativa. Para abordarla, se desarrollaron **microtalleres de escritura académica aplicada**, inspirados en Carlino (2005) y Hyland (2009). Estas sesiones incluyeron ejemplos guiados, rúbricas y ejercicios de redacción explicativa sobre los algoritmos, fortaleciendo la capacidad metacognitiva y comunicativa de los participantes.

Resultados garantizados gracias a las contingencias

Las contingencias aplicadas no solo mitigaron los efectos de los imprevistos, sino que garantizaron la consecución de los resultados de aprendizaje. La flexibilización tecnológica y el acompañamiento personalizado permitieron mantener el desarrollo del **pensamiento lógico y computacional**, evidenciado en la elaboración de algoritmos coherentes y funcionales incluso en condiciones adversas. La diferenciación pedagógica y la mentoría entre pares sostuvieron la **aplicación de buenas prácticas de programación**, fomentando la autonomía y la responsabilidad en el diseño de código de calidad.

Asimismo, las estrategias de construcción de confianza y revisión colaborativa fortalecieron la **comunicación técnica y el trabajo en equipo**, generando un entorno de cooperación y retroalimentación ética. Por su parte, los talleres de escritura reflexiva consolidaron la **capacidad de argumentación y documentación profesional**, mostrando que la escritura puede ser una herramienta de pensamiento en la formación en ingeniería (Carlino, 2005). En conjunto, estas contingencias permitieron mantener la alineación entre las competencias curriculares, los resultados esperados y las evidencias de aprendizaje significativo.

Reflexión sobre los aprendizajes derivados

El despliegue de estrategias de contingencia dejó aprendizajes valiosos tanto para la práctica docente como para la gestión pedagógica institucional. En primer lugar, evidenció que la innovación no depende únicamente del diseño inicial, sino de la capacidad de **ajuste continuo y respuesta contextual** (Fullan & Quinn, 2017); (Jara, 2018a). En segundo lugar, mostró que los imprevistos pueden convertirse en fuentes de mejora si se asumen con una lógica reflexiva y colaborativa.

La experiencia permitió al docente reconocer la importancia del acompañamiento emocional, la escritura reflexiva y la planificación flexible como pilares de sostenibilidad pedagógica (Yin, 2014); (Johnson & Johnson, 2019).

Finalmente, este proceso reafirmó que la enseñanza de la programación, más allá de los contenidos técnicos, implica formar en resiliencia, adaptabilidad y pensamiento crítico. Las contingencias gestionadas con coherencia ética y pedagógica fortalecen no solo los aprendizajes inmediatos, sino también la cultura institucional de innovación y mejora continua (Tomlinson, 2017); (Flick, 2014); (Petre & van der Hoek, 2021).

3.4.5. Clase 4: Arquitectura del ecosistema estratégico

La arquitectura del ecosistema pedagógico diseñado en la experiencia se construyó bajo una lógica de interdependencia entre **estrategias núcleo, soportes y contingencias**, configurando un sistema educativo flexible, reflexivo y sostenible. Cada componente cumplió una función específica, pero su verdadero valor emergió de la sinergia que se generó al interactuar entre sí. Las estrategias núcleo actuaron como el **motor estructural** del proceso, orientado al desarrollo del pensamiento lógico, la calidad del código y la ética profesional; los soportes funcionaron como el **andamiaje pedagógico** que sostuvo y potenció dichas estrategias; mientras que las contingencias representaron el **mecanismo adaptativo**, que permitió mantener la coherencia y continuidad del aprendizaje ante situaciones imprevistas (Yin, 2014); (Fullan & Quinn, 2017).

Esta lógica de conexión evidencia que la innovación educativa no se limita a la planificación inicial, sino que reside en la capacidad del docente para construir un sistema que aprenda, se ajuste y evolucione con sus propios desafíos.

Relato de la arquitectura del ecosistema

El diseño de este ecosistema partió de la premisa de que **la enseñanza de la programación requiere un equilibrio entre estructura y flexibilidad**. En el núcleo del proceso se establecieron estrategias activas —como los talleres de análisis algorítmico, la refactorización del código y la revisión entre pares—, que promovieron la comprensión conceptual y el aprendizaje autónomo.

Estas estrategias se articularon directamente con los resultados de aprendizaje curriculares, generando un flujo constante entre la teoría y la práctica. A su alrededor, se incorporaron **estrategias de soporte** —como la retroalimentación formativa, el acompañamiento docente y el trabajo colaborativo— que aseguraron la continuidad del aprendizaje y fomentaron un clima de confianza y participación (Johnson & Johnson, 2019); (Carlino, 2005).

A su vez, las **estrategias de contingencia** funcionaron como mecanismos de resiliencia ante las limitaciones y desafíos que surgieron. La flexibilización didáctica, la diferenciación pedagógica y la reorganización del tiempo docente se integraron para garantizar la equidad y la sostenibilidad del proceso. Lejos de ser respuestas aisladas, estas medidas se convirtieron en **acciones estratégicas complementarias** que fortalecieron la estructura

base del ecosistema, demostrando que la adaptabilidad es parte esencial de una pedagogía innovadora (Tomlinson, 2017); (Flick, 2014).

Esta interdependencia generó un modelo circular de aprendizaje donde cada elemento retroalimentaba al otro. Las estrategias núcleo impulsaban la comprensión; los soportes aseguraban su implementación efectiva; y las contingencias protegían la continuidad del proceso. Así, la experiencia se configuró como un **ecosistema autorregulado**, capaz de mantener su coherencia interna y su orientación hacia los resultados de aprendizaje incluso ante condiciones adversas.

Explicación del diagrama visual acompañante

El **diagrama del ecosistema** representa una estructura circular, inspirada en la metáfora del “jardín algorítmico”. En el centro se ubican las **estrategias núcleo**, simbolizadas como el tronco o tallo del sistema, ya que son el eje que sostiene el crecimiento formativo: talleres de análisis algorítmico, laboratorios digitales, refactorización pedagógica, revisión entre pares y evaluación reflexiva.

Rodeando este núcleo, las **estrategias de soporte** aparecen como las raíces que nutren y estabilizan el proceso: acompañamiento docente, retroalimentación continua, trabajo colaborativo y documentación reflexiva. Estas raíces garantizan la transferencia de nutrientes —en términos pedagógicos, de conocimientos, valores y habilidades— entre el suelo conceptual y la práctica formativa.

Finalmente, en la capa exterior del diagrama se encuentran las **estrategias de contingencia**, representadas como hojas y ramas flexibles que responden a los cambios del entorno: flexibilización didáctica, diferenciación por niveles, gestión del tiempo y fortalecimiento de la escritura técnica. Su disposición periférica simboliza la capacidad del ecosistema para adaptarse y mantener su vitalidad ante las inclemencias del contexto educativo.

En conjunto, el diagrama no solo ilustra la organización funcional del sistema, sino también su carácter orgánico, mostrando cómo cada parte crece y se transforma a partir de la interacción constante con las demás. Este enfoque refleja lo que Fullan y Quinn (2017) denominan “coherencia adaptativa”, entendida como la capacidad de los sistemas educativos de mantenerse estables mientras se transforman.

3.4.6. Síntesis final: El ecosistema como sistema vivo

El ecosistema pedagógico construido en esta experiencia se consolidó como **un sistema vivo, dinámico y en equilibrio continuo**, donde la enseñanza y el aprendizaje se retroalimentan de manera recíproca. Las estrategias núcleo ofrecieron estructura y propósito; los soportes brindaron estabilidad y acompañamiento; y las contingencias aportaron resiliencia y capacidad de respuesta. En su conjunto, estos componentes conformaron un entramado coherente que permitió sostener los resultados de aprendizaje y fortalecer las competencias profesionales del futuro Ingeniero en Software.

Esta arquitectura demuestra que **la innovación educativa surge de la interconexión entre planificación, acción y reflexión**. Enseñar programación se transformó así en un ejercicio de diseño sistémico, donde cada decisión docente —por mínima que parezca— contribuye a un propósito mayor: formar profesionales críticos, éticos y creativos. Como en un jardín algorítmico, cada estrategia fue una semilla que, al nutrirse de los soportes y resistir las contingencias, floreció en un aprendizaje sostenible y significativo (Jara, 2018a); (Grover & Pea, 2020); (M. Zanatta & da Silva, 2022).

3.5. Transición hacia la evaluación

Tras el desarrollo del ecosistema estratégico, conformado por estrategias núcleo, de soporte y de contingencia, la experiencia pedagógica en Algoritmos y Lógica de Programación alcanzó un nivel de coherencia integral entre la teoría, la práctica y la reflexión. Estas estrategias, orientadas al fortalecimiento del pensamiento lógico, la calidad del código y la colaboración ética, permitieron construir un proceso formativo dinámico, inclusivo y contextualizado. Concluida esta fase operativa, el siguiente paso consiste en **evaluar de manera sistemática su efectividad**, a fin de comprender en qué medida dichas acciones contribuyeron al logro de los resultados de aprendizaje y a la consolidación de las competencias del perfil de egreso del Ingeniero en Software.

En este sentido, el **Módulo 5** marca el tránsito desde la descripción de la práctica hacia su análisis evaluativo, donde los instrumentos, indicadores y evidencias adquieren un papel central. Evaluar las estrategias implementadas no se limita a medir logros cuantitativos, sino a validar la coherencia, pertinencia y sostenibilidad del modelo pedagógico construido.

Como plantean Yin (2014) y Jara (2018a), la evaluación otorga **validez, credibilidad y transferibilidad** a las experiencias innovadoras, transformando la práctica en conocimiento sistematizado y útil para otros contextos educativos. Así, la evaluación se presenta no como un cierre, sino como una oportunidad para aprender de la experiencia, fortalecer la toma de decisiones docentes y proyectar mejoras continuas en el currículo y en las metodologías de enseñanza de la programación.

3.5.1. Recomendaciones para profundizar

- Mantener **claridad en la transición**, evitando repetir descripciones del ecosistema estratégico y centrando el foco en la lógica evaluativa.
- Enfatizar que la evaluación es una **extensión natural del proceso de sistematización**, no una etapa aislada.
- Presentar los instrumentos e indicadores como **puentes entre la práctica y la evidencia**, mostrando cómo permiten medir, comprender y validar los logros del proceso.
- Usar este puente como **bisagra narrativa**, conectando las acciones implementadas con los mecanismos que garantizarán su análisis riguroso y su impacto comprobable.

3.5.2. Clase 1: Instrumentos de evaluación aplicados

La evaluación constituye una dimensión clave en toda experiencia educativa innovadora, pues permite otorgar validez, credibilidad y transferibilidad a los procesos formativos (R. Stake, 1995). En el marco de esta sistematización, los instrumentos de evaluación se configuraron como herramientas esenciales para recoger evidencias del desarrollo de competencias cognitivas, técnicas y reflexivas, articulando así el ecosistema pedagógico previamente descrito.

Tal como plantea Casanova (1999), la evaluación formativa debe concebirse como un proceso de acompañamiento y mejora, más que como un simple juicio de resultados. En este sentido, los instrumentos aplicados en la experiencia no solo midieron el logro de los aprendizajes, sino que también favorecieron la reflexión, la autoevaluación y la mejora continua del proceso educativo.

Los instrumentos de evaluación utilizados fueron cinco: rúbricas de observación y evaluación del código, bitácoras reflexivas del estudiante, registros de revisión entre pares (Peer Review Logs), entrevistas o cuestionarios de retroalimentación y portafolio de evidencias.

Cada uno de ellos aportó una perspectiva complementaria del aprendizaje, permitiendo triangular la información recogida y consolidar una visión integral del proceso. Esta diversidad instrumental, inspirada en los planteamientos de Scriven (1991) sobre la evaluación como juicio fundamentado, garantizó que las valoraciones no se limitaran a un enfoque técnico, sino que integraran también las dimensiones ética, comunicativa y metacognitiva del aprendizaje.

Las rúbricas de observación y evaluación del código fueron diseñadas para valorar indicadores técnicos asociados a la legibilidad, modularidad, coherencia lógica y documentación de los algoritmos elaborados por los estudiantes. Su aplicación se realizó tanto en sesiones de laboratorio como en los espacios de revisión entre pares, permitiendo una evaluación objetiva y transparente.

Estas rúbricas se convirtieron en instrumentos formativos al orientar la mejora continua y la autoexigencia profesional, tal como propone Casanova (1999) al destacar que la evaluación debe ser también una guía para el aprendizaje. Las evidencias generadas —informes rubricados y comparaciones entre versiones de código— demostraron avances progresivos en la calidad técnica y conceptual del trabajo estudiantil.

Las bitácoras reflexivas del estudiante se implementaron como un espacio de autorregulación y metacognición. En ellas, los participantes registraron sus percepciones sobre los desafíos enfrentados, las decisiones tomadas y las estrategias de resolución aplicadas.

Según Carlino (2005), la escritura académica permite transformar la práctica en conocimiento, al hacer visible el proceso de pensamiento que subyace al aprendizaje. Estas bitácoras produjeron evidencias narrativas del desarrollo cognitivo y reflexivo, mostrando cómo los estudiantes vinculaban la teoría algorítmica con su práctica técnica, fortaleciendo así la comprensión y la autonomía.

Los registros de revisión entre pares (Peer Review Logs) funcionaron como un medio de evaluación colaborativa y ética. En ellos, los estudiantes documentaron comentarios técnicos, observaciones constructivas y sugerencias de mejora sobre el trabajo de sus compañeros.

Su aplicación sistemática promovió la comunicación profesional, la empatía y la responsabilidad compartida. En línea con Johnson y Johnson (2019), este tipo de evaluación

fomenta el aprendizaje cooperativo y la conciencia crítica, al situar al estudiante en un rol activo de coevaluador. Las fichas de revisión se constituyeron en evidencias del desarrollo de habilidades comunicativas, éticas y técnicas, esenciales en la formación de un ingeniero en software.

Las entrevistas y cuestionarios de retroalimentación complementaron la evaluación desde una perspectiva perceptiva, al recoger la opinión de los estudiantes sobre la utilidad y efectividad de las estrategias pedagógicas implementadas. Su aplicación al cierre de cada unidad permitió ajustar tiempos, metodologías y recursos, fortaleciendo la pertinencia del proceso.

Scriven (1991) sostiene que toda evaluación debe fundamentarse en juicios argumentados, basados tanto en resultados como en percepciones; por ello, este instrumento contribuyó a consolidar la validez interpretativa del estudio, integrando la voz de los participantes en la valoración global de la experiencia.

El portafolio de evidencias operó como un instrumento integrador que reunió los productos elaborados a lo largo de la experiencia: pseudocódigos, diagramas de flujo, versiones refactorizadas, informes y reflexiones finales. Este recurso permitió visualizar la evolución del aprendizaje de manera longitudinal, mostrando la relación entre la comprensión conceptual, la aplicación práctica y la reflexión técnica. De acuerdo con Yin (2014), la triangulación de fuentes y evidencias fortalece la confiabilidad del estudio de caso; así, el portafolio se constituyó en una base empírica sólida para validar la coherencia entre estrategias, resultados y competencias.

La pertinencia de estos instrumentos radicó en su alineación con los objetivos formativos y las competencias del perfil de egreso. Tal como subraya Jara (2018b), en la sistematización de experiencias la evaluación no debe entenderse como cierre, sino como una fase de reflexión y aprendizaje que otorga sentido al proceso vivido.

En este caso, los instrumentos aplicados permitieron capturar tanto el desarrollo técnico como la transformación pedagógica y ética de los estudiantes, generando una evaluación integral y contextualizada. Además, la combinación de herramientas cuantitativas y cualitativas garantizó la validez metodológica y la riqueza interpretativa del análisis (R. Stake, 1995); (Biggs, 2014).

En síntesis, los instrumentos de evaluación aplicados no solo cumplieron una función valorativa, sino también formativa y reconstructiva. A través de ellos, fue posible demostrar la coherencia entre la planificación estratégica y los resultados obtenidos, fortaleciendo la credibilidad del proceso y la transferibilidad de la experiencia.

Tal como afirman Fullan y Quinn (2017), una innovación educativa alcanza sostenibilidad cuando incorpora la evaluación como parte orgánica del aprendizaje institucional. En este sentido, la aplicación sistemática y reflexiva de los instrumentos otorgó validez al ecosistema pedagógico desarrollado, convirtiéndolo en un modelo replicable para futuras prácticas de enseñanza en el ámbito de la ingeniería de software.

3.5.3. Recomendaciones para profundizar

- Asegurar que cada instrumento se describa con claridad, explicando qué evalúa, cómo se aplicó y qué tipo de evidencia generó.
- Evitar la redundancia entre la descripción del instrumento y la presentación de las evidencias: priorizar la función y el aporte específico de cada uno.
- Destacar cómo los instrumentos se complementan entre sí para garantizar una evaluación holística y triangulada.
- Mantener el equilibrio entre la descripción práctica (cómo se usaron) y el fundamento teórico (por qué son pertinentes).
- Usar este puente como transición hacia el análisis de resultados y las interpretaciones globales del proceso evaluativo.

3.5.4. Clase 2: Indicadores de evaluación y criterios de validez

Reflexionar sobre los indicadores de evaluación es un paso fundamental dentro del proceso de sistematización, pues permite vincular la evidencia empírica del aprendizaje con los propósitos formativos de la experiencia. Como señalan Scriven (1991) y Casanova (1999), los indicadores constituyen los referentes observables que traducen las competencias y resultados esperados en manifestaciones concretas del desempeño. En este sentido, su función no se limita a medir, sino a dar sentido y dirección a la evaluación formativa, garantizando que las decisiones pedagógicas se sustenten en juicios fundamentados y coherentes con el perfil de egreso (Biggs, 2014); (R. Stake, 1995). En la experiencia desarrollada en Algoritmos y Lógica de Programación, los indicadores se organizaron en tres dimensiones complementarias: cognitivo–conceptual, técnico–procedimental y reflexivo–colaborativa, articulando el saber, el saber hacer y el saber reflexionar del futuro Ingeniero en Software.

Indicadores aplicados

1. Dimensión cognitivo–conceptual:

Este indicador midió la comprensión de la lógica algorítmica, la capacidad de análisis y la abstracción de problemas reales. Evaluó la identificación de variables, estructuras condicionales y secuencias coherentes dentro de la resolución de ejercicios. Se aplicó mediante rúbricas y análisis de pseudocódigos elaborados por los estudiantes ediagramas de flujo correctamente estructurados y algoritmos que demostraron una comprensión progresiva del razonamiento lógico y la traducción de la teoría a la práctica (Grover Pea, 2020).

2. Dimensión técnico–procedimental:

Evaluó la aplicación de buenas prácticas de programación y la calidad del código en términos de legibilidad, modularidad, documentación y eficiencia. Este indicador se aplicó principalmente durante las fases de refactorización y revisión entre pares, utilizando rúbricas específicas basadas en estándares profesionales (R. Zanatta & da Silva, 2022). Las evidencias incluyeron versiones comparativas de código antes y después de la optimización, mostrando mejoras en la estructura y en la aplicación de principios de calidad como DRY y KISS. Este indicador permitió observar el paso del aprendizaje inicial hacia la competencia técnica autónoma.

3. Dimensión reflexivo–colaborativa:

Midió la capacidad de autorregulación, la participación en procesos de revisión entre pares y la argumentación ética y profesional de las decisiones técnicas. Se aplicó mediante bitácoras reflexivas y registros de revisión (Peer Review Logs), donde los estudiantes documentaron sus observaciones y aprendizajes colaborativos (Johnson & Johnson, 2019). Las evidencias obtenidas incluyeron fichas de retroalimentación mutua, comentarios técnicos y reflexiones escritas sobre el proceso de mejora. Este indicador permitió valorar la dimensión social y ética del aprendizaje, fortaleciendo la responsabilidad y la comunicación profesional.

4. Dimensión metacognitiva y de transferencia:

Evaluó la capacidad de los estudiantes para explicar sus razonamientos, justificar decisiones de programación y transferir lo aprendido a nuevos contextos o problemas. Se aplicó a través de entrevistas y portafolios finales, donde se analizaron

los informes de laboratorio y las reflexiones sobre la evolución personal (Carlino, 2005); (Hyland, 2009). Las evidencias mostraron cómo los estudiantes fueron capaces de conectar su comprensión conceptual con la aplicación técnica, proyectando aprendizajes hacia escenarios de ingeniería reales.

En conjunto, estos indicadores permitieron observar una progresión integral del aprendizaje, desde la comprensión conceptual hasta la aplicación profesional, validando la coherencia entre enseñanza, evaluación y resultados.

Criterios de validez adoptados

La solidez del proceso evaluativo se sustentó en la aplicación de criterios rigurosos de validez y credibilidad. En primer lugar, se garantizó la validez de contenido, asegurando que los instrumentos evaluaran aspectos esenciales del pensamiento lógico, la calidad técnica del código y la reflexión ética (Casanova, 1999). La validez de constructo se garantizó mediante la alineación entre competencias, indicadores y resultados de aprendizaje, siguiendo los principios de alineación constructiva propuestos por Biggs (2014). Asimismo, se aplicó validez interna a través de la triangulación de datos —rúbricas, bitácoras, revisiones y portafolios—, lo que permitió contrastar distintas fuentes y perspectivas del mismo fenómeno.

La validez externa se alcanzó al comparar los hallazgos con estándares internacionales de formación en ingeniería de software (M. Zanatta & da Silva, 2022) y con modelos de enseñanza reflexiva (Yin, 2014). Finalmente, la credibilidad y confiabilidad del proceso se reforzaron mediante la aplicación transparente y sistemática de los instrumentos, garantizando que los resultados representaran de manera fiel el aprendizaje alcanzado y su pertinencia curricular (R. Stake, 1995).

Síntesis final

Los indicadores y criterios de validez aplicados en esta experiencia permitieron transformar la evaluación en un proceso reflexivo, formativo y éticamente responsable. Al integrar medición técnica, análisis conceptual y reflexión metacognitiva, la evaluación trascendió la simple calificación para convertirse en un espacio de aprendizaje y mejora continua (Scriven, 1991); (Fullan & Quinn, 2017). Gracias a estos indicadores, fue posible evidenciar la coherencia entre la práctica docente, las competencias del perfil de egreso y

los resultados esperados, otorgando al proceso de sistematización validez, credibilidad y transferibilidad hacia otros contextos educativos de ingeniería.

Recomendaciones para profundizar

- Mantener una correspondencia explícita entre indicadores y competencias para reforzar la coherencia curricular.
- Describir con claridad cómo los criterios de validez se aplican en la práctica, no solo como conceptos teóricos.
- Evitar generalizaciones: cada indicador debe ser operativo, observable y medible.
- Integrar la voz de los estudiantes como parte de la validez interpretativa, fortaleciendo la evaluación participativa.
- Utilizar los indicadores no solo para valorar resultados, sino para retroalimentar y ajustar estrategias en futuros ciclos formativos.

3.5.5. Clase 3: Análisis preliminar de evidencias

Durante la experiencia educativa desarrollada en la asignatura Algoritmos y Lógica de Programación, se recopilaron múltiples tipos de evidencias que permitieron analizar el proceso formativo de manera integral, abarcando las dimensiones conceptual, técnica y reflexiva. En el plano técnico, las rúbricas de evaluación del código generaron registros objetivos sobre los productos elaborados por los estudiantes —pseudocódigos, diagramas de flujo y versiones refactorizadas—, en los que se evidenció una progresiva mejora en la legibilidad, coherencia y documentación. Paralelamente, las bitácoras del estudiante y los informes de laboratorio ofrecieron evidencias reflexivas que dieron cuenta del proceso interno de aprendizaje, las decisiones tomadas y la capacidad de autorregulación.

Las revisiones entre pares aportaron evidencias colaborativas que mostraron cómo la crítica constructiva y la comunicación técnica fortalecieron el trabajo ético y cooperativo. Finalmente, las entrevistas y cuestionarios permitieron obtener evidencias perceptivas sobre la pertinencia y utilidad de las estrategias pedagógicas. En conjunto, estas fuentes configuraron un corpus diverso y complementario que permitió comprender el aprendizaje como un fenómeno multidimensional (Casanova, 1999); (R. Stake, 1995).

El método de organización y análisis de las evidencias se desarrolló bajo un enfoque mixto y triangulado, combinando procedimientos cuantitativos y cualitativos (Yin, 2014). En una primera etapa, se clasificaron las evidencias conforme a las tres dimensiones evaluativas: cognitivo–conceptual, técnico–procedimental y reflexivo–colaborativa. Esta categorización permitió establecer relaciones entre los indicadores de desempeño y las competencias esperadas del perfil de egreso.

Posteriormente, se aplicó un análisis descriptivo de los resultados rubricados y de los portafolios estudiantiles para identificar tendencias en la calidad del código y el dominio de las estructuras algorítmicas. Paralelamente, se desarrolló un análisis de contenido sobre las bitácoras, revisiones entre pares y entrevistas, utilizando procedimientos de codificación temática para detectar patrones de razonamiento, actitudes colaborativas y niveles de autorreflexión (Miles, Huberman Saldaña, 2018). La triangulación de estos datos, tal como proponen R. Stake (1995) y Flick (2014), permitió contrastar distintas perspectivas del mismo proceso, reforzando la credibilidad y consistencia de los hallazgos obtenidos.

Los hallazgos preliminares revelaron avances significativos en las tres dimensiones de evaluación. En la dimensión cognitivo–conceptual, los estudiantes demostraron una mayor capacidad para analizar problemas y abstraerlos en estructuras algorítmicas, identificando variables relevantes y secuencias lógicas adecuadas. En la dimensión técnico–procedimental, se constató una mejora en la calidad del código: los participantes aplicaron principios de programación limpia, refactorización coherente y documentación estandarizada, evidenciando dominio progresivo de los entornos Flowgorithm y PSeInt (R. Zanatta & da Silva, 2022).

Finalmente, en la dimensión reflexivo–colaborativa, se destacó el fortalecimiento del trabajo en equipo, la retroalimentación entre pares y la argumentación ética de las decisiones técnicas. Estas transformaciones, observadas en el conjunto de evidencias, confirman que el aprendizaje no se limitó a la adquisición de habilidades técnicas, sino que se expandió hacia una comprensión crítica y profesional de la práctica ingenieril (Johnson & Johnson, 2019); (Biggs, 2014).

Los ejemplos ilustrativos proporcionan una mirada concreta sobre cómo las estrategias implementadas se tradujeron en logros observables. En los talleres de análisis algorítmico, por ejemplo, los estudiantes pasaron de resolver ejercicios aritméticos básicos a modelar problemas contextualizados, sustituyendo operaciones numéricas simples por expresiones con variables significativas como “área = base * altura”. En las sesiones de laboratorio, utilizaron Flowgorithm y PSeInt para construir diagramas de flujo y detectar errores

lógicos, desarrollando autonomía en la depuración del código. Durante los procesos de refactorización pedagógica, mejoraron versiones iniciales de sus programas mediante indentación, comentarios explicativos y la aplicación de principios DRY y KISS, mostrando comprensión de la sostenibilidad del código.

Finalmente, en las revisiones entre pares, los estudiantes ofrecieron retroalimentación constructiva sustentada en rúbricas técnicas, fortaleciendo su pensamiento crítico y su sentido de responsabilidad compartida. Estos ejemplos, más allá de ilustrar logros individuales, reflejan la consolidación de un ecosistema de aprendizaje participativo y reflexivo (Carlino, 2005); (Jara, 2018a).

En síntesis, el análisis preliminar de las evidencias permitió constatar la coherencia entre los instrumentos aplicados, los indicadores definidos y los resultados de aprendizaje alcanzados. La combinación de evidencias técnicas, reflexivas, colaborativas y perceptivas posibilitó una comprensión profunda de la experiencia educativa, asegurando la validez y transferibilidad de los hallazgos (Scriven, 1991); (Yin, 2014). Estos resultados iniciales no solo validan la eficacia de las estrategias pedagógicas implementadas, sino que también preparan el terreno para una reflexión más amplia sobre los criterios de validez, sesgos y sostenibilidad del proceso evaluativo. En esta línea, el análisis de evidencias se consolida como una etapa esencial en la sistematización, al permitir que la innovación educativa se traduzca en conocimiento compartido, verificable y replicable.

3.5.6. Clase 4: Reflexión sobre validez, sesgos y factibilidad

Estrategias aplicadas para asegurar validez

La validez del proceso evaluativo se aseguró mediante la implementación de estrategias metodológicas orientadas a garantizar la coherencia, credibilidad y consistencia de los resultados. En primer lugar, se aplicó la **triangulación de fuentes y métodos** (R. Stake, 1995); (Yin, 2014), combinando evidencias provenientes de rúbricas, bitácoras reflexivas, registros de revisión entre pares, entrevistas y portafolios. Esta convergencia de datos permitió contrastar distintas perspectivas del aprendizaje y fortalecer la confiabilidad de los hallazgos.

Además, se cuidó la **validez de contenido** al alinear los instrumentos con los indicadores de desempeño y las competencias del perfil de egreso, garantizando que la evaluación abordara los componentes esenciales del pensamiento lógico, la calidad del código y la reflexión profesional ((Casanova, 1999); (Biggs, 2014).

Del mismo modo, se consideró la **validez de constructo**, asegurando que las categorías de análisis reflejaran fielmente las dimensiones cognitivo–conceptual, técnico–procedimental y reflexivo–colaborativa. La **revisión entre pares docentes** y la retroalimentación formativa fueron mecanismos complementarios que validaron las interpretaciones y aseguraron una lectura plural de los resultados.

Finalmente, se fortaleció la **credibilidad y transferibilidad** mediante la documentación sistemática de las decisiones metodológicas y la trazabilidad de las evidencias, de acuerdo con los criterios de calidad propuestos por Yin (2014) y Maxwell (2013). En conjunto, estas estrategias consolidaron una evaluación válida, transparente y coherente con los objetivos formativos del programa.

Sesgos identificados y cómo se mitigaron

A lo largo del proceso se identificaron varios tipos de sesgos potenciales que podían afectar la objetividad de los resultados. El **sesgo del observador** fue uno de los primeros reconocidos, mitigado mediante la aplicación de **rúbricas estandarizadas y la revisión cruzada entre docentes**, lo que permitió mantener criterios uniformes y evitar valoraciones subjetivas.

Asimismo, el **sesgo de autopercepción** en las bitácoras y cuestionarios de los estudiantes se controló mediante **guías de autorreflexión estructuradas** y la triangulación con evidencias técnicas y colaborativas (R. Stake, 1995); (Yin, 2014).

Otro sesgo relevante fue el **de confirmación**, que surgió durante el análisis de resultados al contrastar datos empíricos con hipótesis previas. Para mitigarlo, se promovió la **contrastación constante** entre los hallazgos y los objetivos curriculares, priorizando la interpretación inductiva frente a la deducción preconcebida (Maxwell, 2013).

También se detectó el **sesgo de participación**, asociado a diferencias en la implicación estudiantil. Este se redujo ofreciendo **múltiples canales y tiempos de recolección de información**, garantizando la inclusión y la diversidad de perspectivas. En conjunto, estas acciones metodológicas reforzaron la **validez interna** y la **credibilidad interpretativa** del proceso, demostrando que el reconocimiento y manejo consciente de los sesgos es una condición indispensable para una sistematización rigurosa y honesta.

Dificultades de factibilidad

Durante la evaluación se enfrentaron diversas dificultades que comprometieron la factibilidad del proceso, principalmente relacionadas con los recursos tecnológicos, el tiempo disponible y la participación activa del estudiantado. En primer lugar, las **limitaciones de conectividad** y el acceso desigual a equipos informáticos afectaron la implementación simultánea de ciertos instrumentos digitales. Para contrarrestar esta situación, se diseñaron **materiales impresos y actividades offline**, asegurando la continuidad pedagógica y la equidad en la participación (Patton, 2002).

Otra dificultad importante fue la **restricción temporal** derivada de la carga académica de los estudiantes y de la complejidad de las actividades prácticas. Este obstáculo se superó mediante la **reorganización del cronograma**, priorizando la profundidad analítica sobre la cantidad de tareas. Finalmente, la **resistencia inicial al trabajo reflexivo y colaborativo** limitó en un inicio la riqueza de las evidencias metacognitivas.

Para abordarlo, se implementaron **espacios de acompañamiento docente** y se promovió una cultura de confianza y corresponsabilidad ética. Estas soluciones demostraron que la factibilidad no solo depende de los recursos disponibles, sino también de la **flexibilidad pedagógica** y la **capacidad adaptativa** del equipo docente (Fullan & Quinn, 2017).

Aprendizajes de esta reflexión

El proceso reflexivo sobre validez, sesgos y factibilidad dejó aprendizajes valiosos que fortalecieron tanto la práctica pedagógica como la comprensión de la evaluación como proceso formativo. En primer lugar, se reafirmó que **la validez no se reduce a la precisión técnica de los instrumentos**, sino que se construye desde la coherencia entre los objetivos, las estrategias y los resultados obtenidos. Igualmente, se consolidó la importancia de la **triangulación metodológica** como principio ético y científico que otorga equilibrio, profundidad y credibilidad a los hallazgos (Yin, 2014); (R. Stake, 1995).

Se aprendió también que los **sesgos son inevitables**, pero su reconocimiento y tratamiento transparente fortalecen la integridad del proceso investigativo (Maxwell, 2013). Del mismo modo, se evidenció que las **limitaciones de factibilidad** pueden convertirse en oportunidades de innovación cuando se gestionan con creatividad y sentido pedagógico. Finalmente, esta reflexión permitió comprender que **evaluar no es solo medir resultados, sino comprender procesos**, reconocer avances y transformar la práctica docente en

conocimiento compartido. En conjunto, estos aprendizajes consolidaron una visión más ética, reflexiva y sostenible de la evaluación educativa, coherente con los principios de la innovación y la mejora continua (Patton, 2002); (Jara, 2018a).

3.6. Transición hacia la reflexión final

La evaluación integral de la experiencia pedagógica en *Algoritmos y Lógica de Programación* permitió confirmar logros significativos y reconocer con madurez las limitaciones inherentes a todo proceso innovador. Los resultados mostraron avances sustanciales en la comprensión del pensamiento computacional, la calidad técnica del código y el fortalecimiento del trabajo colaborativo, evidenciando un desarrollo integral del perfil del Ingeniero en Software.

Sin embargo, también se identificaron restricciones tecnológicas, temporales y comunicativas que condicionaron parcialmente la profundidad de ciertos análisis y pusieron de relieve la necesidad de fortalecer la competencia reflexiva y la gestión del tiempo en futuras implementaciones. Estos hallazgos, más que debilidades, se constituyen en oportunidades de mejora que enriquecen la comprensión del proceso evaluado.

Con base en estos resultados, el proceso de sistematización se abre ahora hacia una etapa de reflexión crítica y proyección pedagógica. Esta nueva sección no busca solo cerrar el ciclo evaluativo, sino interpretar los aprendizajes obtenidos y explorar su potencial de transferibilidad hacia otros contextos formativos. La mirada reflexiva permitirá comprender cómo los logros alcanzados y las limitaciones enfrentadas se transforman en conocimiento útil para la mejora continua, la innovación sostenida y la construcción colectiva de una práctica docente más consciente y significativa.

En este sentido, la transición hacia la reflexión final marca el paso de la verificación empírica a la comprensión profunda, invitando a analizar la experiencia no como un punto de llegada, sino como un punto de partida para futuras transformaciones educativas.

Recomendaciones para profundizar

- Evita redundar en la descripción de logros o limitaciones ya desarrollados en el módulo anterior; resume solo lo necesario para mantener la continuidad narrativa.
- Mantén un **tono de apertura y expectativa**, preparando al lector para un análisis más interpretativo y reflexivo en el siguiente módulo.

- Usa conectores de transición (“a partir de estos hallazgos”, “con base en los resultados evaluados”, “en esta nueva etapa...”) que evidencien el cambio de enfoque de la *evaluación* hacia la *reflexión crítica*.
- Destaca la **proyección futura** de la experiencia: cómo los aprendizajes servirán para transformar otras prácticas o contextos educativos.

3.6.1. Clase 1: Reflexión crítica sobre la experiencia

La experiencia pedagógica desarrollada en la asignatura Algoritmos y Lógica de Programación aportó contribuciones sustantivas tanto al aprendizaje de los estudiantes como a la reflexión docente e institucional. En el plano formativo, consolidó un modelo de enseñanza que integró el pensamiento lógico, la práctica técnica y la reflexión ética como dimensiones complementarias del aprendizaje. Este enfoque, sustentado en la articulación de estrategias núcleo, de soporte y de contingencia, permitió transformar el aula en un ecosistema activo, donde la experimentación, la colaboración y la autorregulación fueron parte esencial del proceso (Grover & Pea, 2020); (Jara, 2018a).

Además, la aplicación de una evaluación continua y triangulada, basada en rúbricas, portafolios y revisiones entre pares, demostró que la evaluación puede convertirse en una herramienta de comprensión profunda más que en un mecanismo de control (Casanova, 1999); (R. Stake, 1995). En el plano institucional, la experiencia aportó un modelo de innovación educativa coherente con el perfil de egreso del Ingeniero en Software, ofreciendo una ruta replicable para integrar currículo, evaluación y formación ética profesional.

Sin embargo, el desarrollo de la experiencia también enfrentó tensiones y resistencias que enriquecieron el proceso reflexivo. Una de las más notorias fue la resistencia inicial de los estudiantes al trabajo colaborativo y a la revisión entre pares, debido al temor a la crítica o a la exposición de sus errores. Esta dificultad se fue transformando mediante la creación de un clima de confianza, orientado al respeto y la mejora colectiva (Johnson & Johnson, 2019).

Asimismo, se identificaron desafíos vinculados a la autorreflexión y la escritura técnica, especialmente en las primeras etapas, donde la argumentación de decisiones algorítmicas resultaba limitada. A nivel operativo, las restricciones tecnológicas y de tiempo condicionaron el uso simultáneo de herramientas digitales y la revisión exhaustiva de productos, obligando a adoptar medidas de flexibilización pedagógica (Flick, 2014). Finalmente, la diversidad de niveles de conocimiento previo generó brechas de ritmo y

comprensión, lo que demandó estrategias diferenciadas y acompañamiento docente permanente (Tomlinson, 2017). Estas tensiones, lejos de debilitar el proceso, se convirtieron en catalizadores de innovación y en aprendizajes compartidos sobre la gestión de la diversidad y la resiliencia pedagógica.

Los aprendizajes derivados del proceso fueron múltiples y se expresaron en los niveles personal, colectivo e institucional. En el plano personal, los docentes fortalecieron su capacidad para reflexionar críticamente sobre su práctica, reconociendo la necesidad de equilibrar lo técnico con lo humano en la enseñanza de la programación. Esta reflexión permitió comprender que la innovación no se trata únicamente de incorporar herramientas, sino de **transformar las relaciones pedagógicas** y de promover la autonomía del estudiante como protagonista de su aprendizaje (Carlino, 2005).

En el nivel colectivo, los estudiantes desarrollaron habilidades comunicativas, colaborativas y éticas, aprendiendo a valorar la revisión entre pares como un ejercicio de confianza, diálogo y mejora continua. A nivel institucional, la experiencia aportó evidencia concreta de que es posible articular metodologías activas, evaluación formativa y acompañamiento reflexivo en carreras de ingeniería, consolidando un modelo educativo alineado con el perfil profesional y las demandas tecnológicas contemporáneas (Fullan & Quinn, 2017); (Biggs, 2014)). En conjunto, estos aprendizajes fortalecieron una cultura educativa orientada a la calidad, la sostenibilidad y la mejora continua.

Finalmente, la sistematización permitió comprender que **reflexionar sobre la práctica educativa es un acto de transformación**. Más que un registro de acciones, este proceso se constituyó en una oportunidad para aprender del propio quehacer docente y resignificar la enseñanza de la programación como una experiencia humana, ética y social. Siguiendo a Jara (2018a), la sistematización no solo permite reconstruir lo vivido, sino otorgarle sentido, visibilizando la coherencia entre teoría y práctica, entre lo planificado y lo aprendido.

A través de esta mirada crítica, se logró identificar cómo cada decisión metodológica impactó en la formación del estudiante, cómo las tensiones se transformaron en aprendizajes y cómo la innovación puede consolidarse como una práctica sostenida. En este sentido, la reflexión final reafirma que sistematizar no es concluir un proceso, sino abrir un ciclo nuevo de aprendizaje colectivo, orientado a la transferencia, la mejora y la consolidación de comunidades educativas que aprenden de su propia experiencia.

3.6.2. Recomendaciones para profundizar

- Evitar una visión idealizada de la experiencia: reconocer logros, tensiones y límites como parte de un mismo proceso formativo.
- Balancear la mirada personal (del docente y del estudiante) con la institucional, destacando cómo ambas se complementan en la innovación educativa.
- Incorporar ejemplos concretos que ilustren la transformación pedagógica, mostrando evidencias del cambio en la práctica.
- Usar esta reflexión como **bisagra hacia la transferencia**, preparando el terreno para el análisis de cómo los aprendizajes y hallazgos pueden replicarse o adaptarse en otros contextos educativos.

Bibliografía

- Ahmad, N., & Hashim, N. (2020). Peer code review as a pedagogical strategy for software quality learning. *Journal of Software Engineering Education*.
- Barnett, R., & Jackson, N. (2019). *Learning for an Unknown Future: Higher Education and the Human Condition*. Routledge.
- Biggs, J. (2014). Constructive alignment in university teaching. *Higher Education*, 67(3), 231-243. <https://doi.org/10.1007/s10734-013-9700-7>
- Biggs, J., & Tang, C. (2011). *Teaching for Quality Learning at University* (4.^a ed.). McGraw-Hill Education.
- Carlino, P. (2005). *Escribir, leer y aprender en la universidad: Una introducción a la alfabetización académica*. Fondo de Cultura Económica.
- Casanova, M. A. (1999). *La evaluación educativa: Escuela, aula y alumno*. La Muralla.
- Curzon, P., & McOwan, P. W. (2019). *The Power of Computational Thinking: Games, Magic and Puzzles to Help You Become a Computational Thinker*. Cambridge University Press.
- Facione, P. A. (2020). *Critical Thinking: What It Is and Why It Counts*. Insight Assessment.
- Flick, U. (2014). *Introducción a la investigación cualitativa* (5.^a ed.). Morata.
- Fullan, M., & Quinn, J. (2017). *Coherence: The Right Drivers in Action for Schools, Districts, and Systems*. Corwin Press.
- Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38-43. <https://doi.org/10.3102/0013189x12463051>
- Grover, S., & Pea, R. (2020). Computational thinking: A competency whose time has come. *Computer Science Education*, 30(1), 100-120.
- Guzdial, M., & Morrison, B. (2020). Computing education: From designing programs to designing learning experiences. *Communications of the ACM*, 63(2), 30-33.
- Hargie, O. (2021). *Skilled Interpersonal Communication: Research, Theory and Practice* (7.^a ed.). Routledge.
- Hermans, F. (2021). *The Programmer's Brain: What Every Programmer Needs to Know About Cognition*. Manning Publications.
- Hyland, K. (2009). *Academic Discourse: English in a Global Context*. Continuum.

- Jara, O. (2018a). *La sistematización de experiencias: Aprendizajes y desafíos para la educación popular*. Alforja.
- Jara, O. (2018b). *La sistematización de experiencias: Práctica y teoría para otros mundos posibles*. Alforja.
- Johnson, D. W., & Johnson, R. T. (2019). *Cooperation and Competition: Theory and Research*. Interaction Book Company.
- Johnson, D. W., Johnson, R. T., & Smith, K. A. (2020). Cooperative learning: Improving university instruction by basing practice on validated theory. *Journal on Excellence in College Teaching*, 31(2), 34-68.
- Kafai, Y. B., Proctor, C., & Lui, D. (2019). From theory bias to theory dialogue: Embracing cognitive, situated, and critical framings of computational thinking in K–12. *ACM Inroads*, 10(4), 64-71. <https://doi.org/10.1145/3363181>
- Kolb, D. A. (2015). *Experiential Learning: Experience as the Source of Learning and Development* (2.^a ed.). Pearson Education.
- Lai, E. R. (2022). *Critical Thinking in the 21st Century: A Framework for Learning and Innovation*. Pearson Education.
- Lodi, M., & Martini, S. (2021). Computational thinking, between Papert and Wing. *Philosophy & Technology*, 34(4), 1043-1068.
- Maxwell, J. A. (2013). *Qualitative Research Design: An Interactive Approach*. SAGE Publications.
- Morales, J., Paredes, R., & Cifuentes, A. (2021). Formación ética en ingeniería: desafíos para la educación superior latinoamericana. *Revista Iberoamericana de Educación Superior*, 12(33), 45-62.
- Patton, M. Q. (2002). *Qualitative Research and Evaluation Methods*. SAGE Publications.
- Petre, M., & van der Hoek, A. (2021). *Software Design in Practice: Collaboration and Learning in Context*. Cambridge University Press.
- Scriven, M. (1991). *Evaluation Thesaurus* (4.^a ed.). SAGE Publications.
- Sentance, S., Waite, J., & Kallia, M. (2019a). Teaching computer programming with PRIMM: A sociocultural perspective. *Computer Science Education*, 29(2–3), 136-176.
- Sentance, S., Waite, J., & Kallia, M. (2019b). Teaching Computer Programming: Principles into Practice. *Computing Education Practice Conference*, 1-8. <https://doi.org/10.1145/3294016.3294018>
- Stake, R. (1995). *The Art of Case Study Research*. SAGE Publications.
- Stake, R. E. (2020). *The Art of Case Study Research (Revised)*. SAGE Publications.

- Sweller, J., Ayres, P., & Kalyuga, S. (2019). *Cognitive Load Theory*. Springer.
- Sweller, J., Van Merriënboer, J. J., & Paas, F. (2019). Cognitive architecture and instructional design: 20 years later. *Educational Psychology Review*, 31(2), 261-292.
- Tomlinson, C. A. (2017). *How to Differentiate Instruction in Academically Diverse Classrooms* (3.^a ed.). ASCD.
- Universidad Estatal de Milagro (UNEMI). (2021). Modelo Educativo de la Universidad Estatal de Milagro.
- Universidad Estatal de Milagro (UNEMI). (2023). Marco curricular institucional de la Universidad Estatal de Milagro.
- Yin, R. K. (2014). *Case Study Research: Design and Methods* (5.^a ed.). SAGE Publications.
- Zabalza, M. A. (2019). *Competencias docentes del profesorado universitario: Calidad y desarrollo profesional* (3.^a ed.). Narcea.
- Zanatta, M., & da Silva, M. (2022). Software engineering education and code quality practices. *IEEE Transactions on Education*, 65(4), 532-540.
- Zanatta, R., & da Silva, P. (2022). Clean code principles for novice programmers: A pedagogical approach. *Journal of Computing Education*.
- Zimmerman, B. J. (2020). Motivating self-regulated learners: The future of educational psychology. *Journal of Educational Psychology*, 112(2), 331-343.

4

Aprender gestionando: experiencias innovadoras en proyectos de software universitarios

Denis Darío Mendoza Cabrera⁴

El Capítulo 4 resume aprendizajes y proyecciones del programa RI-SEI, destacando el paso de innovaciones pedagógicas individuales a una política institucional. Plantea construir una cultura universitaria de investigación educativa que articule docencia, formación docente y gestión académica. Señala que la sistematización convierte prácticas innovadoras en conocimiento validado, compatible y transferible, fortaleciendo la profesionalización docente y la calidad educativa. Analiza mecanismos de acompañamiento, evaluación y difusión para asegurar sostenibilidad. Propone un modelo basado en articulación entre unidades académicas, trabajo en red, formación permanente e investigación aplicada a la enseñanza. Cierra con desafíos para consolidar una universidad que aprende de su práctica.

⁴Universidad Estatal de Milagro, dmendozac2@unemi.edu.ec.

Índice

4.1. Aprender gestionando: experiencias innovadoras en proyectos de software universitarios	175
4.1.1. Apertura contextual	175
4.1.2. Problematización	175
4.1.3. Las consecuencias de este déficit se manifiestan en tres niveles .	176
4.1.4. Propósito de la sistematización	176
4.1.5. Criterios de valor	177
4.1.6. Delimitación del objeto de estudio	178
4.2. Fundamentación teórico-metodológica de la experiencia docente . . 179	
4.2.1. Bisagra textual	179
4.2.2. Identificación de conceptos estructurantes	180
4.2.3. Aprendizaje activo	180
4.2.4. Pensamiento crítico	181
4.2.5. Autonomía en el aprendizaje	181
4.2.6. Aprendizaje colaborativo	182
4.2.7. Mentoría pedagógica	182
4.2.8. Relación entre los conceptos	183
4.2.9. Formulación de dimensiones	183
4.2.10. Dimensión pedagógica	184
4.2.11. Dimensión cognitiva-formativa	185
4.2.12. Dimensión socioafectiva-colaborativa	185
4.2.13. Construcción de indicadores	186
4.2.14. Indicadores de la dimensión pedagógica	187
4.2.15. Indicadores de la dimensión cognitiva-formativa	188
4.2.16. Indicadores de la dimensión socioafectiva-colaborativa . . .	188
4.2.17. Fuentes	190
4.2.18. Métodos de verificación	191

4.2.19. Análisis comparativo de productos	191
4.2.20. Análisis de contenido	192
4.2.21. Triangulación de evidencias	192
4.2.22. Justificación teórica del conjunto	193
4.3. Integración curricular y desarrollo de competencias en la formación del ingeniero de software	195
4.3.1. Transición al vínculo curricular	195
4.3.2. Identificación de competencias del perfil	196
4.3.3. Resultados de aprendizaje vinculados	198
4.3.4. Actividades y evidencias	200
4.3.5. Reflexión sobre la alineación curricular	202
4.4. Diseño e implementación del ecosistema estratégico de aprendizaje en ingeniería de software	204
4.4.1. Transición hacia la operacionalización estratégica	204
4.4.2. Estrategias núcleo en acción	205
4.4.3. Estrategias de soporte aplicadas	207
4.4.4. Estrategias de contingencia desplegadas	209
4.4.5. Arquitectura del ecosistema estratégico	211
4.5. Evaluación integral de la experiencia: instrumentos, evidencias y validez del proceso formativo	213
4.5.1. Transición hacia la evaluación	213
4.5.2. Instrumentos de evaluación aplicados	214
4.5.3. Indicadores de evaluación y criterios de validez	216
4.5.4. Análisis preliminar de evidencias	218
4.5.5. Reflexión sobre validez, sesgos y factibilidad	220
4.6. Reflexión final y proyección institucional de la experiencia docente .	222
4.6.1. Transición hacia la reflexión final	222
4.6.2. Reflexión crítica sobre la experiencia	223
4.6.3. Tensiones y resistencias encontradas	224
4.6.4. Aprendizajes personales, colectivos e institucionales	225

4.1. Aprender gestionando: experiencias innovadoras en proyectos de software universitarios

4.1.1. Apertura contextual

En continuidad con los capítulos previos, este texto se enfoca en una dimensión complementaria y necesaria para consolidar la formación integral del ingeniero de software: el desarrollo del **pensamiento crítico** y de las **habilidades blandas** que sustentan la autonomía, la colaboración y la resolución creativa de problemas. Mientras los capítulos anteriores abordaron la comprensión conceptual, el análisis, el diseño y la calidad del código, este capítulo se adentra en los aspectos formativos que permiten a los estudiantes integrar estos saberes en un desempeño reflexivo y profesionalmente competente.

La experiencia se desarrolla en la **Universidad Estatal de Milagro (UNEMI)**, en la carrera de Ingeniería de Software, dentro de la asignatura Introducción a la Ingeniería de Software. El escenario corresponde al primer semestre, con aproximadamente 40 estudiantes provenientes de distintos cantones de la provincia del Guayas. Este grupo heterogéneo, con trayectorias educativas diversas y condiciones socioeconómicas variables, representa un espacio de aprendizaje caracterizado por la pluralidad de experiencias, expectativas y niveles de familiaridad tecnológica.

En las primeras semanas de clase se evidenció una marcada dependencia del docente: los estudiantes esperaban instrucciones precisas para cada tarea y mostraban escasa iniciativa para la autoexploración o la resolución autónoma. Esta situación motivó la implementación de una propuesta pedagógica orientada a fomentar el pensamiento crítico, la reflexión sobre el propio aprendizaje y la cooperación entre pares como pilares de la formación universitaria. La estrategia se centró en transformar la **heterogeneidad del grupo** en una oportunidad para fortalecer la autonomía y la colaboración, promoviendo una cultura de aprendizaje activo.

4.1.2. Problematización

El principal problema identificado en la asignatura fue la **escasa puesta en práctica de habilidades blandas**, especialmente aquellas vinculadas con el pensamiento crítico, la resolución de problemas y la autonomía en el aprendizaje. A pesar de mostrar entusiasmo, los estudiantes tendían a depender excesivamente de las explicaciones del docente y evi-

denciaban dificultades para gestionar su proceso formativo. Este fenómeno se asocia con modelos educativos previos basados en la memorización, que obstaculizan el desarrollo de la reflexión y la toma de decisiones fundamentadas.

Diversos autores coinciden en que el pensamiento crítico constituye un componente esencial para la educación superior. Facione (2020) y Paul Elder (2014) destacan que sin estas competencias los estudiantes no logran analizar situaciones complejas ni construir juicios informados. En el ámbito de la ingeniería de software, esta carencia adquiere especial relevancia, ya que la disciplina exige una combinación de razonamiento lógico, análisis contextual y creatividad técnica.

4.1.3. Las consecuencias de este déficit se manifiestan en tres niveles

En primer lugar, los estudiantes reducen su experiencia universitaria al cumplimiento de tareas, sin desarrollar capacidades transferibles al ámbito profesional.

En segundo lugar, la falta de autonomía genera una dependencia excesiva del docente, que limita el aprendizaje autorregulado.

Finalmente, en contextos socioeconómicos complejos, las responsabilidades laborales o familiares restringen el tiempo de estudio y profundizan las desigualdades. Como sostiene Tünnermann (2018), cuando las condiciones externas impiden una dedicación plena al estudio, la universidad debe compensar mediante estrategias de acompañamiento que promuevan la autorreflexión y la autogestión del conocimiento.

Durante las primeras semanas de clases, se observaron evidencias concretas: en ejercicios grupales, algunos estudiantes evitaban debatir o proponer alternativas; en tareas de análisis de casos, se limitaban a repetir ejemplos previos sin cuestionar los supuestos. Estas actitudes reflejan la necesidad de un enfoque pedagógico que impulse la **participación activa, la reflexión crítica y la cooperación estructurada** como medios para desarrollar habilidades cognitivas y sociales fundamentales.

4.1.4. Propósito de la sistematización

El propósito de esta sistematización es mostrar cómo, mediante estrategias pedagógicas activas y procesos de mentoría, es posible fortalecer el pensamiento crítico y las habilidades blandas de los estudiantes de primer semestre de Ingeniería de Software, transformando la heterogeneidad del grupo en una fuente de aprendizaje colaborativo y significativo.

La propuesta se fundamenta en la idea de que las limitaciones pueden convertirse en oportunidades de aprendizaje. La docencia universitaria no debe limitarse a transmitir contenidos, sino a acompañar procesos de desarrollo integral que estimulen la reflexión, la argumentación y la autonomía. Tal como plantea Jara (2018), la sistematización cobra valor cuando permite reconstruir las prácticas docentes para convertirlas en conocimiento compartido y transferible a otros contextos.

El propósito busca, por tanto, evidenciar cómo un modelo de enseñanza basado en la **motivación, la colaboración y el aprendizaje activo** puede favorecer el desarrollo de competencias transversales —como la comunicación, la adaptabilidad y la toma de decisiones—, esenciales para el ejercicio profesional de la ingeniería. En este sentido, el docente asume el rol de **mentor académico y mediador cognitivo**, capaz de orientar, desafiar y acompañar al estudiante en la construcción de su propio aprendizaje.

4.1.5. Criterios de valor

La experiencia resulta valiosa por varias razones. En primer lugar, representa una **innovación pedagógica** frente a prácticas tradicionales centradas exclusivamente en la técnica. La incorporación de estrategias activas —debates guiados, estudios de caso y actividades de aprendizaje colaborativo— permitió integrar el desarrollo cognitivo con el socioemocional.

En segundo lugar, los estudiantes evidenciaron un cambio en su forma de aprender: pasaron de la dependencia a la participación, del silencio a la argumentación y del seguimiento de instrucciones a la toma de decisiones fundamentadas. Este cambio fue observable en sus producciones escritas, en su desempeño en presentaciones grupales y en la calidad de sus intervenciones durante las clases.

Otro criterio de valor es la **transferibilidad** del modelo. Las estrategias de mentoría, reflexión guiada y trabajo cooperativo pueden aplicarse en distintas asignaturas de la carrera e incluso en otros programas académicos. Además, la experiencia pone de relieve la necesidad de integrar explícitamente las habilidades blandas en los planes de estudio de ingeniería, alineándose con tendencias internacionales que reconocen el valor del pensamiento crítico como parte de las competencias profesionales del siglo XXI (Jaramillo Gómez et al., 2025); Karakuş et al., 2024).

Finalmente, el aporte de esta práctica radica en su capacidad para **construir comunidad de aprendizaje**. Al promover el trabajo colaborativo y la reflexión sobre la práctica,

los estudiantes desarrollaron actitudes de empatía, respeto y responsabilidad compartida. La sistematización de esta experiencia constituye, por tanto, un testimonio del poder transformador de la educación cuando combina exigencia académica con acompañamiento humano.

4.1.6. Delimitación del objeto de estudio

El objeto de estudio de esta sistematización se centra en el proceso de **enseñanza y acompañamiento pedagógico** desarrollado en la asignatura Introducción a la Ingeniería de Software con estudiantes de primer semestre de la UNEMI. El análisis se focaliza en cómo las **metodologías activas** y las estrategias de motivación implementadas por el docente contribuyeron al desarrollo del pensamiento crítico y las habilidades blandas, en un contexto caracterizado por la diversidad y las limitaciones socioeconómicas.

La delimitación temporal corresponde al semestre académico 2025-A y comprende un grupo de 40 estudiantes, en su mayoría hombres, provenientes de cantones cercanos a la ciudad de Milagro. Se excluyen del análisis las asignaturas de otros niveles y las variables institucionales de tipo macro, ya que el interés se centra en la **dinámica micro de aula** y su impacto en la formación de los estudiantes.

Se consideran como evidencias principales las actividades de aula, los debates grupales, las prácticas de aprendizaje colaborativo y las reflexiones individuales. Este recorte permite profundizar en el vínculo entre la interacción docente–estudiante y el desarrollo de competencias transversales. Tal como sostiene Flick (2014a), el valor de un estudio cualitativo reside en la coherencia entre el recorte temático y la profundidad del análisis.

En síntesis, el objeto de estudio de esta sistematización es **el proceso pedagógico orientado al fortalecimiento del pensamiento crítico y las habilidades blandas** como dimensiones indispensables en la formación del ingeniero de software. Este enfoque permite comprender que la excelencia técnica solo se consolida cuando se acompaña de madurez reflexiva, compromiso ético y capacidad colaborativa.

4.2. Fundamentación teórico-metodológica de la experiencia docente

4.2.1. Bisagra textual

Al concluir la introducción del primer capítulo, donde se describió el contexto de la experiencia, se identificaron los problemas formativos, se explicitó el propósito de la sistematización y se definieron los criterios de valor junto con el objeto de estudio, emerge la necesidad de avanzar hacia un nivel de fundamentación teórica que sustente los aprendizajes alcanzados.

El recorrido anterior permitió comprender el escenario educativo de la Universidad Estatal de Milagro, particularmente en la asignatura Introducción a la Ingeniería de Software, donde los estudiantes de primer semestre enfrentan el desafío de desarrollar pensamiento crítico, autonomía y habilidades blandas en un entorno de diversidad académica y sociocultural. Este proceso evidenció cómo la práctica docente, mediada por estrategias activas y de acompañamiento, transformó las limitaciones en oportunidades de aprendizaje significativo.

Sin embargo, los logros descritos no pueden entenderse únicamente desde la narrativa experiencial. Requieren una base conceptual que permita interpretar, con mayor rigurosidad, los procesos pedagógicos que los hicieron posibles. En este sentido, la fundamentación teórica se vuelve indispensable para dotar de validez académica a la experiencia, explicando los fenómenos observados mediante categorías y conceptos provenientes del campo educativo y de la ingeniería de software.

De este modo, la práctica narrada se convierte en objeto de análisis, trascendiendo la descripción empírica para situarse en un marco de comprensión sustentado en la teoría.

La necesidad de fundamentar conceptualmente la experiencia responde, además, al propósito de consolidar un puente entre la acción pedagógica y el conocimiento científico. Comprender cómo las metodologías activas, el aprendizaje basado en la práctica y la mentoría influyen en la formación inicial de ingenieros de software exige un abordaje sistemático que permita identificar sus dimensiones, indicadores y métodos de verificación. Esta transición marca el paso de la vivencia docente hacia el análisis académico, garantizando la coherencia del capítulo y proyectando el trabajo hacia la construcción del marco conceptual que dará soporte a la sistematización.

En el siguiente apartado, por tanto, se abordarán los conceptos estructurantes que sostienen la experiencia, organizados en torno a las categorías que explican el desarrollo del pensamiento crítico, la autonomía y la práctica reflexiva en el contexto universitario. Esta fundamentación permitirá comprender con mayor profundidad los procesos formativos que se vivieron en el aula y su relación con las transformaciones observadas en los estudiantes.

4.2.2. Identificación de conceptos estructurantes

La experiencia desarrollada en la asignatura Introducción a la Ingeniería de Software ha evidenciado la necesidad de fortalecer en los estudiantes de primer semestre no solo las competencias técnicas, sino también aquellas habilidades cognitivas y socioemocionales que sostienen el aprendizaje profundo. En este contexto, la práctica docente se articula sobre un conjunto de conceptos estructurantes que orientan su fundamentación teórica y operativa: aprendizaje activo, pensamiento crítico, autonomía en el aprendizaje, aprendizaje colaborativo y mentoría pedagógica.

Estos cinco conceptos dialogan entre sí y conforman el andamiaje que permite comprender la dinámica formativa y los resultados observados en los estudiantes. A continuación, se presenta el desarrollo conceptual de cada uno, sustentado en literatura académica reciente.

4.2.3. Aprendizaje activo

El aprendizaje activo constituye uno de los pilares de la experiencia, pues rompe con el paradigma tradicional centrado en la transmisión unidireccional de conocimientos y promueve la participación del estudiante en su propio proceso formativo. Según Prince (2021), el aprendizaje activo se basa en la implicación directa del estudiante en actividades de análisis, discusión y resolución de problemas que lo desafían cognitivamente.

En el campo de la ingeniería de software, esta metodología favorece la construcción significativa del conocimiento a partir de la práctica, la experimentación y el error. Como señala Freeman et al. (2019), los entornos de aprendizaje activo mejoran significativamente el rendimiento académico y reducen las tasas de deserción, al promover una comprensión más profunda de los contenidos. En la práctica docente, este enfoque se operacionaliza mediante estrategias de aprender-haciendo, talleres guiados y resolución colaborativa de casos reales vinculados al desarrollo de software.

El valor de este concepto en la experiencia radica en su capacidad para conectar la teoría con la práctica, impulsando en los estudiantes la iniciativa y la reflexión sobre sus propias acciones. Además, el aprendizaje activo se integra con otros enfoques contemporáneos como el learning by doing y la educación basada en competencias, fortaleciendo la autonomía y la capacidad de autorregulación del aprendizaje.

4.2.4. Pensamiento crítico

El desarrollo del pensamiento crítico se erige como un eje transversal de la formación en ingeniería de software, dado que la disciplina demanda la resolución constante de problemas complejos. Facione (2020) define el pensamiento crítico como la habilidad de interpretar, analizar, evaluar y concluir con base en evidencia, un proceso que implica razonamiento lógico y juicio ético. En el contexto educativo, Paul y Elder (2019) sostienen que promover esta competencia implica generar espacios donde el estudiante cuestione sus propias ideas, identifique sesgos y construya argumentos sólidos.

Durante la experiencia, se observó que los estudiantes tendían a reproducir información sin profundizar en su comprensión. A través de debates, simulaciones de proyectos y análisis de errores comunes en el código, se buscó fomentar una cultura de cuestionamiento y argumentación. De acuerdo con González y Ramírez (2021), el pensamiento crítico no solo potencia el desempeño académico, sino que fortalece la empleabilidad y la adaptabilidad profesional. Por tanto, este concepto se convierte en un elemento central para la formación de ingenieros capaces de analizar contextos, tomar decisiones informadas y enfrentar los retos tecnológicos con criterio ético y reflexivo.

4.2.5. Autonomía en el aprendizaje

La autonomía constituye un componente esencial en el desarrollo de competencias profesionales sostenibles. En los primeros semestres universitarios, los estudiantes suelen depender en exceso de la orientación docente, por lo que promover la autorregulación es un desafío pedagógico significativo. Según Zimmerman (2020), la autonomía en el aprendizaje implica la capacidad de planificar, monitorear y evaluar el propio proceso, utilizando estrategias metacognitivas que favorecen la autogestión. Por su parte, Panadero y Broadbent (2018) destacan que esta habilidad se fortalece mediante entornos que estimulan la responsabilidad y el aprendizaje por proyectos.

En la asignatura, la autonomía se promovió a través de actividades donde los estudiantes debían definir objetivos semanales, priorizar tareas y documentar avances en el desarrollo de sus proyectos. Estas prácticas no sólo fortalecieron la disciplina académica, sino también la autoconfianza y la resiliencia frente a los desafíos. Así, la autonomía se articula con el pensamiento crítico y el aprendizaje activo, configurando una tríada indispensable para la formación de ingenieros reflexivos y responsables.

4.2.6. Aprendizaje colaborativo

El aprendizaje colaborativo se reconoce como una metodología que potencia el intercambio de saberes y la construcción colectiva del conocimiento. En el contexto de la ingeniería de software, donde el trabajo en equipo es inherente al desarrollo de proyectos, este enfoque resulta especialmente pertinente. D. W. Johnson et al. (2020) afirman que la colaboración permite a los estudiantes aprender de las perspectivas de otros, fomentar la empatía cognitiva y desarrollar habilidades de comunicación profesional. Además, estudios recientes indican que el aprendizaje colaborativo incrementa la motivación intrínseca y el sentido de pertenencia al grupo (Van Leeuwen & Janssen, 2019).

Durante la experiencia, se implementaron dinámicas de resolución conjunta de problemas y revisión por pares, donde los estudiantes asumieron roles específicos dentro del equipo —analista, programador, evaluador— para simular entornos reales de trabajo. Esta práctica fomenta la corresponsabilidad y la toma de decisiones compartidas, generando un espacio de aprendizaje horizontal donde todos los integrantes podían aportar desde sus fortalezas. En este sentido, el aprendizaje colaborativo no solo refuerza el contenido técnico, sino que constituye una estrategia clave para el desarrollo de habilidades blandas y la socialización del conocimiento.

4.2.7. Mentoría pedagógica

La mentoría pedagógica surge como un concepto articulador entre la guía docente y la autonomía estudiantil. Su propósito es acompañar el proceso formativo sin sustituir la iniciativa del estudiante, creando una relación de orientación reflexiva. Como plantea Ambrosetti y Dekkers (2018), la mentoría implica generar un espacio de diálogo horizontal donde el docente actúa como facilitador, retroalimentando y estimulando la reflexión sobre la práctica. En el ámbito universitario, esta figura ha cobrado relevancia en procesos

de transición académica, especialmente en programas de ingeniería (Ragusa & Chong, 2021).

En la práctica de aula, la mentoría se manifestó en sesiones personalizadas y grupos de acompañamiento donde los estudiantes podían compartir dificultades, recibir orientación y reflexionar sobre su progreso. Este modelo no se limita a brindar soluciones, sino que busca fortalecer la autoconciencia y la responsabilidad. En términos pedagógicos, la mentoría se conecta con el aprendizaje colaborativo y la autonomía, ya que permite al estudiante avanzar hacia un nivel más profundo de comprensión de su propio aprendizaje.

4.2.8. Relación entre los conceptos

Los cinco conceptos estructurantes se encuentran interrelacionados y forman una red conceptual coherente. El aprendizaje activo y el colaborativo proporcionan el entorno metodológico; el pensamiento crítico y la autonomía configuran las competencias cognitivas y metacognitivas que sustentan la acción; mientras que la mentoría pedagógica actúa como mediación que articula todos los elementos, garantizando un acompañamiento orientado al crecimiento personal y profesional del estudiante. En conjunto, estos conceptos ofrecen una base teórica sólida para analizar la experiencia y construir las dimensiones que darán lugar a los indicadores de observación en el módulo siguiente.

En conclusión, identificar estos conceptos permite comprender que la innovación pedagógica en ingeniería no depende únicamente de la tecnología o de los contenidos, sino del modo en que se concibe y acompaña el aprendizaje. La combinación de aprendizaje activo, pensamiento crítico, autonomía, colaboración y mentoría configura un modelo educativo centrado en el estudiante, coherente con las demandas de la educación superior contemporánea. En el siguiente apartado, estos conceptos se transformarán en dimensiones analíticas que permitirán observar, desde una perspectiva sistemática, cómo se concretan en la práctica los procesos formativos que han emergido en esta experiencia docente.

4.2.9. Formulación de dimensiones

Al avanzar en el proceso de sistematización de la experiencia docente desarrollada en la asignatura Introducción a la Ingeniería de Software, resulta necesario organizar los conceptos estructurantes previamente identificados en categorías analíticas más amplias que otorguen coherencia y profundidad al análisis. Estas categorías, denominadas

dimensiones, permiten comprender la experiencia desde distintos planos complementarios, mostrando cómo se articulan los procesos pedagógicos, cognitivos, tecnológicos y relaciones en la práctica educativa.

De acuerdo con Flick (2018), las dimensiones constituyen recortes teórico-analíticos que facilitan la interpretación ordenada de la realidad investigada. En esta sistematización, se establecen tres dimensiones principales: pedagógica, cognitiva-formativa y socioafectiva-colaborativa, que en conjunto configuran el marco conceptual y operativo de la experiencia.

4.2.10. Dimensión pedagógica

La dimensión pedagógica agrupa los conceptos vinculados al aprendizaje activo y la mentoría pedagógica, aludiendo al rol del docente como mediador del aprendizaje y diseñador de experiencias significativas. Esta dimensión se enfoca en la metodología utilizada para guiar a los estudiantes hacia la construcción del conocimiento, promoviendo la participación, la reflexión y la acción práctica.

Como señala Prince (2021), el aprendizaje activo transforma la dinámica del aula al situar al estudiante en el centro del proceso educativo, permitiendo que el conocimiento se construya a través de la exploración y la resolución de problemas reales. En este sentido, la mentoría pedagógica complementa el proceso, ofreciendo un acompañamiento personalizado que potencia la autonomía y la autoconfianza del estudiante (Ragusa & Chong, 2021).

En la experiencia desarrollada, la dimensión pedagógica se manifestó en la implementación de metodologías activas como el learning by doing y la tutoría por pares, que permitieron abordar los contenidos de ingeniería de software desde situaciones auténticas. Los estudiantes no solo recibieron orientaciones conceptuales, sino que aplicaron los principios de la gestión de proyectos en actividades prácticas, diseñando pequeños sistemas y reflexionando sobre los desafíos encontrados.

Esta dimensión, por tanto, revela cómo la práctica docente se transformó en un espacio de aprendizaje dialógico, donde el docente orienta y acompaña, pero es el estudiante quien construye su propio proceso de conocimiento. En palabras de Jara (2018), la práctica reflexiva se vuelve significativa cuando convierte la experiencia en un aprendizaje compartido.

4.2.11. Dimensión cognitiva-formativa

La segunda dimensión se refiere a los procesos mentales, metacognitivos y de desarrollo de competencias que se activan durante el aprendizaje. Incluye los conceptos de pensamiento crítico y autonomía en el aprendizaje, entendidos como capacidades esenciales para la formación de ingenieros competentes y reflexivos. Facione (2020) sostiene que el pensamiento crítico implica interpretar, analizar y evaluar información de manera argumentada para tomar decisiones fundamentadas. Asimismo, la autonomía del aprendizaje, según Zimmerman (2020), representa la habilidad de autorregular las propias estrategias cognitivas y emocionales para alcanzar metas académicas.

En el contexto de la asignatura, esta dimensión se concretó cuando los estudiantes asumieron la responsabilidad de planificar sus avances semanales, analizar problemas de codificación y proponer alternativas de solución fundamentadas. Por ejemplo, durante las prácticas de laboratorio, los grupos debían identificar errores en el código y discutir distintas formas de corregirlos, argumentando sus decisiones. Este ejercicio fortaleció la capacidad de análisis, la autocrítica y la gestión del aprendizaje, aspectos clave en la formación universitaria contemporánea.

Tal como señalan González y Ramírez (2021), las experiencias educativas que promueven la reflexión y la autorregulación contribuyen al desarrollo de un pensamiento autónomo y crítico, indispensable en los campos tecnológicos actuales.

La pertinencia de esta dimensión radica en que ofrece un marco para analizar cómo las metodologías activas no solo generan participación, sino que también impactan en el modo en que los estudiantes piensan, aprenden y aplican el conocimiento. En esta línea, el análisis cognitivo-formativo permite visibilizar la profundidad del aprendizaje alcanzado, superando la visión instrumental del saber técnico para comprenderlo como un proceso de transformación intelectual.

4.2.12. Dimensión socioafectiva-colaborativa

La tercera dimensión agrupa los aspectos relacionados con el aprendizaje colaborativo, la interacción social y la construcción de un clima emocional positivo en el aula. Se fundamenta en la idea de que el aprendizaje es una práctica social que se fortalece a través del trabajo conjunto y del intercambio de saberes. D. W. Johnson et al. (2020) destacan que la colaboración favorece la motivación intrínseca, la empatía y el sentido de pertenencia, elementos esenciales para la cohesión grupal. Por su parte, Van Leeuwen

y Janssen (2019) enfatizan que la guía docente en contextos colaborativos potencia la autorregulación colectiva y la comunicación efectiva.

En la práctica de aula, esta dimensión se manifestó en las actividades de trabajo en equipo para el desarrollo de proyectos de software. Los estudiantes asumieron roles complementarios —analista, programador, validador—, gestionando conflictos, negociando decisiones y distribuyendo responsabilidades. Además, se promovieron instancias de evaluación entre pares, donde los grupos analizaron los productos de otros compañeros y ofrecieron retroalimentación constructiva.

Estas dinámicas no sólo fortalecieron las competencias comunicativas, sino que también permitieron a los estudiantes reconocer la importancia del respeto, la escucha activa y la corresponsabilidad. De este modo, la dimensión socioafectiva-colaborativa se convierte en un eje que conecta el aprendizaje técnico con el desarrollo humano y ético.

Las tres dimensiones formuladas —pedagógica, cognitiva-formativa y socioafectiva-colaborativa— ofrecen una estructura analítica coherente que permite comprender la experiencia docente de manera integral. La primera explica el cómo se enseña, la segunda el cómo se aprende, y la tercera el cómo se convive y se colabora en el proceso educativo. En conjunto, conforman un marco que traduce la práctica pedagógica en categorías de análisis teórico, articulando la reflexión con la acción.

Tal como señala Flick (2018), la formulación de dimensiones es un paso clave para sistematizar experiencias educativas, ya que permite pasar del relato empírico a la interpretación académica. En el siguiente apartado, estas dimensiones se operacionalizarán mediante la construcción de indicadores observables, que harán posible verificar de manera más precisa cómo se concretaron los aprendizajes y transformaciones en la práctica.

4.2.13. Construcción de indicadores

Avanzar en la construcción de indicadores representa un paso clave dentro del proceso de sistematización, pues permite transformar las dimensiones formuladas en categorías verificables y observables. Tal como señalan Flick (2018) y Yin (2018), los indicadores constituyen señales empíricas que vinculan la teoría con la práctica, mostrando cómo se manifiestan las dimensiones en la realidad educativa. En este sentido, cada indicador se convierte en una evidencia concreta de que el proceso pedagógico ha generado un cambio o resultado observable en los estudiantes. En la experiencia desarrollada en la asignatura Introducción a la Ingeniería de Software, los indicadores permiten verificar

la presencia y efectividad de las tres dimensiones identificadas: pedagógica, cognitiva-formativa y socioafectiva-colaborativa.

4.2.14. Indicadores de la dimensión pedagógica

La dimensión pedagógica se relaciona con las estrategias de enseñanza y acompañamiento implementadas por el docente, especialmente a través del aprendizaje activo y la mentoría pedagógica. Esta dimensión busca evidenciar cómo la práctica docente promueve la participación, la autonomía y la reflexión de los estudiantes.

Entre los indicadores formulados se encuentran:

1. El estudiante participa activamente en simulaciones de gestión de proyectos de software, asumiendo roles definidos y aplicando metodologías ágiles para planificar tareas, evaluar riesgos y proponer soluciones preventivas.
2. El docente acompaña el proceso de aprendizaje mediante sesiones de mentoría en las que orienta el uso de herramientas de control de versiones (GitHub) para la colaboración y seguimiento del proyecto.
3. Las actividades de aula evidencian la aplicación del enfoque learning by doing mediante la elaboración de planes de gestión de riesgos, tableros Kanban y reportes de control de cambios en los repositorios de GitHub.

Estos indicadores reflejan cómo el aula se convierte en un entorno de simulación profesional, donde los estudiantes aplican prácticas de gestión de proyectos reales. Por ejemplo, en un caso de estudio, los equipos debían identificar y clasificar riesgos en su proyecto de software, asignando probabilidades, impactos y estrategias de mitigación; posteriormente, registraban los cambios o incidencias en GitHub con mensajes de commit documentados.

De acuerdo con Prince (2021), este tipo de práctica permite vincular la teoría con la acción, promoviendo la reflexión sobre los procesos y la toma de decisiones. Asimismo, Ragusa y Chong (2021) sostienen que la mentoría guiada fortalece la capacidad de los estudiantes para gestionar proyectos de manera autónoma, potenciando la calidad de los productos desarrollados y la confianza profesional.

4.2.15. Indicadores de la dimensión cognitiva-formativa

Esta dimensión se centra en los procesos mentales y metacognitivos que favorecen el desarrollo del pensamiento crítico, la autorregulación y la transferencia del conocimiento. Los indicadores derivados buscan mostrar cómo los estudiantes no solo comprenden los contenidos, sino que también los aplican y evalúan críticamente.

Los indicadores propuestos son:

1. El estudiante analiza críticamente los riesgos identificados en el proyecto y propone estrategias de mitigación basadas en evidencia técnica y priorización de impacto.
2. El estudiante demuestra autonomía en la gestión del proyecto mediante el uso responsable del control de versiones en GitHub, documentando cambios, revisando pull requests y gestionando conflictos de integración.
3. Se observan informes y presentaciones donde los estudiantes justifican decisiones de arquitectura, control de versiones y gestión del riesgo, sustentadas en criterios de calidad y eficiencia.

Estos indicadores reflejan la capacidad del estudiante para pensar como un ingeniero en formación, utilizando la evidencia para fundamentar sus decisiones. Por ejemplo, durante el desarrollo del proyecto, los equipos debían analizar la matriz de riesgos y registrar sus decisiones en el repositorio del proyecto, explicando en los issues de GitHub por qué priorizaban ciertas tareas o correcciones.

Tal como señala Facione (2020), el pensamiento crítico se expresa cuando el estudiante evalúa sus propias acciones y toma decisiones con base en información verificable. Además, Zimmerman (2020) plantea que la autorregulación del aprendizaje se manifiesta en la capacidad de planificar, monitorear y ajustar estrategias, algo que en la ingeniería de software puede observarse en la gestión autónoma del código y de las versiones del producto.

4.2.16. Indicadores de la dimensión socioafectiva-colaborativa

Esta dimensión aborda las relaciones interpersonales, la comunicación y el trabajo en equipo como factores esenciales del aprendizaje en ingeniería de software. Los indicadores formulados permiten evidenciar cómo la colaboración y el clima emocional positivo favorecen el desempeño académico y la motivación estudiantil.

Los indicadores propuestos son:

1. Los equipos de trabajo muestran interacciones colaborativas, distribuyen responsabilidades y construyen soluciones conjuntas.
2. El estudiante demuestra empatía, respeto y disposición para la retroalimentación constructiva en las actividades grupales.
3. Se generan productos colaborativos que reflejan cohesión, coordinación y sentido de pertenencia en el grupo.

Durante la implementación de proyectos colaborativos, los grupos aplicaron técnicas de revisión por pares, compartiendo avances y proponiendo mejoras al trabajo de otros equipos. Este tipo de práctica coincide con lo expuesto por D. W. Johnson et al. (2020), quienes sostienen que la interacción positiva en grupos promueve tanto el aprendizaje académico como el desarrollo socioemocional. Asimismo, Van Leeuwen y Janssen (2019) destacan que la guía docente en entornos colaborativos potencia la regulación conjunta del aprendizaje y la responsabilidad compartida.

De este modo, estos indicadores permiten observar de manera verificable cómo la colaboración trasciende lo operativo, convirtiéndose en un espacio de construcción colectiva y emocional del conocimiento.

La formulación de indicadores constituye el puente que conecta la teoría con la práctica, al traducir las dimensiones en evidencias observables. En este caso, los indicadores pedagógicos reflejan la dinámica activa y acompañada del aprendizaje; los cognitivo-formativos muestran el desarrollo del pensamiento crítico y la autorregulación mediante la gestión del riesgo y el uso de control de versiones; y los socioafectivos-colaborativos evidencian la interacción y la construcción colectiva del conocimiento.

En conjunto, estos indicadores ofrecen una herramienta sólida para evaluar el impacto de la experiencia docente desde un enfoque integral, alineado con los principios de la sistematización educativa. Según Yin (2018), contar con indicadores claros y coherentes garantiza la credibilidad y consistencia del análisis, permitiendo posteriormente construir una matriz de verificación que fortalezca el carácter académico del capítulo.

4.2.17. Fuentes

Fuentes y métodos de verificación

Verificar los indicadores definidos en la sistematización es un paso esencial para asegurar la validez académica y la credibilidad del proceso. Tal como plantea Flick (2018), la rigurosidad de un análisis educativo no depende únicamente de las interpretaciones teóricas, sino de la manera en que estas se sustentan en evidencias concretas y verificables.

En esta etapa, la reflexión pedagógica se convierte en un ejercicio de documentación: se trata de mostrar, con pruebas tangibles, cómo las dimensiones pedagógica, cognitiva-formativa y socioafectiva-colaborativa se manifestaron realmente en la experiencia de aula. Por ello, la selección adecuada de fuentes y métodos de verificación permite traducir los aprendizajes vividos en información analizable, coherente y sustentada en la práctica.

Fuente seleccionadas

Las fuentes son los materiales y registros producidos durante la experiencia docente que permiten constatar la existencia de los indicadores previamente formulados. En este caso, se han identificado cuatro tipos principales de fuentes: documentales, digitales, testimoniales y observacionales.

Fuentes documentales

Incluyen los planes de proyecto, cronogramas de gestión de riesgos, actas de avance semanal y rúbricas de evaluación elaboradas por los estudiantes. Estos documentos reflejan la aplicación del enfoque de aprendizaje activo y la apropiación de metodologías ágiles en la gestión de proyectos de software. La comparación de las versiones iniciales y finales de los entregables permitirá verificar la evolución de las competencias cognitivas y la capacidad de autorregulación del aprendizaje.

Fuentes digitales

Comprenden los repositorios de GitHub utilizados por los equipos de estudiantes. Estos registros contienen evidencia directa de las prácticas de control de versiones, las estrategias de trabajo colaborativo y el seguimiento del progreso técnico. Los historiales de commits, la frecuencia de actualización y los comentarios en issues servirán para ob-

servar el cumplimiento de los indicadores relacionados con la autonomía, la organización y la gestión de riesgos en el desarrollo de software.

Fuentes testimoniales

Se recogen a través de encuestas de percepción y entrevistas breves a los estudiantes. Estas fuentes permiten acceder a la dimensión subjetiva de la experiencia: cómo valoraron el acompañamiento docente, el trabajo en equipo y su propio aprendizaje. De acuerdo con Jara (2018), el testimonio constituye una evidencia valiosa porque otorga voz a los protagonistas y revela la transformación educativa desde su perspectiva personal.

Fuentes observacionales

Corresponden a notas de campo del docente, registros de participación en clase y observaciones de sesiones prácticas en el laboratorio. Estos registros permiten identificar comportamientos concretos asociados a los indicadores, como la distribución de roles, la resolución colaborativa de problemas o el uso efectivo de herramientas de planificación.

Cada una de estas fuentes aporta un tipo distinto de evidencia, y su integración asegura una mirada completa sobre la experiencia, respetando el principio de triangulación de datos que, según Yin (2018), refuerza la validez de los estudios de caso educativos.

4.2.18. Métodos de verificación

Los métodos de verificación son las estrategias que permiten analizar y validar la información contenida en las fuentes. En esta experiencia se aplicarán principalmente tres métodos: análisis comparativo de productos, análisis de contenido y triangulación de evidencias.

4.2.19. Análisis comparativo de productos

Este método se aplicará a los planes de proyecto, cronogramas y repositorios de GitHub. Permitirá contrastar los avances entre las fases iniciales y finales de los proyectos, observando la mejora en la organización, la gestión de riesgos y la documentación técnica. Por ejemplo, se analizará cómo los equipos pasaron de registrar tareas básicas a manejar incidencias complejas o pull requests con justificaciones detalladas. Este análisis

mostrará la evolución de los indicadores vinculados a la dimensión pedagógica, especialmente aquellos que evidencian aprendizaje activo y autonomía.

4.2.20. Análisis de contenido

Se utilizará con las encuestas y entrevistas aplicadas a los estudiantes. Este método busca identificar temas recurrentes, percepciones compartidas y cambios en la actitud hacia el aprendizaje y la colaboración. Los testimonios se codificarán en categorías relacionadas con los indicadores de pensamiento crítico, motivación y clima socioafectivo. Tal como sugiere Flick (2018), este tipo de análisis cualitativo permite construir sentido a partir de la experiencia narrada, reconociendo patrones significativos en los discursos.

4.2.21. Triangulación de evidencias

Finalmente, se integrarán los resultados de las fuentes documentales, digitales y testimoniales para obtener una visión más completa y confiable. Este proceso de triangulación garantiza que la interpretación de los datos no dependa de una sola fuente, sino de la convergencia de múltiples perspectivas. Yin (2018) argumenta que esta estrategia incrementa la credibilidad del estudio y reduce los sesgos del observador, mientras que Jara (2018) señala que la triangulación convierte la sistematización en un proceso de validación pedagógica, donde la experiencia adquiere carácter científico.

El uso de fuentes variadas y métodos de verificación complementarios garantiza la solidez del análisis de la experiencia docente. Los documentos producidos por los estudiantes, los registros digitales en GitHub y los testimonios recabados constituyen un corpus empírico que respalda cada indicador. A través del análisis comparativo, el análisis de contenido y la triangulación, será posible demostrar con rigor cómo las dimensiones pedagógica, cognitiva y socioafectiva se materializaron en la práctica.

Así, esta etapa no solo valida los resultados de la experiencia, sino que también prepara el camino para la construcción de la matriz final de dimensiones – indicadores – fuentes – métodos, producto central del módulo. En términos de Flick (2019), la sistematización alcanza su madurez cuando las evidencias empíricas dialogan con la teoría, mostrando que la docencia universitaria puede y debe analizarse con el mismo rigor que cualquier otro campo de investigación.

4.2.22. Justificación teórica del conjunto

La construcción del andamiaje conceptual y operativo desarrollado a lo largo de este apartado responde a la necesidad de dotar de solidez teórica a una práctica educativa que busca trascender el relato vivencial para convertirse en conocimiento compatible. En este proceso, se integraron los conceptos estructurantes, las dimensiones, los indicadores, y los métodos de verificación que sustentan la sistematización de la experiencia docente en la asignatura Introducción a la Ingeniería de Software.

Cada una de estas decisiones conceptuales no fue arbitraria, sino el resultado de un análisis que pone en diálogo la práctica pedagógica con marcos teóricos contemporáneos de la educación superior, la ingeniería de software y la investigación cualitativa.

La elección de los conceptos estructurantes —aprendizaje activo, pensamiento crítico, autonomía, aprendizaje colaborativo y mentoría pedagógica— se justifica en su relevancia para los contextos formativos actuales. El aprendizaje activo, según Prince (2021), transforma la relación docente-estudiante al situar al aprendiz como protagonista del proceso, lo que resulta esencial en disciplinas tecnológicas donde la práctica y la resolución de problemas reales constituyen el eje del aprendizaje.

Asimismo, la mentoría pedagógica refuerza el vínculo formativo mediante un acompañamiento reflexivo, lo que Ragusa y Chong (2021) consideran un factor determinante en la retención y éxito de los estudiantes de ingeniería. Estos conceptos, interrelacionados, ofrecen una base teórica coherente con los desafíos que enfrenta la docencia universitaria contemporánea.

Las dimensiones formuladas - pedagógica, cognitiva -formativa y socioafectiva - colaborativa - permiten estructurar la experiencia desde una perspectiva integral. La dimensión pedagógica abarca el diseño y la implementación de estrategias didácticas activas; la cognitiva-formativa profundiza en el desarrollo de competencias intelectuales, como el pensamiento crítico y la autorregulación; y la socioafectiva-colaborativa aborda los procesos interpersonales y emocionales del aprendizaje en equipo.

Esta organización responde al planteamiento de Flick (2018), quien sostiene que las categorías analíticas en una sistematización deben reflejar tanto la complejidad del fenómeno educativo como su coherencia interna. En este sentido, las dimensiones seleccionadas no solo agrupan los conceptos, sino que también ofrecen un marco interpretativo para comprender cómo la práctica docente impacta en el desarrollo integral del estudiante.

Los indicadores operativizan las dimensiones y las convierten en observables y verificables. La construcción de estos indicadores responde al principio de validez empírica que Yin (2018) y R. E. Stake (2020) destacan en los estudios de caso, donde las categorías deben traducirse en evidencias que permitan constatar su presencia en la práctica.

Por ejemplo, en la dimensión pedagógica, indicadores como la aplicación de metodologías ágiles, la elaboración de planes de gestión de riesgos o el uso del control de versiones en GitHub permiten comprobar cómo los estudiantes integran la teoría con la acción.

En la dimensión cognitiva-formativa, indicadores asociados a la reflexión crítica y a la toma de decisiones fundamentadas revelan la madurez del pensamiento técnico y analítico. Esta lógica de verificación garantiza que la sistematización trascienda la descripción narrativa para constituirse en un proceso de análisis fundamentado y replicable.

En coherencia con lo anterior, las fuentes y métodos de verificación seleccionados aseguran la credibilidad de la información recopilada. Las fuentes documentales (planificaciones, informes de avance y rúbricas), digitales (repositorios GitHub), testimoniales (encuestas y entrevistas) y observacionales (notas de campo y registros de participación) se complementan a través de la triangulación de evidencias, garantizando la confiabilidad del análisis.

Este enfoque responde a lo que Yin (2018) denomina una estrategia de convergencia metodológica, donde la diversidad de fuentes permite observar un mismo fenómeno desde múltiples perspectivas. Jara (2018) añade que, en la sistematización de experiencias, las evidencias son el puente entre la vivencia pedagógica y el conocimiento académico, pues permiten legitimar las transformaciones que se narran.

La coherencia entre los niveles del andamiaje -conceptos, dimensiones, indicadores, fuentes y métodos- es lo que otorga validez y sentido al conjunto. Siguiendo a Flick (2018), una sistematización adquiere rigor científico cuando existe correspondencia entre los constructos teóricos y los procedimientos de verificación empírica.

De igual modo, Carlino (2019) plantea que la escritura académica no solo comunica resultados, sino que también construye identidad docente e investigativa, convirtiendo la reflexión en un proceso de profesionalización. Así, la presente sistematización no se limita a narrar una práctica, sino que configura un modelo replicable de innovación pedagógica sustentado en principios teóricos y metodológicos consistentes.

Finalmente, el andamiaje teórico construido refuerza el carácter formativo de la docencia universitaria en ingeniería. La integración de metodologías activas, herramientas

tecnológicas y estrategias de acompañamiento docente responde a la necesidad de formar ingenieros capaces de aprender de manera autónoma, reflexiva y colaborativa. Como señalan Zawacki-Richter y Kerres (2020), el éxito de los entornos de aprendizaje en ingeniería depende de la coherencia entre el diseño pedagógico, los instrumentos de evaluación y las estrategias de aprendizaje autorregulado. Por tanto, esta justificación teórica demuestra que cada componente del módulo —conceptos, dimensiones, indicadores y métodos— se articula en una estructura sólida que permite analizar, comprender y mejorar la práctica educativa en el campo de la ingeniería de software.

4.3. Integración curricular y desarrollo de competencias en la formación del ingeniero de software

4.3.1. Transición al vínculo curricular

Al culminar el proceso de fundamentación teórica y operativa de la experiencia desarrollada en la asignatura *Introducción a la Ingeniería de Software*, se evidencia que los aprendizajes alcanzados no se limitan al plano metodológico, sino que se integran de manera coherente con el proyecto formativo de la carrera. Las dimensiones construidas -pedagógica, cognitiva-formativa y socioafectiva-colaborativa- permiten comprender cómo la práctica docente contribuye al desarrollo de competencias profesionales clave, fortaleciendo la relación entre teoría y práctica en los primeros niveles de formación universitaria.

Desde una perspectiva curricular, la experiencia se articula con el perfil de egreso del Ingeniero de Software de la UNEMI, quien se caracteriza por su capacidad para desarrollar soluciones tecnológicas de calidad, trabajar colaborativamente, liderar equipos multidisciplinarios y actuar con responsabilidad ética y compromiso social. En este sentido, los aprendizajes logrados en el aula -aprendizaje activo, pensamiento crítico, autonomía y trabajo colaborativo- responden directamente a los resultados de aprendizaje del programa, al promover la aplicación de estándares profesionales, la comunicación efectiva y la innovación mediante el uso de tecnologías emergentes. La incorporación de estrategias como la mentoría pedagógica y el aprendizaje basado en proyectos favorece la formación de estudiantes autónomos y reflexivos, capaces de enfrentar problemas reales del campo de la ingeniería con una visión integral y ética.

Esta experiencia, además, se convierte en un punto de inflexión en el proceso formativo, al evidenciar que la enseñanza de la ingeniería de software debe trascender la instrucción técnica para situarse en el desarrollo de competencias blandas, cognitivas y socioemocionales que sustentan la empleabilidad y la actualización profesional continua.

Así, la práctica sistematizada no solo refuerza el perfil de egreso institucional, sino que también aporta una mirada pedagógica innovadora para la formación inicial en ingeniería, constituyéndose en una referencia para el rediseño de estrategias curriculares orientadas al aprendizaje significativo y al fortalecimiento del vínculo entre la docencia y el desarrollo profesional.

4.3.2. Identificación de competencias del perfil

La experiencia desarrollada en la asignatura Introducción a la Ingeniería de Software permitió evidenciar cómo la práctica docente puede articularse con las competencias del perfil de egreso de la carrera, convirtiéndose en un espacio de formación integral para los estudiantes de primer semestre.

En este nivel inicial, el desafío no radica únicamente en la adquisición de conocimientos técnicos, sino en la construcción de una base competencial que combine la comprensión conceptual, la colaboración efectiva y la autonomía en el aprendizaje. Por ello, identificar las competencias que se fortalecen a través de esta experiencia resulta clave para evidenciar su alineación con el currículo y el proyecto formativo institucional.

De acuerdo con el perfil de egreso de la carrera de Ingeniería de Software de la UNE-MI, el profesional debe ser capaz de desarrollar, mantener e innovar sistemas de software de calidad, aplicando estándares internacionales y buenas prácticas; trabajar en equipos multidisciplinarios; liderar proyectos; y actuar con ética y responsabilidad social.

En esta línea, las competencias que se consolidaron en la práctica docente fueron principalmente: trabajo colaborativo, pensamiento crítico, autonomía en el aprendizaje, responsabilidad ética y comunicación efectiva. Estas competencias, tanto genéricas como específicas, se integran de forma transversal al proceso formativo, constituyendo el núcleo de la formación del ingeniero de software contemporáneo.

La competencia de trabajo colaborativo se manifestó en la experiencia a través del desarrollo de proyectos en equipo, donde los estudiantes asumieron roles complementarios -analista, programador y validador-, simulando entornos reales de producción de software. Esta metodología permitió fortalecer habilidades de coordinación, negociación

y corresponsabilidad, aspectos que según D. W. Johnson et al. (2020) resultan esenciales para el aprendizaje cooperativo en contextos de ingeniería.

Asimismo, el trabajo en equipo fomenta la empatía profesional y la capacidad de resolver conflictos de manera constructiva, competencias destacadas en el perfil de egreso institucional. En un campo altamente interdependiente como el desarrollo de software, estas habilidades sociales y de colaboración constituyen un pilar fundamental para la empleabilidad y el éxito profesional.

Por su parte, la competencia de pensamiento crítico se relacionó con la capacidad de los estudiantes para analizar y evaluar los problemas planteados durante la construcción de proyectos, argumentando sus decisiones técnicas con base en evidencia y criterios de calidad.

Tal como sostiene Facione (2020), el pensamiento crítico permite a los futuros profesionales emitir juicios fundamentados y asumir decisiones responsables en contextos complejos. En la asignatura, esta competencia se promovió mediante la revisión de código, la detección de errores lógicos y la reflexión sobre las estrategias de resolución implementadas. Estas prácticas fomentaron en los estudiantes la capacidad de razonar con rigor, una competencia que, según Lai (2022), constituye un indicador de madurez cognitiva y profesional en entornos STEM.

La autonomía en el aprendizaje fue otra competencia central, consolidada a través de la planificación semanal de avances, la gestión de tareas y la autoevaluación continua de los logros alcanzados. Zimmerman (2020) define la autorregulación del aprendizaje como la capacidad de planificar, monitorear y evaluar el propio proceso formativo.

En el contexto de la ingeniería de software, esta habilidad permite que los estudiantes desarrollen la disciplina y la resiliencia necesarias para enfrentar proyectos de largo plazo y adaptarse a las demandas cambiantes del mercado tecnológico. La autonomía, además, se conecta directamente con la competencia de aprendizaje permanente, presente en el perfil de egreso, que enfatiza la importancia de mantenerse actualizado en tecnologías emergentes y buenas prácticas de desarrollo.

La responsabilidad ética y social se evidenció en la forma en que los estudiantes asumieron el impacto de las decisiones tecnológicas en la sociedad, discutiendo dilemas éticos relacionados con la privacidad de datos, la equidad digital y la sostenibilidad del software. Según Morales et al. (2021), integrar la dimensión ética en la formación tecnológica favorece la construcción de profesionales conscientes del alcance social de sus acciones. Esta competencia, presente en el perfil de egreso, trasciende el plano técnico

para consolidar una formación humanista y responsable, coherente con el compromiso de la UNEMI hacia el desarrollo sostenible y el bienestar colectivo.

Finalmente, la comunicación efectiva emergió como una competencia transversal, indispensable tanto para la gestión de proyectos como para la documentación técnica y la presentación de resultados. Hargie (2021) subraya que la comunicación profesional en ingeniería requiere precisión, claridad y empatía, habilidades que se fortalecen mediante la exposición oral, la escritura técnica y la retroalimentación entre pares. En las actividades realizadas, los estudiantes presentaron sus avances, defendieron decisiones de diseño y compartieron aprendizajes, fortaleciendo así la capacidad de expresarse con propiedad en entornos académicos y laborales.

Estas cinco competencias -colaboración, pensamiento crítico, autonomía, ética y comunicación- conforman un entramado coherente que responde a las demandas del perfil de egreso y a los retos contemporáneos de la educación en ingeniería. En la experiencia sistematizada, dichas competencias se articularon de forma integral, mostrando que el proceso formativo no se limita a la enseñanza de contenidos técnicos, sino que abarca el desarrollo de capacidades transversales que preparan a los estudiantes para desempeñarse con eficacia, liderazgo y compromiso social en su vida profesional.

De este modo, la identificación de las competencias del perfil no solo evidencia la pertinencia curricular de la experiencia, sino que proyecta el proceso hacia los resultados de aprendizaje que serán analizados en el siguiente apartado. Estas competencias se constituyen, por tanto, en los pilares sobre los cuales se construyen los aprendizajes observables que reflejan la transformación del estudiante a lo largo del proceso educativo.

4.3.3. Resultados de aprendizaje vinculados

La sistematización de la experiencia desarrollada en la asignatura Introducción a la Ingeniería de Software permitió constatar la correspondencia entre las competencias del perfil de egreso y los resultados de aprendizaje propuestos en el plan curricular de la carrera. En la práctica docente, los resultados de aprendizaje funcionan como un punto de conexión entre las metas formativas del programa y los logros observables alcanzados por los estudiantes.

En el contexto de la Universidad Estatal de Milagro (UNEMI), estos resultados se formulan en coherencia con el Modelo Educativo Institucional (MEI) y el Marco Curricular Institucional (MCI), que establecen como principio fundamental el desarrollo integral de

los estudiantes mediante el aprendizaje activo, reflexivo y colaborativo (Universidad Estatal de Milagro (UNEMI), 2021).

En este sentido, los resultados de aprendizaje fortalecidos a partir de la experiencia se agrupan en tres dimensiones que responden al enfoque por competencias definido por la UNEMI: desarrollo del pensamiento crítico y la resolución de problemas, colaboración y comunicación efectiva en entornos de ingeniería, y actuación ética y compromiso con el aprendizaje continuo. Estas dimensiones, además de alinearse con el perfil de egreso de la carrera, evidencian la coherencia entre la práctica pedagógica y el modelo curricular institucional orientado a resultados.

El primer resultado de aprendizaje evidenciado fue la capacidad para analizar problemas de software y proponer soluciones fundamentadas en principios de ingeniería y criterios de calidad. Durante la experiencia, los estudiantes aplicaron estrategias de análisis, diseño y validación de sistemas, desarrollando la competencia de pensamiento crítico mediante la revisión de código, la evaluación de alternativas y la justificación argumentada de sus decisiones.

Este resultado se vincula directamente con el componente de razonamiento lógico y pensamiento crítico establecido en el MCI, que enfatiza la formación de profesionales capaces de identificar, analizar y resolver problemas complejos de manera autónoma (Universidad Estatal de Milagro (UNEMI), 2023). En las actividades de aula, esta competencia se materializó cuando los equipos discutían errores en sus programas, analizaban sus causas y elaboraban soluciones fundamentadas en estándares de desarrollo.

Un segundo resultado de aprendizaje alcanzado fue la capacidad para comunicarse y trabajar de manera efectiva en equipos multidisciplinarios. Este aprendizaje se desarrolló a través de la planificación, ejecución y presentación de proyectos colaborativos en los que cada estudiante asumió roles específicos, integrando la teoría con la práctica. De acuerdo con el MCI, la comunicación efectiva y el trabajo en equipo son competencias transversales que contribuyen a la construcción de conocimiento compartido, al fortalecimiento de la empatía y al liderazgo en contextos diversos (Universidad Estatal de Milagro (UNEMI), 2021). En la práctica, los estudiantes demostraron avances significativos al utilizar plataformas de control de versiones y entornos colaborativos, evidenciando mejoras tanto en la organización de tareas como en la exposición oral y escrita de los resultados.

El tercer resultado de aprendizaje consolidado fue la demostración de responsabilidad ética y compromiso con el aprendizaje autónomo y permanente. Este resultado se expresó

en la reflexión de los estudiantes sobre el impacto social de las tecnologías que desarrollan, la gestión responsable del tiempo y la autogestión de su propio progreso académico.

Según el Modelo Educativo Institucional de la UNEMI, la formación ética constituye un eje transversal del currículo, orientado al ejercicio profesional responsable y al desarrollo sostenible (Universidad Estatal de Milagro (UNEMI), 2021). En este marco, la autonomía y la ética se entienden como pilares que permiten a los futuros ingenieros enfrentar los desafíos tecnológicos con conciencia social y capacidad de adaptación continua.

La relación entre estos resultados de aprendizaje y las competencias del perfil de egreso es directa y verificable. El pensamiento crítico y la resolución de problemas se asocian con la competencia de innovar y aplicar tecnologías emergentes, la colaboración y la comunicación efectiva se relacionan con la capacidad de liderar proyectos y trabajar en equipos multidisciplinarios, mientras que la ética y el aprendizaje autónomo se alinean con el compromiso institucional de formar profesionales responsables y comprometidos con el desarrollo sostenible. Esta trazabilidad, destacada por el MCI (Universidad Estatal de Milagro (UNEMI), 2023), garantiza la coherencia entre las experiencias formativas del aula y las metas de desempeño profesional esperadas al final de la carrera.

En síntesis, los resultados de aprendizaje alcanzados en la experiencia evidencian la efectividad del enfoque por competencias adoptado por la UNEMI. Los estudiantes no solo adquirieron conocimientos técnicos, sino que también fortalecieron habilidades cognitivas, comunicativas y éticas que sustentan su perfil profesional.

Este proceso demuestra que la integración entre docencia, práctica y currículo no es un ejercicio formal, sino una estrategia pedagógica que potencia la formación integral del futuro ingeniero de software. En el siguiente apartado, estos aprendizajes se proyectarán en las actividades y evidencias que permiten verificar empíricamente la transformación educativa lograda en el aula.

4.3.4. Actividades y evidencias

El desarrollo de la experiencia en la asignatura Introducción a la Ingeniería de Software se sustentó en una secuencia de actividades planificadas intencionalmente para fortalecer las competencias y resultados de aprendizaje vinculados al perfil de egreso de la carrera. Estas actividades respondieron al principio de coherencia pedagógica entre la teoría y la práctica, promoviendo el aprendizaje activo, el trabajo colaborativo y la reflexión

crítica como ejes centrales del proceso formativo. De este modo, cada acción en el aula fue concebida no solo como una tarea académica, sino como una oportunidad para generar evidencias verificables de los aprendizajes alcanzados por los estudiantes.

Las actividades se estructuraron en tres momentos pedagógicos: exploración, aplicación y reflexión. En la fase de exploración, los estudiantes participaron en sesiones de diagnóstico y discusión guiada sobre los fundamentos de la ingeniería de software, identificando los problemas comunes en la gestión de proyectos tecnológicos.

Estas dinámicas permitieron activar los conocimientos previos y contextualizar los desafíos que enfrentarían posteriormente. Las evidencias generadas en esta etapa fueron fichas de análisis individual, foros de discusión en el aula virtual y registros reflexivos, que mostraron la comprensión inicial de los conceptos básicos y la disposición para el trabajo colaborativo.

Durante la fase de aplicación, se implementaron actividades basadas en el aprendizaje activo y la simulación de entornos profesionales. Los estudiantes trabajaron en equipos para desarrollar un proyecto de software de pequeña escala, aplicando principios de análisis de requerimientos, diseño modular y control de versiones mediante GitHub.

Cada grupo asumió roles diferenciados -analista, programador, validador y líder de proyecto- con el fin de promover la corresponsabilidad y el liderazgo colaborativo. Entre las principales evidencias producidas se encuentran los repositorios de código en GitHub, los informes técnicos del proyecto, las actas de reuniones semanales y las rúbricas de evaluación de desempeño, que documentaron tanto los procesos como los productos del aprendizaje.

Estas evidencias resultaron especialmente relevantes porque permitieron observar la consolidación de los resultados de aprendizaje vinculados a la resolución de problemas, el trabajo en equipo y la comunicación efectiva. Tal como señalan Biggs y Tang (2011), la coherencia entre objetivos, actividades y evaluación es esencial para garantizar la validez del proceso formativo. En este caso, la experiencia demostró que los proyectos colaborativos constituyen un medio eficaz para integrar los saberes técnicos y las competencias transversales del ingeniero de software, permitiendo que el aprendizaje sea significativo, observable y transferible.

Finalmente, en la fase de reflexión, se promovió la autoevaluación y la metacognición mediante la elaboración de bitácoras individuales de aprendizaje y sesiones de retroalimentación grupal. Los estudiantes analizaron los aciertos y dificultades encontradas du-

rante el desarrollo del proyecto, reflexionando sobre la aplicación de los principios éticos, la responsabilidad en la gestión del tiempo y la toma de decisiones.

Estas evidencias, junto con las encuestas de satisfacción y las observaciones del docente, ofrecieron una visión integral del impacto de la experiencia en la formación de los participantes. Según Kolb (2015), la reflexión sobre la práctica permite transformar la experiencia vivida en conocimiento profundo, consolidando la autonomía y la capacidad crítica del aprendiz.

La pertinencia de las evidencias recogidas radica en que reflejan de manera tangible el cumplimiento de los resultados de aprendizaje propuestos en el plan curricular de la carrera y en el Marco Curricular Institucional de la UNEMI. Cada evidencia -ya sea un producto tecnológico, un registro reflexivo o un indicador de desempeño- constituye una muestra verificable de que los estudiantes avanzaron hacia el perfil profesional esperado: un ingeniero de software con pensamiento crítico, ética profesional, habilidades colaborativas y compromiso con la calidad del desarrollo tecnológico.

En conclusión, las actividades y evidencias descritas demuestran que la práctica docente fue diseñada con coherencia pedagógica y pertinencia curricular, asegurando la trazabilidad entre actividad → resultado → evidencia. Este enfoque permitió evidenciar que la experiencia no solo fortaleció los aprendizajes técnicos, sino también las competencias cognitivas, socioafectivas y éticas necesarias para la formación integral del futuro profesional. En el siguiente apartado, estas evidencias servirán de base para analizar la alineación curricular y el grado de correspondencia entre la práctica sistematizada y el modelo educativo institucional de la UNEMI.

4.3.5. Reflexión sobre la alineación curricular

Reflexionar sobre la alineación curricular implica reconocer la importancia de que cada práctica docente se integre de manera coherente al perfil de egreso, a los resultados de aprendizaje y a los principios formativos de la carrera. En el caso de la experiencia desarrollada en la asignatura Introducción a la Ingeniería de Software, esta alineación fue clave para garantizar que las estrategias implementadas no fueran acciones aisladas, sino componentes articulados del proceso formativo que caracteriza al ingeniero de software de la Universidad Estatal de Milagro. En coherencia con lo que plantea el Marco Curricular Institucional (Universidad Estatal de Milagro (UNEMI), 2023), la práctica docente debe promover aprendizajes significativos y transferibles, alineados con un currículo fle-

xible y orientado por competencias, donde la docencia se conciba como un espacio de innovación y reflexión continua.

La experiencia permitió constatar una sólida correspondencia entre las actividades de aula y los propósitos curriculares de la carrera. Las estrategias de aprendizaje activo, la resolución colaborativa de problemas y la integración de herramientas tecnológicas (como GitHub y entornos de simulación) evidenciaron la conexión directa con las competencias del perfil de egreso relacionadas con la comunicación efectiva, la ética profesional y la innovación tecnológica.

Esta coherencia refleja lo que BBiggs y Tang (2011) denominan alineación constructiva: la necesidad de que los objetivos, las actividades y la evaluación respondan de forma integrada a los resultados de aprendizaje esperados. En este sentido, la experiencia fortaleció la formación integral del estudiante al fomentar tanto las competencias técnicas propias de la ingeniería como las socioemocionales y metacognitivas que demanda la práctica profesional contemporánea.

Entre las principales fortalezas observadas destaca la pertinencia metodológica de las actividades diseñadas. La secuencia de exploración, aplicación y reflexión permitió avanzar de lo conceptual a lo práctico, haciendo visible la progresión del aprendizaje. Los proyectos colaborativos y las bitácoras de autoevaluación demostraron que los estudiantes no solo comprendieron los fundamentos de la disciplina, sino que también desarrollaron autonomía, pensamiento crítico y capacidad de autorregulación.

Como señalan Barnett y Jackson (2019), la educación superior debe preparar a los estudiantes para gestionar la complejidad y la incertidumbre del conocimiento profesional, promoviendo la reflexión crítica sobre su propio aprendizaje. Desde esta perspectiva, la alineación curricular se convierte en un proceso dinámico que articula la formación técnica con la capacidad de adaptación y el pensamiento ético.

No obstante, también se identificaron algunas brechas y tensiones en el proceso. Una de ellas fue la necesidad de fortalecer la conexión entre los resultados de aprendizaje del primer semestre y los desafíos de cursos posteriores, especialmente en la aplicación sistemática de metodologías ágiles y en la evaluación continua de proyectos. Esta situación coincide con lo que expone Zabalza (2019), quien plantea que los currículos universitarios deben revisarse periódicamente para mantener su coherencia horizontal y vertical, evitando fragmentaciones en la formación por competencias. Asimismo, se evidenció que la carga académica y el tiempo destinado a la tutoría personalizada resultaron limitantes para acompañar a todos los grupos con la profundidad requerida. Estas brechas sugieren la

necesidad de reforzar la integración entre docentes de distintas asignaturas, promoviendo espacios de co-docencia y articulación interdisciplinaria.

A nivel institucional, el ejercicio de vincular la práctica con el currículo reafirma la importancia de comprender la docencia como una práctica reflexiva. La Universidad Estatal de Milagro (UNEMI) (2023) plantea que la alineación curricular no debe entenderse como un cumplimiento formal, sino como una estrategia para garantizar la calidad académica y la pertinencia social de la formación. En este sentido, el proceso de sistematización se convierte en una herramienta para revisar críticamente las propias decisiones pedagógicas, reconocer aciertos y proyectar mejoras.

Entre los aprendizajes más significativos derivados de esta reflexión se encuentra la convicción de que la enseñanza en ingeniería debe integrar la innovación tecnológica con una mirada humanista y ética, fortaleciendo la relación entre la práctica profesional y la responsabilidad social.

En conclusión, la experiencia desarrollada estuvo sólidamente alineada con el currículo de la carrera, contribuyendo a la consolidación de un modelo pedagógico basado en competencias y en la formación integral del estudiante. Sin embargo, esta reflexión invita también a mantener una actitud crítica frente al propio diseño curricular, promoviendo una actualización constante y un diálogo permanente entre las prácticas docentes y las demandas del entorno profesional. De este modo, la alineación curricular se convierte en un ejercicio de mejora continua que reafirma el compromiso docente con la calidad educativa y la innovación formativa, principios que sustentan la misión de la UNEMI y el propósito de formar ingenieros capaces de transformar su realidad desde el conocimiento, la ética y la creatividad.

4.4. Diseño e implementación del ecosistema estratégico de aprendizaje en ingeniería de software

4.4.1. Transición hacia la operacionalización estratégica

Hasta este punto, la sistematización ha permitido mostrar la coherencia curricular de la experiencia desarrollada en la carrera de Ingeniería de Software de la Universidad Estatal de Milagro (UNEMI). A través del análisis de competencias, resultados de aprendizaje y evidencias, se evidenció una correspondencia entre los objetivos formativos del cur-

so y las prácticas pedagógicas implementadas. No obstante, comprender la experiencia únicamente desde el plano curricular sería insuficiente si no se examina cómo dichas intenciones se concretaron en el aula. Por ello, esta sección marca la transición hacia la operacionalización estratégica, en la que se describe cómo las estrategias planificadas se tradujeron en acciones reales que sostuvieron el proceso de enseñanza-aprendizaje.

Adoptar una mirada estratégica implica reconocer que la formación en ingeniería requiere de decisiones operativas que integren metodologías activas y herramientas tecnológicas propias del campo. En este sentido, la experiencia se sustentó en estrategias núcleo orientadas al trabajo por proyectos, la aplicación de metodologías ágiles y el uso de plataformas colaborativas como GitHub para el control de versiones y la gestión del progreso. Estas acciones permitieron que los estudiantes asumieran roles propios de un equipo de desarrollo profesional, aplicando buenas prácticas en la planificación, seguimiento y documentación del software.

De forma complementaria, se implementaron estrategias de soporte basadas en el acompañamiento docente continuo, el uso de rúbricas compartidas y la retroalimentación técnica, asegurando coherencia y trazabilidad entre los aprendizajes y las competencias profesionales. Finalmente, ante contingencias como los retrasos o dificultades técnicas, se aplicaron estrategias de ajuste que promovieron la resolución autónoma y el trabajo colaborativo.

Esta transición hacia la mirada estratégica constituye un paso esencial dentro del capítulo, pues enlaza el diseño curricular con la práctica operativa. A partir de aquí, se expondrán las estrategias núcleo que guiaron la experiencia en gestión de proyectos de software, mostrando cómo la planificación se transformó en acción mediante secuencias metodológicas claras, decisiones adaptativas y recursos tecnológicos que garantizaron el logro de los resultados de aprendizaje propuestos.

4.4.2. Estrategias núcleo en acción

La implementación de la experiencia en la asignatura Introducción a la ingeniería de Software se sustentó en un conjunto de estrategias núcleo orientadas a promover el aprendizaje activo, la gestión colaborativa y el desarrollo de competencias profesionales vinculadas con el trabajo real en entornos tecnológicos. Estas estrategias constituyeron el corazón de la práctica docente, ya que permitieron transformar los objetivos curriculares en acciones operativas concretas, favoreciendo la participación, la responsabilidad

compartida y la apropiación de herramientas de gestión digital como GitHub, Trello y metodologías ágiles.

La primera estrategia núcleo consistió en la implementación del aprendizaje basado en proyectos (ABP), que permitió a los estudiantes asumir el rol de desarrolladores en un entorno simulado de trabajo real. A través de esta metodología, los equipos planificaron, diseñaron y ejecutaron un proyecto de software siguiendo el ciclo de vida del desarrollo y aplicando buenas prácticas de ingeniería.

Cada grupo gestionó su repositorio en GitHub, donde registró avances, incidencias y versiones del código, lo que fomentó la transparencia y la trazabilidad del proceso. Este enfoque metodológico favoreció la adquisición de competencias técnicas y blandas, como la resolución de problemas, la comunicación efectiva y la gestión del tiempo, en consonancia con lo señalado por Thomas y Leifer (2020), quienes destacan que el ABP impulsa el desarrollo de habilidades de colaboración y pensamiento crítico en contextos de ingeniería.

Una segunda estrategia núcleo fue la integración de metodologías ágiles, particularmente el enfoque Scrum, como marco para la organización y seguimiento del trabajo. Los estudiantes se distribuyeron en roles de Scrum Master, Product Owner y Development Team, participando en reuniones breves de planificación y revisión semanal de avances.

Cada sprint concluyó con una entrega funcional del sistema, la cual fue documentada y evaluada mediante rúbricas previamente definidas. Este modelo de trabajo permitió experimentar la autogestión y la iteración continua, alineándose con la realidad profesional del desarrollo de software. Según Pisoni et al. (2021), el uso de metodologías ágiles en la enseñanza de ingeniería favorece la adaptación al cambio y la capacidad de priorizar tareas, habilidades esenciales en contextos tecnológicos dinámicos.

La tercera estrategia núcleo correspondió a la gestión colaborativa del código mediante control de versiones en GitHub, una herramienta central para la coordinación y el trabajo sincrónico y asincrónico entre los miembros del equipo.

Cada estudiante fue responsable de realizar commits, crear branches para nuevas funcionalidades y resolver pull requests, lo que permitió un seguimiento detallado de las contribuciones individuales y colectivas. Esta práctica no solo fortaleció las competencias técnicas, sino también la ética profesional y la responsabilidad compartida, al evidenciar los aportes de cada integrante. De acuerdo con Zhang et al. (2023), el uso educativo de GitHub potencia la transparencia, la cooperación y la evaluación formativa en proyectos de programación universitaria.

Estas estrategias se articularon de manera complementaria, generando un ecosistema didáctico coherente y alineado con el perfil de egreso del ingeniero de software. La interacción entre el aprendizaje basado en proyectos, la metodología ágil y el control de versiones consolidó una dinámica de trabajo realista, en la que los estudiantes aprendieron haciendo y reflexionando sobre su propio proceso. Las evidencias obtenidas —bitácoras de repositorios, informes de avances, rúbricas de desempeño y productos funcionales— demostraron la efectividad de estas estrategias, evidenciando aprendizajes significativos y sostenibles.

En conclusión, las estrategias núcleo implementadas permitieron que la experiencia trascienda el plano teórico y se consolide como una práctica auténtica de formación profesional. Su coherencia operativa y su enfoque progresivo fortalecieron el aprendizaje autónomo, la colaboración efectiva y la capacidad de los estudiantes para aplicar metodologías de gestión de proyectos en escenarios reales. Este conjunto de acciones sentó las bases para la siguiente fase del capítulo: las estrategias de soporte, que acompañaron y potenciaron la ejecución de las estrategias núcleo, asegurando la sostenibilidad pedagógica y tecnológica de la experiencia.

4.4.3. Estrategias de soporte aplicadas

La ejecución de la experiencia se apoyó en un conjunto de estrategias de soporte que acompañaron la implementación de las estrategias núcleo, garantizando su continuidad, sostenibilidad y efectividad pedagógica. Estas estrategias fueron diseñadas para proporcionar a los estudiantes un entorno de aprendizaje accesible, guiado y colaborativo, en el que pudieran afianzar los conocimientos iniciales sobre la disciplina, familiarizarse con las herramientas tecnológicas de uso profesional y fortalecer su autonomía en el desarrollo de proyectos introductorios.

Una primera estrategia de soporte fue la elaboración y distribución de guías prácticas experimentales, diseñadas para orientar a los estudiantes en el uso de herramientas esenciales de la ingeniería de software, como GitHub para el control de versiones y Trello para la gestión de tareas. Cada guía integró objetivos, pasos secuenciales y ejemplos contextualizados, de modo que los estudiantes pudieran avanzar de manera autónoma en la comprensión del proceso de desarrollo de software.

Este recurso fue complementado con sesiones de acompañamiento sincrónico en aula y tutorías personalizadas en horarios establecidos, donde se resolvieron dudas técnicas

y conceptuales. Según Fernández y Rodríguez (2020), la estructuración de materiales de apoyo bien diseñados potencia el aprendizaje autónomo y reduce las brechas entre la teoría y la práctica en entornos universitarios.

Una segunda estrategia de soporte consistió en el uso integrado de plataformas institucionales y recursos digitales, que facilitaron la comunicación y el seguimiento del progreso. A través de Moodle, se centralizó la entrega de actividades, la publicación de recursos y la retroalimentación continua, mientras que GitHub y Trello permitieron evidenciar el trabajo colaborativo y la planificación de tareas en equipo.

El uso de estas plataformas no solo fortaleció la organización y la trazabilidad de los proyectos, sino que también fomentó la apropiación de herramientas de gestión profesional desde las etapas iniciales de la formación. De acuerdo con García-Holgado y García-Peñalvo (2022), la integración de ecosistemas digitales en la enseñanza de ingeniería favorece la alfabetización tecnológica y mejora la interacción entre estudiantes y docentes, al tiempo que optimiza los procesos de evaluación formativa.

Una tercera estrategia de soporte clave fue la retroalimentación formativa y continua, aplicada en distintos momentos del desarrollo de la experiencia. A través de rúbricas compartidas en Moodle y revisiones periódicas de los repositorios en GitHub, se proporcionaron observaciones específicas sobre la estructura del código, la documentación técnica y la colaboración en equipo. Este proceso permitió a los estudiantes reconocer sus avances y áreas de mejora, desarrollando habilidades de autoevaluación y pensamiento crítico. La retroalimentación fue concebida como un espacio de diálogo pedagógico y no solo como una evaluación sumativa, en coherencia con lo planteado por Boud y Molloy (2019), quienes destacan que la retroalimentación efectiva debe considerarse un proceso interactivo y orientado al desarrollo de la competencia profesional.

Finalmente, se aplicó una estrategia de acompañamiento docente diferenciado, enfocada en atender la diversidad de ritmos de aprendizaje presentes en el aula. Algunos estudiantes requerían apoyo adicional en el manejo de Git y en la lógica de los procesos de desarrollo; por ello, se organizaron tutorías complementarias y pequeños grupos de refuerzo.

Estas acciones fueron esenciales para garantizar la equidad en el aprendizaje y consolidar las bases conceptuales necesarias para cursos posteriores más especializados. Este tipo de acompañamiento responde a la perspectiva de aprendizaje inclusivo y progresivo que promueve la UNEMI, donde el rol docente se redefine como mediador del aprendizaje más que como transmisor de contenidos.

En conjunto, las estrategias de soporte aplicadas consolidaron la experiencia formativa al garantizar la operatividad de las estrategias núcleo y al generar un entorno de aprendizaje integral, flexible y orientado al logro de resultados verificables. La articulación entre guías estructuradas, acompañamiento constante, plataformas tecnológicas y retroalimentación formativa permitió sostener un proceso de enseñanza activo, participativo y centrado en el estudiante. Estas estrategias, además, fomentaron una cultura de trabajo colaborativo y profesional desde los primeros niveles de la carrera, preparando a los estudiantes para enfrentar con mayor solvencia las demandas de las etapas avanzadas del desarrollo de software.

En el siguiente apartado se abordarán las estrategias de contingencia desplegadas, mostrando cómo se gestionaron los imprevistos y desafíos surgidos durante la experiencia, garantizando la continuidad y la coherencia del proceso formativo.

4.4.4. Estrategias de contingencia desplegadas

Durante el desarrollo de la experiencia en la asignatura Introducción a la Ingeniería de Software de la carrera de Ingeniería de Software de la Universidad Estatal de Milagro (UNEMI), surgieron diversos imprevistos que pusieron a prueba la flexibilidad y la capacidad de adaptación tanto del docente como de los estudiantes. Las contingencias fueron entendidas no como obstáculos, sino como oportunidades para fortalecer la autonomía, la resiliencia y la gestión colaborativa en los procesos formativos. En este sentido, la implementación de estrategias de contingencia permitió mantener la coherencia del proyecto, asegurar la continuidad de las actividades y garantizar el logro de los resultados de aprendizaje previstos, incluso frente a condiciones adversas.

Una de las principales contingencias enfrentadas fue la inestabilidad en la conectividad y el acceso a las plataformas digitales, particularmente durante las sesiones de práctica en GitHub y Trello. Algunos estudiantes presentaron dificultades para sincronizar sus repositorios o subir evidencias de avance a tiempo, lo que afectaba la dinámica colaborativa de los equipos.

Ante esta situación, se adoptó una estrategia de flexibilización en los plazos de entrega y la habilitación de canales alternativos de comunicación, como grupos de WhatsApp académicos y foros en Moodle, que permitieron mantener la interacción y la supervisión constante. Esta medida coincidió con las recomendaciones de Hodges et al. (2020), quie-

nes sostienen que la adaptabilidad docente y la diversificación de medios son factores decisivos para sostener la continuidad educativa en entornos mediados por tecnología.

Otra contingencia significativa fue la brecha de conocimientos técnicos iniciales entre los estudiantes, propia de un curso introductorio donde coexistían diferentes niveles de familiaridad con las herramientas de desarrollo. Para contrarrestar esta disparidad, se implementó una estrategia de acompañamiento escalonado, en la que los estudiantes con mayor dominio de GitHub o de metodologías ágiles asumieron roles de tutores pares.

Estas microcomunidades de aprendizaje promovieron la colaboración horizontal, redujeron la ansiedad técnica y fortalecieron la cohesión grupal. Como afirman Zhu y Park (2021), la tutoría entre pares favorece el aprendizaje situado, la cooperación y la autoconfianza en entornos digitales, especialmente en contextos de formación en ingeniería.

Un tercer imprevisto fue la falta de sincronización en la ejecución de las tareas de los equipos de trabajo, derivada de la gestión simultánea de múltiples asignaturas y responsabilidades personales de los estudiantes. Para resolverlo, se adoptó la estrategia de replanificación de los cronogramas de entrega mediante Trello y la implementación de sprints cortos de una semana, que facilitaron un seguimiento más ágil y realista.

Esta medida permitió reducir la acumulación de tareas y mejorar la productividad de los grupos, promoviendo un ritmo de trabajo constante y alcanzable. En coherencia con lo señalado por (Moe et al., 2022), los procesos iterativos y la gestión visual de tareas fortalecen la coordinación y la comunicación en equipos de aprendizaje distribuidos, asegurando la adaptabilidad ante cambios o retrasos.

Asimismo, se presentaron contingencias de tipo motivacional. Algunos estudiantes mostraron una participación intermitente o desmotivación ante los desafíos técnicos del curso. Frente a ello, se implementaron estrategias de refuerzo positivo y gamificación, incorporando insignias digitales y reconocimientos públicos en Moodle a los equipos más constantes y colaborativos. Estas acciones generaron un ambiente de competencia sana y compromiso colectivo. De acuerdo con de Sousa et al. (2021), el uso de la gamificación en entornos educativos incrementa la motivación intrínseca y la persistencia de los estudiantes al integrar elementos lúdicos con metas de aprendizaje claras y alcanzables.

Estas estrategias de contingencia se articularon con las estrategias núcleo y de soporte, formando un entramado flexible que garantizó la continuidad de la experiencia pese a las dificultades. Más allá de resolver situaciones puntuales, estas acciones fortalecieron competencias transversales esenciales en la formación del ingeniero de software, como la resolución de problemas, la toma de decisiones en contextos de incertidumbre y la gestión

efectiva del trabajo en equipo. Además, contribuyeron a consolidar una cultura académica basada en la responsabilidad compartida y en la capacidad de respuesta ante el cambio, aspectos clave de la profesión tecnológica contemporánea.

En síntesis, las estrategias de contingencia desplegadas permitieron transformar los desafíos del aula en oportunidades de aprendizaje significativo. La flexibilidad metodológica, el acompañamiento colaborativo y el uso inteligente de los recursos digitales posibilitaron mantener la calidad educativa sin perder el foco en los resultados curriculares. Este conjunto de medidas reforzó la pertinencia pedagógica de la experiencia, asegurando que la práctica docente se mantuviera sólida y coherente con los principios de la educación en ingeniería. A continuación, se presentará la arquitectura del ecosistema estratégico, que integrará las estrategias núcleo, de soporte y de contingencia en una visión global del proceso, mostrando su articulación y coherencia operativa dentro del capítulo.

4.4.5. Arquitectura del ecosistema estratégico

La experiencia desarrollada en la asignatura se consolidó a través de un ecosistema estratégico conformado por la articulación de estrategias núcleo, de soporte y de contingencia. Este ecosistema permitió comprender el proceso formativo no como un conjunto de acciones aisladas, sino como un sistema dinámico, coherente y autorregulado que garantizó el logro de los resultados de aprendizaje previstos. La integración de estos tres tipos de estrategias evidenció una organización pedagógica intencionada, flexible y adaptativa, propia de los entornos de enseñanza-aprendizaje en ingeniería contemporánea.

Las estrategias núcleo constituyeron el eje central del ecosistema. Estas se enfocaron en el aprendizaje basado en proyectos (ABP), la aplicación de metodologías ágiles y el uso de GitHub como herramienta de control de versiones y trabajo colaborativo. A partir de estas acciones, los estudiantes模拟aron contextos reales de desarrollo de software, aplicando principios de planificación, ejecución y validación de productos.

Estas estrategias promovieron un aprendizaje significativo, centrado en la práctica, que alineó las competencias curriculares con la experiencia profesional. Tal como señalan Berglund y Lister (2020), la formación en ingeniería de software debe trascender la teoría y situar a los estudiantes en escenarios donde la experimentación y la iteración guíen la construcción del conocimiento.

En torno a este núcleo, se articularon las estrategias de soporte, que proporcionaron el andamiaje necesario para sostener el proceso pedagógico. Las guías prácticas, la re-

troalimentación formativa y el acompañamiento docente personalizado fueron esenciales para orientar la ejecución de las actividades, garantizando la comprensión progresiva de los contenidos y el desarrollo de la autonomía.

Estas estrategias generaron un entorno de apoyo estable, donde los estudiantes contaron con materiales estructurados, tutorías técnicas y espacios de comunicación digital. De acuerdo con Laurillard (2021), los ecosistemas de aprendizaje exitosos se caracterizan por integrar recursos tecnológicos y estrategias pedagógicas que permitan la participación activa y el soporte continuo a lo largo del proceso educativo.

Por su parte, las estrategias de contingencia actuaron como mecanismos de ajuste ante imprevistos surgidos durante la experiencia. Frente a dificultades de conectividad, desajustes en los tiempos de entrega o diferencias en los niveles de dominio técnico, se adoptaron medidas de flexibilización y acompañamiento entre pares. Estas acciones permitieron mantener la continuidad del aprendizaje y reforzar la colaboración como principio formativo. En coherencia con lo expuesto por Díaz-Noguera et al. (2023), la gestión de la contingencia en entornos educativos digitales debe concebirse como una oportunidad para desarrollar resiliencia y capacidad de respuesta, fortaleciendo la dimensión humana del aprendizaje tecnológico.

La interacción entre los tres tipos de estrategias configuró un entramado funcional, en el cual cada componente cumplió un rol complementario. Mientras las estrategias núcleo definieron el propósito y la estructura operativa del curso, las estrategias de soporte aseguraron las condiciones necesarias para su implementación, y las estrategias de contingencia garantizaron su adaptabilidad ante los cambios.

Este sistema de interdependencias generó un equilibrio entre la planificación y la flexibilidad, entre la estructura y la innovación, asegurando que el aprendizaje se mantuviera activo, contextualizado y significativo. En términos prácticos, el ecosistema estratégico se comportó como una red de procesos interconectados: las acciones pedagógicas (núcleo) fueron alimentadas por recursos y acompañamiento (soporte), y protegidas por respuestas adaptativas (contingencia).

Visualmente, esta arquitectura puede concebirse como un circuito de retroalimentación continua, donde la información fluye entre los distintos niveles del proceso educativo. El aprendizaje basado en proyectos constituye el motor central, sostenido por los recursos didácticos, el acompañamiento docente y la interacción digital, mientras que las contingencias operan como sensores que detectan desajustes y activan mecanismos de corrección.

Este enfoque sistémico refleja la naturaleza propia de la ingeniería de software, en la que el diseño, la implementación y la evaluación se integran en ciclos iterativos de mejora continua. Según (Kapur & Bielaczyc, 2021), este tipo de arquitectura pedagógica favorece la autoorganización del aprendizaje y promueve comunidades de práctica sostenibles en la formación universitaria.

En síntesis, la arquitectura del ecosistema estratégico consolidó una experiencia educativa coherente, sostenible y adaptable, donde cada estrategia se articuló en función de las otras. La complementariedad entre el núcleo, el soporte y la contingencia garantizó que los estudiantes no solo comprendieran los fundamentos de la ingeniería de software, sino que también desarrollaran competencias prácticas y actitudinales esenciales para su futuro profesional.

Este ecosistema permitió que la práctica docente trascendiera la simple transmisión de conocimientos, convirtiéndose en un proceso de diseño, experimentación y aprendizaje colectivo. En el siguiente apartado, se presentará la integración final del módulo, donde se analizará cómo este entramado estratégico contribuyó al logro de las competencias curriculares y a la consolidación de la experiencia como modelo replicable en la enseñanza de la ingeniería de software.

4.5. Evaluación integral de la experiencia: instrumentos, evidencias y validez del proceso formativo

4.5.1. Transición hacia la evaluación

Después de haber descrito el ecosistema estratégico de la experiencia -con sus estrategias núcleo, de soporte y de contingencia-, resulta necesario dar un paso hacia la dimensión evaluativa. En la docencia universitaria, especialmente en el ámbito de la Ingeniería de Software, evaluar no es solo medir resultados, sino validar la coherencia entre las estrategias aplicadas y las competencias que se pretendían alcanzar. Por ello, esta etapa busca garantizar que los logros formativos observados durante el proceso respondan efectivamente a los resultados de aprendizaje y al perfil de egreso planteado en el currículo.

Evaluar la experiencia permitió confirmar la pertinencia de las decisiones metodológicas adoptadas en la planificación y ejecución de las guías prácticas. Asimismo, posibilitó verificar que las estrategias implementadas -como el trabajo colaborativo, el uso de

repositorios en GitHub y la gestión de proyectos con metodologías ágiles- impactaron positivamente en el desarrollo de competencias técnicas y profesionales en los estudiantes. La evaluación, en este sentido, se convirtió en una herramienta reflexiva que fortaleció la mejora continua del proceso formativo, permitiendo reconocer aciertos, identificar áreas de ajuste y consolidar evidencias que respaldan el aprendizaje alcanzado.

Mostrar los instrumentos y criterios utilizados en la valoración de la experiencia aporta transparencia y credibilidad a la sistematización. Cada instrumento fue diseñado para medir la aplicación práctica de los conocimientos y la capacidad de los estudiantes para resolver problemas reales de ingeniería, asegurando que los resultados obtenidos sean válidos y consistentes con los objetivos propuestos. Evaluar, entonces, no solo implica calificar, sino comprender cómo las estrategias pedagógicas contribuyeron a transformar la práctica docente en evidencia verificable de aprendizaje.

En los apartados siguientes se presentarán los instrumentos de evaluación aplicados, los indicadores que guiaron el análisis y la forma en que las evidencias fueron interpretadas para otorgar sentido académico y confiabilidad al proceso. Esta sección constituye, por tanto, el puente entre la acción estratégica y la validación de sus resultados.

4.5.2. Instrumentos de evaluación aplicados

En el desarrollo de la experiencia docente vinculada con la asignatura, la evaluación ocupó un papel central para garantizar la coherencia entre las estrategias implementadas, los resultados de aprendizaje y las competencias profesionales del perfil de egreso. Para ello, se diseñó y aplicó un conjunto de instrumentos que permitieron valorar de forma integral los avances de los estudiantes, considerando tanto el desempeño técnico como las habilidades transversales desarrolladas a lo largo de las guías prácticas experimentales.

El primer instrumento empleado fue la rúbrica de evaluación del desempeño práctico, elaborada específicamente para valorar la ejecución de las guías de laboratorio. Esta rúbrica contempló criterios como: dominio de los contenidos teóricos, aplicación de metodologías ágiles en el desarrollo de proyectos, calidad del código en GitHub, cumplimiento de los requisitos funcionales, documentación técnica, y presentación de evidencias. Cada criterio fue ponderado sobre una escala de 1 a 10, con descriptores que permitían identificar niveles de logro desde “insuficiente” hasta “sobresaliente”. La rúbrica se aplicó al final de cada sesión práctica, lo que permitió ofrecer retroalimentación inmediata a los estudiantes y promover la mejora continua. Este instrumento fue clave para garantizar ob-

jetividad en la calificación y coherencia con las competencias específicas del curso, tales como la capacidad de diseñar, construir y verificar soluciones de software.

El segundo instrumento fue el cuestionario de autoevaluación, orientado a que los estudiantes reflexionaran sobre su propio proceso de aprendizaje. Este cuestionario se aplicó al finalizar cada unidad temática y contenía ítems de tipo Likert, centrados en el nivel de comprensión de los contenidos, el manejo de herramientas colaborativas (como GitHub y Trello), y el grado de participación en el trabajo en equipo. La autoevaluación permitió identificar percepciones individuales sobre el desempeño y promover la autorregulación del aprendizaje. Además, facilitó reconocer aspectos emocionales y motivacionales relacionados con la experiencia práctica, proporcionando información cualitativa complementaria a la evaluación objetiva de la rúbrica.

Como complemento, se utilizó un registro de observación sistemática, mediante el cual el docente registró comportamientos y evidencias durante el desarrollo de las actividades prácticas. Este instrumento permitió documentar aspectos como la puntualidad en la entrega de proyectos, la interacción entre pares, el uso correcto de entornos de desarrollo y la aplicación de buenas prácticas de programación. La observación directa fue especialmente útil para identificar el nivel de autonomía alcanzado por los estudiantes en la resolución de problemas técnicos y en la aplicación de metodologías ágiles dentro del entorno de trabajo.

Asimismo, se implementó una rúbrica de evaluación grupal destinada a valorar los productos colectivos de aprendizaje, tales como los proyectos integradores desarrollados en equipos. Esta rúbrica consideró la colaboración, la comunicación efectiva, la organización del repositorio en GitHub y la integración funcional del sistema web. De esta manera, se valoró no solo el resultado final del proyecto, sino también la dinámica del equipo y la capacidad de gestionar tareas bajo principios de responsabilidad compartida.

La pertinencia de estos instrumentos radicó en su coherencia con el enfoque de aprendizaje basado en proyectos y en la práctica experimental propia de la Ingeniería de Software. Mientras las rúbricas ofrecieron un marco estructurado y transparente de evaluación, los cuestionarios y registros de observación aportaron una mirada más reflexiva y contextual sobre los procesos de aprendizaje. En conjunto, permitieron evaluar tanto el saber hacer (desarrollo técnico) como el saber ser (trabajo colaborativo, pensamiento crítico y responsabilidad profesional), consolidando así una visión holística del desempeño estudiantil.

En conclusión, los instrumentos aplicados no solo facilitaron la valoración de los resultados obtenidos, sino que también constituyeron una fuente valiosa de evidencias para el análisis posterior. Su implementación sistemática permitió verificar la efectividad de las estrategias didácticas y garantizar la validez de los aprendizajes alcanzados. En el siguiente apartado se presentarán los indicadores de evaluación y criterios de validez, que permitirán interpretar con mayor precisión la información recolectada a través de estos instrumentos y asegurar la consistencia del proceso evaluativo dentro de la sistematización.

4.5.3. Indicadores de evaluación y criterios de validez

Una vez definidos y aplicados los instrumentos de evaluación, el siguiente paso consistió en establecer los indicadores de evaluación que permitieran interpretar los resultados y determinar el grado de logro de las competencias curriculares previstas en la asignatura Introducción a la Ingeniería de Software. Estos indicadores fueron diseñados de manera coherente con los objetivos del curso y con las competencias específicas del perfil de egreso, de modo que su aplicación proporcionara evidencia objetiva, confiable y verificable sobre los aprendizajes alcanzados.

El primer indicador establecido fue el de cumplimiento de entregables técnicos, el cual permitió valorar la responsabilidad, la organización y la capacidad de gestión del tiempo de los estudiantes frente a las actividades propuestas. Este indicador se reflejó en la entrega oportuna de proyectos en GitHub, el cumplimiento de los requisitos funcionales de las aplicaciones web y la documentación técnica generada durante las guías prácticas. Su pertinencia radica en que la puntualidad y la calidad de las entregas son evidencias tangibles del desarrollo de competencias profesionales como la disciplina, la planificación y la responsabilidad técnica, fundamentales en el ámbito de la ingeniería de software.

El segundo indicador fue la calidad del producto de software, entendido como la correspondencia entre la solución desarrollada y los criterios de diseño, usabilidad y funcionalidad definidos en la guía práctica. Este indicador permitió verificar el nivel de dominio de los conocimientos técnicos, la aplicación de buenas prácticas de programación, la eficiencia del código y la correcta implementación de patrones arquitectónicos. Para su medición se utilizaron las rúbricas de desempeño y las revisiones técnicas realizadas por el docente, asegurando así un proceso de evaluación estandarizado y transparente.

Un tercer indicador fue la participación activa y colaborativa en el desarrollo de las actividades de laboratorio y proyectos grupales. Este indicador buscó evidenciar la contribución individual dentro del equipo de trabajo, la comunicación efectiva, la toma de decisiones en conjunto y la resolución de conflictos. Los registros de observación docente y los reportes del repositorio GitHub (historial de commits y contribuciones) sirvieron como evidencia de este proceso. Este criterio es esencial en el campo de la ingeniería de software, donde el trabajo colaborativo constituye una competencia transversal imprescindible para el ejercicio profesional.

El cuarto indicador se centró en la capacidad de reflexión y mejora continua, evaluando la disposición del estudiante para analizar sus propios errores, recibir retroalimentación y realizar ajustes en las siguientes iteraciones del proyecto. Este indicador se midió mediante los cuestionarios de autoevaluación y las bitácoras personales, donde los estudiantes registraron los aprendizajes obtenidos, las dificultades enfrentadas y las estrategias aplicadas para superarlas. Su relevancia se sustenta en la noción de aprendizaje autorregulado, que según Zimmerman (2020) y Panadero (2021), constituye una competencia clave para el desarrollo de profesionales autónomos y críticos.

En cuanto a los criterios de validez aplicados para garantizar la confiabilidad de los resultados, se consideraron tres principios fundamentales: la triangulación de fuentes, la consistencia interna y la transparencia del proceso evaluativo.

La triangulación se logró al contrastar los resultados obtenidos a través de distintos instrumentos (rúbricas, autoevaluaciones, observaciones y entregables digitales), lo que permitió obtener una visión más completa y evitar sesgos derivados de una única fuente de información. Flick (2014b) destaca que este procedimiento fortalece la credibilidad de los estudios educativos, al combinar evidencia cualitativa y cuantitativa para sustentar los hallazgos.

La consistencia interna se aseguró mediante la aplicación uniforme de los instrumentos y la revisión cruzada de los resultados entre sesiones prácticas, evitando variaciones arbitrarias en la interpretación de los criterios. Para ello, se diseñaron matrices de seguimiento y se realizaron sesiones de revisión con los docentes colaboradores del laboratorio, con el fin de mantener la coherencia en la aplicación de los estándares de evaluación. Este enfoque responde a lo planteado por Yin (2014), quien señala que la confiabilidad en la investigación educativa depende de la estabilidad de los procedimientos aplicados.

Finalmente, la transparencia se garantizó a través de la documentación detallada del proceso de evaluación, incluyendo la publicación de las rúbricas en la plataforma ins-

titucional y la entrega de retroalimentaciones personalizadas. Esta práctica fortaleció la confianza de los estudiantes en el proceso, asegurando que cada calificación estuviera sustentada en evidencias observables y criterios previamente definidos.

En síntesis, los indicadores diseñados y los criterios de validez implementados permitieron construir un proceso evaluativo riguroso, alineado con los estándares de calidad académica de la universidad y con las exigencias del perfil profesional del ingeniero de software. En el siguiente apartado se presentará el análisis preliminar de las evidencias, donde se interpretarán los resultados obtenidos a partir de estos indicadores, identificando patrones de aprendizaje y áreas de mejora para la siguiente fase del proceso de sistematización.

4.5.4. Análisis preliminar de evidencias

Luego de definir los indicadores y criterios de validez, el siguiente paso consistió en analizar las evidencias recolectadas a través de los distintos instrumentos aplicados durante la experiencia docente. Este análisis permitió transformar los datos obtenidos - tanto cuantitativos como cualitativos- en información significativa que reflejó el nivel de logro de las competencias y los resultados de aprendizaje previstos en la asignatura de Introducción a la Ingeniería de Software.

Las principales evidencias surgieron de las rúbricas de desempeño, los repositorios GitHub de los proyectos estudiantiles, los cuestionarios de autoevaluación y los registros de observación realizados durante las sesiones de laboratorio. Estas fuentes permitieron observar, de manera integral, cómo los estudiantes aplicaron los conocimientos teóricos en contextos prácticos, evidenciando progresos en su capacidad para diseñar, desarrollar y validar aplicaciones web.

El análisis de las rúbricas mostró que un 85 % de los estudiantes alcanzó niveles de desempeño altos o muy altos en los criterios relacionados con la funcionalidad del sistema, la organización del código y la documentación técnica. Este resultado evidenció que las estrategias metodológicas implementadas -como el aprendizaje basado en proyectos y el uso de GitHub para la gestión del código- contribuyeron de manera directa al desarrollo de competencias técnicas. Sin embargo, un 15 % de los estudiantes presentó dificultades en la integración de componentes del sistema, lo que sugiere la necesidad de reforzar el acompañamiento docente en la fase de pruebas e implementación.

Por otro lado, los cuestionarios de autoevaluación aportaron una perspectiva cualitativa sobre la experiencia formativa. La mayoría de los estudiantes manifestó sentirse más seguros al utilizar herramientas de control de versiones, trabajar en equipo y documentar sus proyectos de manera profesional. Algunos reconocieron, no obstante, desafíos relacionados con la coordinación grupal y la gestión del tiempo, aspectos que influyeron en la calidad final de los entregables. Estas percepciones evidencian la relevancia de mantener espacios de reflexión individual como parte del proceso de evaluación, tal como plantean Panadero y Alonso-Tapia (2020) al destacar el valor metacognitivo de la autoevaluación en el aprendizaje autónomo.

Los registros de observación docente también proporcionaron información clave sobre el comportamiento y la interacción de los estudiantes durante las actividades prácticas. Se identificó un incremento progresivo en la participación y colaboración entre pares, así como en el uso responsable de los recursos digitales institucionales. Las observaciones indicaron que, en las primeras sesiones, los equipos mostraban una distribución desigual de tareas; sin embargo, hacia el final del módulo se evidenció una mayor organización interna, favorecida por el uso de tableros Kanban y la comunicación asertiva entre los miembros del grupo.

Adicionalmente, el análisis de los repositorios GitHub permitió obtener evidencia objetiva de la participación individual de cada estudiante. Los datos de commits y actualizaciones demostraron un incremento sostenido en la frecuencia de contribuciones, especialmente en las semanas finales, lo que sugiere una consolidación del aprendizaje técnico y una mejora en la gestión colaborativa del código fuente. Este tipo de evidencia digital resultó especialmente útil para verificar el indicador de participación activa y para asegurar la transparencia del proceso evaluativo.

Desde un enfoque cuantitativo, los resultados generales evidenciaron que los estudiantes alcanzaron, en promedio, un 82 % de cumplimiento de los indicadores establecidos, destacando en las dimensiones de desempeño técnico, colaboración y documentación. Este valor respalda la coherencia entre las estrategias didácticas aplicadas y los logros obtenidos, al mismo tiempo que permite identificar oportunidades de mejora en la planificación de futuras guías prácticas, particularmente en la etapa de validación de software y pruebas unitarias.

En términos de análisis cualitativo, las evidencias recogidas a través de las reflexiones estudiantiles y las observaciones docentes reflejan un proceso de aprendizaje gradual y significativo. Se observaron mejoras notables en la comprensión de los principios de

ingeniería de software, el uso responsable de herramientas digitales y la integración de metodologías ágiles en contextos académicos. Estos hallazgos concuerdan con lo propuesto por R. Stake (1995), quien sostiene que el análisis interpretativo de la evidencia educativa permite comprender no solo los resultados, sino también los procesos que los generan.

En síntesis, el análisis preliminar de evidencias confirmó la efectividad de las estrategias implementadas y la validez de los instrumentos aplicados. Las rúbricas proporcionaron datos objetivos, mientras que las autoevaluaciones y observaciones enriquecieron la interpretación de los resultados desde una mirada reflexiva y formativa. En conjunto, estas evidencias permitieron identificar patrones de mejora, fortalecer la retroalimentación docente y consolidar un proceso evaluativo integral, transparente y coherente con las competencias profesionales del futuro ingeniero de software.

En el siguiente apartado se presentará una reflexión crítica sobre la validez, los sesgos y la factibilidad del proceso evaluativo, con el propósito de reconocer sus fortalezas y limitaciones, y de proyectar estrategias para optimizar las futuras experiencias de evaluación en contextos prácticos de formación universitaria.

4.5.5. Reflexión sobre validez, sesgos y factibilidad

La evaluación realizada durante la sistematización de la experiencia docente permitió obtener resultados significativos sobre el logro de las competencias curriculares. Sin embargo, como todo proceso evaluativo, no estuvo exenta de limitaciones y desafíos que ameritan una reflexión crítica. Analizar la validez, los posibles sesgos y la factibilidad del procedimiento es esencial para garantizar la credibilidad de los hallazgos y fortalecer la mejora continua del proceso en futuras experiencias.

En cuanto a la validez del proceso evaluativo, esta se sustentó en la coherencia entre los instrumentos aplicados, los indicadores establecidos y los resultados de aprendizaje definidos en el programa de la asignatura Introducción a la Ingeniería de Software. La consistencia metodológica permitió verificar que los datos obtenidos representaran con fidelidad el desempeño de los estudiantes. La triangulación de información -a través de rúbricas, autoevaluaciones, observaciones y evidencias digitales- reforzó la confiabilidad de los resultados. De acuerdo con Creswell y Creswell (2020), la validez en la investigación educativa se consolida cuando los datos convergen desde diferentes fuentes y métodos, permitiendo una interpretación más robusta de los hallazgos.

Además, se procuró mantener una validez de contenido, asegurando que los instrumentos de evaluación midieran efectivamente las competencias planteadas, tales como el dominio de herramientas de control de versiones, la colaboración en entornos ágiles y la capacidad para desarrollar soluciones tecnológicas funcionales. La claridad de los criterios de las rúbricas y la retroalimentación continua fueron factores determinantes para garantizar que los resultados fueran representativos de los aprendizajes reales. McMillan y Schumacher (2022) destacan que la validez de contenido depende de la correspondencia directa entre lo que se enseña, lo que se evalúa y lo que se espera demostrar en términos de desempeño.

En relación con los sesgos identificados, uno de los más evidentes fue el derivado de la autoevaluación. Algunos estudiantes tendieron a sobrevalorar su desempeño, lo cual afectó parcialmente la objetividad de los resultados individuales. Para mitigar este sesgo, se compararon las respuestas con los registros de participación en GitHub y las observaciones directas del docente, aplicando un enfoque de contraste de datos. También se detectaron ligeras variaciones en la interpretación de los criterios de la rúbrica por parte de los docentes auxiliares del laboratorio, lo que pudo influir en la calificación de ciertas evidencias. Este tipo de sesgo, conocido como sesgo de interpretación, fue atendido mediante reuniones de calibración y revisión cruzada de las evaluaciones, una práctica que Kvale y Brinkmann (2021) recomiendan para fortalecer la fiabilidad y la coherencia interevaluador.

Otro aspecto considerado fue el sesgo de selección de evidencias, especialmente en el análisis de proyectos grupales. En algunos casos, los equipos presentaron sus repositorios con información incompleta o sin actualizar, lo que limitó la capacidad de análisis integral de la participación de todos los miembros. Para reducir este riesgo, se establecieron controles de registro semanales y se reforzó la importancia de documentar las actividades en tiempo real dentro de la plataforma GitHub, garantizando la trazabilidad de los avances y el seguimiento del progreso individual.

Respecto a la factibilidad del proceso evaluativo, los resultados demuestran que la planificación y ejecución fueron viables, aunque requirieron un esfuerzo sostenido de seguimiento. La implementación de múltiples instrumentos de evaluación supuso una carga significativa de tiempo para el docente, especialmente durante las fases de revisión y retroalimentación personalizada. Sin embargo, este esfuerzo se justificó por la calidad y profundidad de la información obtenida, que permitió un análisis integral y transparente. En consonancia con R. B. Johnson y Christensen (2023), la factibilidad de un proceso

evaluativo depende del equilibrio entre la rigurosidad metodológica y la disponibilidad real de recursos humanos y tecnológicos.

En cuanto a las condiciones institucionales, la infraestructura tecnológica disponible —laboratorios equipados, conexión estable y acceso a plataformas digitales— favoreció la factibilidad técnica de la evaluación. No obstante, se identificaron limitaciones asociadas a la dependencia de la conectividad y al tiempo extracurricular necesario para la colaboración remota de los estudiantes. Estas dificultades, aunque puntuales, sirvieron para ajustar la planificación de futuras guías prácticas, promoviendo espacios asincrónicos y estrategias de gestión del tiempo más flexibles, tal como sugieren Boud y Falchikov (2020) al referirse a la importancia de diseñar procesos evaluativos sostenibles y adaptables al contexto educativo.

En términos generales, la reflexión sobre la validez, los sesgos y la factibilidad del proceso permitió reconocer la solidez del enfoque evaluativo adoptado. Los instrumentos y criterios aplicados demostraron ser pertinentes y confiables, mientras que los desafíos identificados constituyen oportunidades de mejora para futuras iteraciones. La experiencia evidenció que un proceso de evaluación no solo debe enfocarse en medir resultados, sino también en garantizar su legitimidad metodológica y ética, reconociendo las condiciones reales del contexto educativo.

4.6. Reflexión final y proyección institucional de la experiencia docente

4.6.1. Transición hacia la reflexión final

Al culminar el proceso de evaluación de la experiencia educativa, se hace necesario detenerse un momento para mirar con distancia crítica todo lo vivido. Las evidencias analizadas, los indicadores aplicados y los resultados obtenidos no son únicamente datos que confirman logros, sino también huellas que invitan a pensar sobre el sentido profundo de la práctica docente y su impacto en la formación de los estudiantes. Este es, precisamente, el propósito de esta transición: abrir el espacio para la reflexión final, donde la evaluación deja de ser un cierre técnico y se convierte en una oportunidad de aprendizaje profesional.

Reflexionar críticamente sobre la experiencia permite reconocer que detrás de cada estrategia, instrumento o evidencia, existe un proceso humano de búsqueda, error y me-

jora continua. En el caso de la enseñanza universitaria, particularmente en el campo de la Ingeniería de Software, esta mirada reflexiva posibilita comprender cómo las metodologías aplicadas, las dinámicas de trabajo colaborativo y la integración de la tecnología contribuyeron no solo al desarrollo de competencias técnicas, sino también al fortalecimiento de capacidades analíticas, comunicativas y éticas. La reflexión, por tanto, amplía la mirada del docente y lo conecta con un horizonte de transformación más allá de su propio contexto institucional.

Esta etapa final abre la posibilidad de transferir lo aprendido a nuevos escenarios, adaptando las estrategias exitosas y revisando críticamente aquellas que requirieron ajustes. Es el momento de reconocer que toda experiencia formativa se consolida cuando se comparte, se analiza y se resignifica desde la práctica. De esta manera, la transición hacia la reflexión final no solo marca el cierre de un ciclo de sistematización, sino que inaugura un proceso continuo de mejora, innovación y compromiso con una docencia universitaria más consciente, crítica y transformadora.

4.6.2. Reflexión crítica sobre la experiencia

La experiencia desarrollada permitió evidenciar el valor transformador de los procesos pedagógicos cuando se construyen desde la reflexión, la planificación y la evaluación constante. En el contexto universitario, particularmente en el ámbito de la Ingeniería de Software, los resultados obtenidos muestran que la integración de metodologías activas -como el aprendizaje basado en proyectos y la práctica experimental guiada- propició un cambio sustancial en la dinámica del aula.

Los estudiantes pasaron de ser receptores de información a protagonistas de su propio proceso formativo, asumiendo responsabilidades concretas en la resolución de problemas y en la construcción de conocimiento aplicable a escenarios reales de desarrollo tecnológico. Esta autonomía favoreció la adquisición de competencias profesionales tales como la gestión de proyectos, la colaboración interdisciplinaria y el pensamiento analítico.

Desde la perspectiva docente, la experiencia representó una oportunidad para fortalecer la capacidad de innovación y la mirada crítica frente a las propias prácticas pedagógicas. El diseño de estrategias de evaluación coherentes con los resultados de aprendizaje permitió observar avances más cualitativos que cuantitativos, vinculados a la comprensión profunda de los conceptos y a la transferencia del conocimiento hacia la práctica.

Asimismo, el trabajo colaborativo entre docentes tutores facilitó la creación de entornos de aprendizaje más inclusivos, donde se valoró la participación activa, la creatividad y el compromiso con la calidad de los productos desarrollados. En este sentido, el proceso no solo mejoró los resultados académicos, sino también las actitudes hacia el aprendizaje autónomo y la responsabilidad profesional.

A nivel institucional, el proyecto aportó evidencias relevantes para fortalecer la cultura de innovación educativa y la coherencia curricular dentro de la carrera. Los resultados obtenidos demostraron que es posible alinear las actividades prácticas con las competencias del perfil de egreso sin perder rigurosidad técnica ni profundidad conceptual.

Además, la sistematización permitió generar materiales, guías y buenas prácticas que hoy pueden ser replicadas por otros docentes, promoviendo una visión compartida sobre la enseñanza universitaria basada en la experimentación, la evaluación formativa y la transferencia del conocimiento.

4.6.3. Tensiones y resistencias encontradas

El desarrollo de la experiencia también implicó enfrentar diversos desafíos que pusieron a prueba la capacidad de adaptación del equipo docente y de los estudiantes. Una de las principales tensiones estuvo relacionada con el cambio de paradigma pedagógico: pasar de una enseñanza tradicional centrada en la exposición teórica hacia una metodología práctica y colaborativa generó incertidumbre, tanto en el profesorado como en los alumnos.

Algunos estudiantes manifestaron resistencia inicial a trabajar en equipos multidisciplinarios, a planificar proyectos de largo alcance o a asumir evaluaciones basadas en evidencias más que en exámenes convencionales. Este proceso de transición demandó tiempo, acompañamiento constante y estrategias de motivación que permitieran superar la inercia del modelo tradicional.

A nivel institucional, se evidenciaron limitaciones vinculadas con la disponibilidad de recursos tecnológicos y la infraestructura necesaria para sostener las prácticas experimentales. En ocasiones, el acceso restringido a laboratorios, equipos y conectividad afectó el desarrollo oportuno de las actividades planificadas.

No obstante, estas dificultades se transformaron en oportunidades para fortalecer la gestión académica, optimizar los recursos disponibles y promover el trabajo colaborativo entre diferentes áreas. Asimismo, la carga administrativa y la necesidad de cumplir con

cronogramas ajustados generaron momentos de tensión que exigieron una organización más eficiente y el fortalecimiento de la comunicación entre docentes y autoridades.

También surgieron resistencias sutiles vinculadas a la cultura institucional y a la concepción tradicional del rol docente. Algunos colegas percibieron inicialmente la sistematización de experiencias como una tarea adicional y no como una oportunidad de aprendizaje profesional.

Sin embargo, el proceso demostró que reflexionar y documentar la práctica permite reconocer los avances, visibilizar los resultados y generar conocimiento compartido. Superar estas resistencias fue posible gracias a la apertura al diálogo, la evidencia de los logros alcanzados y el compromiso ético con la mejora continua. Estas tensiones, lejos de constituir obstáculos, se convirtieron en aprendizajes sobre la importancia de la colaboración, la flexibilidad y la empatía institucional.

4.6.4. Aprendizajes personales, colectivos e institucionales

En el plano personal, esta experiencia me permitió comprender que la docencia universitaria es un proceso en permanente construcción y que la verdadera innovación surge del diálogo entre la teoría, la práctica y la evaluación.

Aprendí a mirar el aula como un espacio de investigación aplicada, donde cada actividad y cada interacción constituyen una oportunidad para generar conocimiento nuevo. Esta perspectiva fortaleció mi sentido de responsabilidad como docente investigador y me impulsó a sistematizar las experiencias con un enfoque crítico, consciente de que el valor de la enseñanza radica tanto en los resultados como en la reflexión que los sustenta.

En el ámbito colectivo, el trabajo articulado entre docentes, coordinadores y estudiantes consolidó una comunidad de aprendizaje comprometida con la mejora continua. Las reuniones de planificación, los intercambios de experiencias y las retroalimentaciones cruzadas generaron un clima de confianza y colaboración que trascendió las fronteras del aula.

Este aprendizaje compartido permitió uniformar criterios, optimizar las estrategias metodológicas y fortalecer la coherencia entre las asignaturas del plan de estudios, asegurando que los resultados de aprendizaje respondan efectivamente al perfil de egreso. Además, el diálogo entre pares docentes favoreció la construcción de una cultura institucional orientada hacia la calidad educativa y la innovación permanente.

En el nivel institucional, la sistematización dejó aprendizajes valiosos relacionados con la gestión académica y el desarrollo de políticas de apoyo a la docencia. Se evidenció la necesidad de seguir fortaleciendo la infraestructura tecnológica, los procesos de acompañamiento pedagógico y la evaluación basada en evidencias.

La experiencia sirvió también como modelo de referencia para futuras prácticas de sistematización y como insumo para la autoevaluación institucional, en el marco de los procesos de acreditación y mejora continua. Estos aprendizajes confirman que cuando la docencia universitaria se documenta, analiza y comparte, se convierte en conocimiento institucional transferible que impulsa la transformación educativa.

4.6.5. Síntesis reflexiva

Mirar en retrospectiva este proceso permite afirmar que la experiencia fue más que un ejercicio de aplicación metodológica; constituyó un camino de transformación profesional y humana. La evaluación, los resultados y las tensiones se integran en una narrativa coherente que evidencia crecimiento, aprendizaje y compromiso.

La docencia, entendida como práctica reflexiva, demanda observar críticamente los propios actos y comprender que cada dificultad puede convertirse en un punto de inflexión hacia la mejora. Así, las resistencias encontradas fueron, en realidad, señales del cambio y recordatorios de que innovar implica salir de la zona de confort.

En síntesis, la experiencia fortaleció la visión de una docencia universitaria orientada al aprendizaje significativo, a la interdisciplinariedad y a la construcción colaborativa del conocimiento. La reflexión crítica permitió reconocer que los logros alcanzados no son un punto de llegada, sino un punto de partida para seguir avanzando hacia prácticas más inclusivas, sostenibles y alineadas con los desafíos actuales de la educación superior.

Proyectar estos aprendizajes hacia nuevos escenarios implica asumir la responsabilidad de transferir el conocimiento generado, inspirar a otros docentes y consolidar comunidades académicas que aprendan unas de otras. En última instancia, esta reflexión confirma que la verdadera evaluación de una experiencia educativa no está solo en sus resultados, sino en la capacidad de transformar a quienes la viven y en su potencial para seguir generando cambios positivos en la institución y en la sociedad.

Bibliografía

- Barnett, R., & Jackson, N. (2019). *Learning for an Unknown Future: Higher Education and the Human Condition*. Routledge.
- Berglund, A., & Lister, R. (2020). Project-based and practice-oriented learning in software engineering education: Lessons from global contexts. *ACM Transactions on Computing Education*, 20(4), 1-21. <https://doi.org/10.1145/3415197>
- Biggs, J., & Tang, C. (2011). *Teaching for Quality Learning at University* (4.^a ed.). McGraw-Hill Education.
- Boud, D., & Falchikov, N. (2020). *Rethinking Assessment in Higher Education: Learning for the Longer Term* (2.^a ed.). Routledge.
- Boud, D., & Molloy, E. (2019). *Feedback in Higher and Professional Education: Understanding It and Doing It Well*. Routledge. <https://doi.org/10.4324/9780429483180>
- Carlino, P. (2019). *Escribir, leer y aprender en la universidad: Una introducción a la alfabetización académica* (2.^a ed.). Fondo de Cultura Económica.
- Creswell, J. W., & Creswell, J. D. (2020). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches* (5.^a ed.). SAGE Publications.
- de Sousa, B., Costa, R., & Gomes, A. (2021). Gamification in engineering education: Increasing student motivation and engagement through digital strategies. *Education and Information Technologies*, 26(5), 5373-5390. <https://doi.org/10.1007/s10639-021-10522-4>
- Díaz-Noguera, M. D., Rodríguez-García, A. M., & Sánchez-Prieto, J. C. (2023). Digital resilience in higher education: Lessons learned from flexible teaching strategies in engineering programs. *Computers & Education*, 197, 104749. <https://doi.org/10.1016/j.compedu.2023.104749>
- Facione, P. A. (2020). *Critical Thinking: What It Is and Why It Counts*. Insight Assessment.
- Fernández, J., & Rodríguez, M. (2020). Diseño de guías didácticas interactivas para el desarrollo de competencias en ingeniería. *Revista Iberoamericana de Educación Superior*, 11(31), 129-146. <https://doi.org/10.22201/iisue.20072872e.2020.31.581>
- Flick, U. (2014a). *Introducción a la investigación cualitativa* (5.^a ed.). Morata.
- Flick, U. (2014b). *La gestión de la calidad en investigación cualitativa*. Ediciones Morata.
- Flick, U. (2019). *An Introduction to Qualitative Research* (6.^a ed.). SAGE Publications.

- Freeman, S., Eddy, S. L., & Hogan, K. A. (2019). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 116(39), 19251-19257. <https://doi.org/10.1073/pnas.1910153116>
- García-Holgado, A., & García-Peñalvo, F. J. (2022). Digital ecosystems in engineering education: Towards an integrated learning environment for the 21st century. *Education and Information Technologies*, 27(6), 8675-8693. <https://doi.org/10.1007/s10639-022-11048-2>
- González, M., & Ramírez, J. (2021). Pensamiento crítico y habilidades blandas en la educación superior: Retos para la formación profesional. *Revista Educación y Sociedad*, 19(2), 45-60.
- Hargie, O. (2021). *Skilled Interpersonal Communication: Research, Theory and Practice* (7.ª ed.). Routledge.
- Hodges, C., Moore, S., Lockee, B., Trust, T., & Bond, A. (2020). The difference between emergency remote teaching and online learning. *Educause Review*, 55(1), 1-12.
- Jara, O. (2018). *La sistematización de experiencias: Aprendizajes y desafíos para la educación popular*. Alforja.
- Jaramillo Gómez, D. L., Álvarez Maestre, A. J., Parada Trujillo, A. E., Pérez Fuentes, C. A., Bedoya Ortiz, D. H., & Sanabria Alarcón, R. K. (2025). Determining factors for the development of critical thinking in higher education. *Journal of Intelligence*, 13(6), 59. <https://doi.org/10.3390/jintelligence13060059>
- Johnson, D. W., Johnson, R. T., & Smith, K. A. (2020). Cooperative learning: Improving university instruction by basing practice on validated theory. *Journal on Excellence in College Teaching*, 31(2), 34-68.
- Johnson, R. B., & Christensen, L. (2023). *Educational Research: Quantitative, Qualitative, and Mixed Approaches* (7.ª ed.). SAGE Publications.
- Kapur, M., & Bielaczyc, K. (2021). Designing for productive failure: Principles for integrating adaptive and collaborative learning. *Educational Psychologist*, 56(4), 243-256. <https://doi.org/10.1080/00461520.2021.1963667>
- Kolb, D. A. (2015). *Experiential Learning: Experience as the Source of Learning and Development* (2.ª ed.). Pearson Education.
- Kvale, S., & Brinkmann, S. (2021). *InterViews: Learning the Craft of Qualitative Research Interviewing* (4.ª ed.). SAGE Publications.

- Lai, E. R. (2022). *Critical Thinking in the 21st Century: A Framework for Learning and Innovation*. Pearson Education.
- Laurillard, D. (2021). *Teaching as a Design Science: Building Pedagogical Patterns for Learning Technologies* (2.^a ed.). Routledge. <https://doi.org/10.4324/9781003142012>
- McMillan, J. H., & Schumacher, S. (2022). *Research in Education: Evidence-Based Inquiry* (9.^a ed.). Pearson.
- Moe, N. B., Dingsøyr, T., & Kasauli, R. (2022). Agile project management in software engineering education: Lessons learned from student teams. *Journal of Systems and Software*, 188, 111279. <https://doi.org/10.1016/j.jss.2022.111279>
- Morales, J., Paredes, R., & Cifuentes, A. (2021). Formación ética en ingeniería: desafíos para la educación superior latinoamericana. *Revista Iberoamericana de Educación Superior*, 12(33), 45-62.
- Panadero, E. (2021). A review of self-regulated learning: Six models and four directions for research. *Frontiers in Psychology*, 12, 795-808. <https://doi.org/10.3389/fpsyg.2021.795408>
- Panadero, E., & Alonso-Tapia, J. (2020). Revisión de las bases teóricas y la evidencia empírica sobre la autoevaluación del aprendizaje. *Revista de Psicología Educativa*, 26(2), 79-88. <https://doi.org/10.5093/psed2020a9>
- Panadero, E., & Broadbent, J. (2018). Developing self-regulated learners: A review of current evidence and future directions. *Frontiers in Psychology*, 9, 1664. <https://doi.org/10.3389/fpsyg.2018.01664>
- Pisoni, G., Godoy, D., & Ponce, J. (2021). Implementing agile methodologies in software engineering education: An empirical study of student collaboration and learning outcomes. *IEEE Transactions on Education*, 64(3), 250-259. <https://doi.org/10.1109/TE.2021.3056789>
- Prince, M. (2021). Active learning for engineering education: Current practices and future directions. *European Journal of Engineering Education*, 46(1), 5-22. <https://doi.org/10.1080/03043797.2020.1818698>
- Ragusa, G., & Chong, E. (2021). Engineering mentorship programs and their impact on student retention and success. *Journal of Engineering Education*, 110(3), 512-530. <https://doi.org/10.1002/jee.20397>
- Stake, R. (1995). *The Art of Case Study Research*. SAGE Publications.
- Stake, R. E. (2020). *The Art of Case Study Research* (Revised). SAGE Publications.

- Thomas, J., & Leifer, L. (2020). Project-based learning in engineering education: Redefining student engagement and applied knowledge. *European Journal of Engineering Education*, 45(5), 711-728. <https://doi.org/10.1080/03043797.2020.1718612>
- Universidad Estatal de Milagro (UNEMI). (2021). Modelo Educativo de la Universidad Estatal de Milagro.
- Universidad Estatal de Milagro (UNEMI). (2023). Marco curricular institucional de la Universidad Estatal de Milagro.
- Van Leeuwen, A., & Janssen, J. (2019). A systematic review of teacher guidance during collaborative learning in primary and secondary education. *Educational Research Review*, 27, 71-89. <https://doi.org/10.1016/j.edurev.2019.02.001>
- Yin, R. K. (2014). *Case Study Research: Design and Methods* (5.^a ed.). SAGE Publications.
- Yin, R. K. (2018). *Case Study Research and Applications: Design and Methods* (6.^a ed.). SAGE Publications.
- Zabalza, M. A. (2019). *Competencias docentes del profesorado universitario: Calidad y desarrollo profesional* (3.^a ed.). Narcea.
- Zawacki-Richter, O., & Kerres, M. (2020). Educational technologies and the changing paradigms in higher education. *British Journal of Educational Technology*, 51(6), 1905-1919. <https://doi.org/10.1111/bjet.12983>
- Zhang, Q., Wang, Y., & Liu, Z. (2023). Collaborative coding and learning analytics in higher education: A GitHub-based instructional model. *Computers & Education*, 197, 104700. <https://doi.org/10.1016/j.compedu.2023.104700>
- Zhu, M., & Park, J. (2021). Peer tutoring and collaborative learning in online engineering courses: Impacts on engagement and achievement. *Computers & Education*, 174, 104302. <https://doi.org/10.1016/j.compedu.2021.104302>
- Zimmerman, B. J. (2020). Motivating self-regulated learners: The future of educational psychology. *Journal of Educational Psychology*, 112(2), 331-343.

Ingeniería de software en la educación universitaria: experiencias didácticas para una formación integral

Resumen

El libro RISEI I presenta los resultados del Programa Institucional de Sistematización e Innovación Educativa de la Universidad Católica de Cuyo, orientado a fortalecer la investigación pedagógica, la mejora continua de la docencia universitaria y la profesionalización del rol docente. A lo largo de cuatro capítulos, se reconstruyen experiencias que integran la innovación metodológica, la reflexión crítica y la producción de conocimiento situado, enmarcadas en el compromiso institucional con la calidad educativa y la formación integral del estudiante. Cada capítulo documenta procesos de enseñanza transformadores en distintas áreas disciplinares, destacando el empleo de metodologías activas, tecnologías educativas, estrategias de evaluación formativa y criterios de coherencia curricular. La obra articula teoría y práctica mediante la sistematización de experiencias como vía para convertir la práctica docente en evidencia, aprendizaje institucional y motor de cambio. Presenta problemas pedagógicos reales, decisiones didácticas, instrumentos de evaluación y resultados de aprendizaje, además de reflexiones sobre el trabajo colaborativo y la transferencia de buenas prácticas. El volumen consolida un modelo de investigación educativa que vincula formación docente, producción académica, gestión institucional y cultura de innovación, promoviendo comunidades de práctica y redes intersede. En síntesis, RISEI I documenta, analiza y proyecta un camino institucional hacia la transformación de la enseñanza universitaria desde la investigación aplicada, la sistematización y la innovación pedagógica sostenida, contribuyendo al desarrollo de una universidad crítica, reflexiva y comprometida con su tiempo.

Palabras claves: Innovación educativa; Docencia universitaria; Investigación educativa; Sistematización; Educación superior

Abstract

The book RISEI I presents the results of the Institutional Programme for Systematisation and Educational Innovation at the Catholic University of Cuyo, aimed at strengthening pedagogical research, the continuous improvement of university teaching and the professionalisation of the teaching role. Throughout four chapters, experiences are reconstructed that integrate methodological innovation, critical reflection, and the production of situated knowledge, framed within the institutional commitment to educational quality and comprehensive student training. Each chapter documents transformative teaching processes in different disciplinary areas, highlighting the use of active methodologies, educational technologies, formative assessment strategies, and curriculum coherence criteria. The work articulates theory and practice by systematising experiences as a way of turning teaching practice into evidence, institutional learning and a driver of change. It presents real pedagogical problems, teaching decisions, assessment tools and learning outcomes, as well as reflections on collaborative work and the transfer of good practices. The volume consolidates an educational research model that links teacher training, academic production, institutional management and a culture of innovation, promoting communities of practice and inter-site networks. In short, RISEI I documents, analyses and projects an institutional path towards the transformation of university teaching through applied research, systematisation and sustained pedagogical innovation, contributing to the development of a critical, reflective university committed to its time.

Keywords : Educational innovation; University teaching; Educational research; Systematisation; Higher education